Shielded Virtual Machines | Go Unsafe with C# | Apr Pools | Engineering best practices

ISSUE 7

MICRO FOCUS

TECH JOURNAL 2017

www.microfocus.com

THINK . INNOVATE . WRITE

WHAT IS HOT?

Cloud Access Security Broker

DATABASE

Capacity Planning for Big Data Applications

MOBILE

"pCloudy" Simplifying Mobile App Testing

TESTING

Exploring the Exploratory Testing

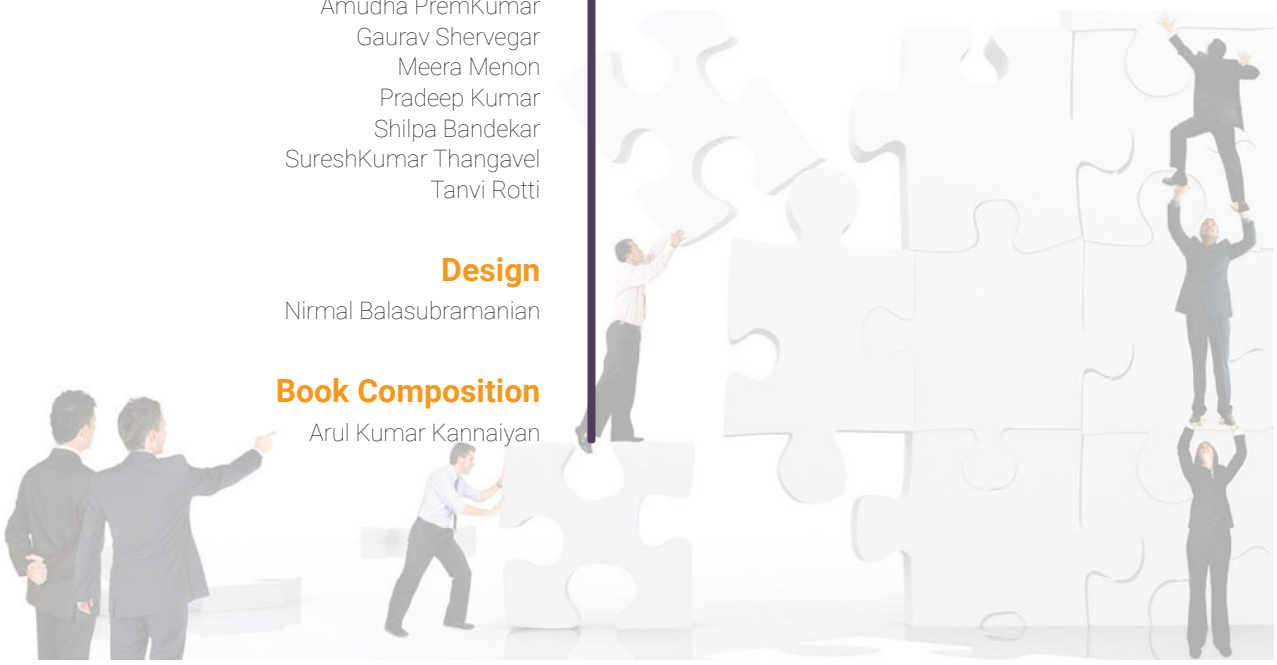
WEB

Angular 2 vs React

Ransomware!

The most profitable malware in the history of cybersecurity!

HOW TO – Manage your IaaS with Apache CloudStack



IDC Tech Journal

Issue 7 – June 2017

Angular 2 vs React

Written by **Sridivya Katha**, PAM, Security Group

A brief comparison of the two front end tools of JavaScript.

When it comes to JavaScript front end tools, one of the frequently asked questions these days is how can one choose between Angular 2 and React JS. Let us discuss the basics.

Angular 2 is a modern and high-performant application development platform supported by Google. It can be used to build significant single page web applications. The Angular 2 framework includes libraries for animation, network access, reactive event driven programming, routing, and component development. It uses dependency injection to wire components together. It is written in **Typescript** language, a superset of modern JavaScript that includes types and annotations. It also includes extensive out-of-the-box support for testing which includes unit, component, and web testing.

React is a web development library written by Facebook and Instagram developers. It focuses on creating web components using HTML and an embedded subset of JavaScript called **JSX**. React is a library and is best suited for developers who want to build their own application platform from the best of the APIs available. For example, it includes libraries for state management, routing, form management, and so on. It is easy to learn because the core API is small.

Syntax

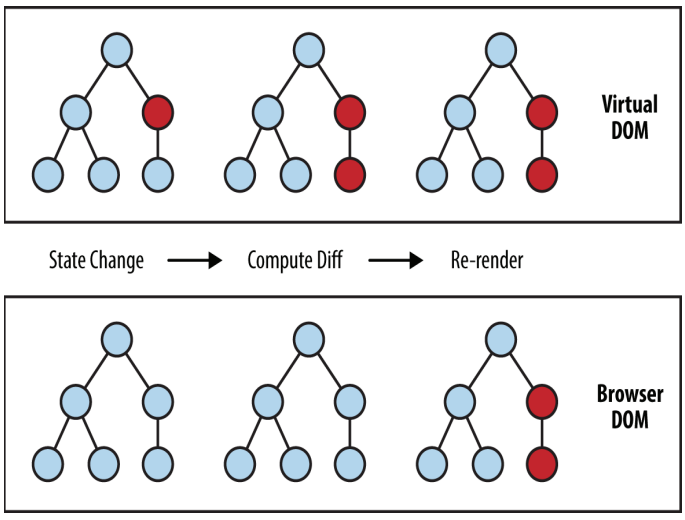
Syntax is the fundamental difference between React and Angular. Angular 2 continues to be HTML-centric rather than JavaScript-centric. Angular 2 puts "Angularized markup" in its HTML templates. React's JSX puts HTML like syntax into JavaScript. This has a massive impact on the development experience. **Angular's HTML-centric design** remains its greatest drawback. JavaScript is far more powerful than HTML. Thus, it makes more sense to enhance JavaScript to support markup than to enhance HTML to support logic. HTML and JavaScript need to work together somehow, and **React's JavaScript-centric** approach is fundamentally superior to Angular's HTML-centric approach. That's the beauty of React, it embraces the power of JavaScript instead of forcing the developers to make an effort to learn unique syntax like "ng-something" for using the framework.

Size and performance

Angular 2, like any other framework includes so many functionalities out-of-the-box which you may or may not use in your app - making it bulkier unnecessarily. Whereas, with React, you have the freedom of choosing the libraries as per the requirements for your application; thus keeping it at the right size without carrying unwanted pieces of code. Also in future, if you ever want to make any changes to your code, you can easily swap the existing libraries with newer and better ones. However, every time a new library is used, the licensing and support concerned with the library has to be taken care separately, as all libraries are not supported by Facebook.

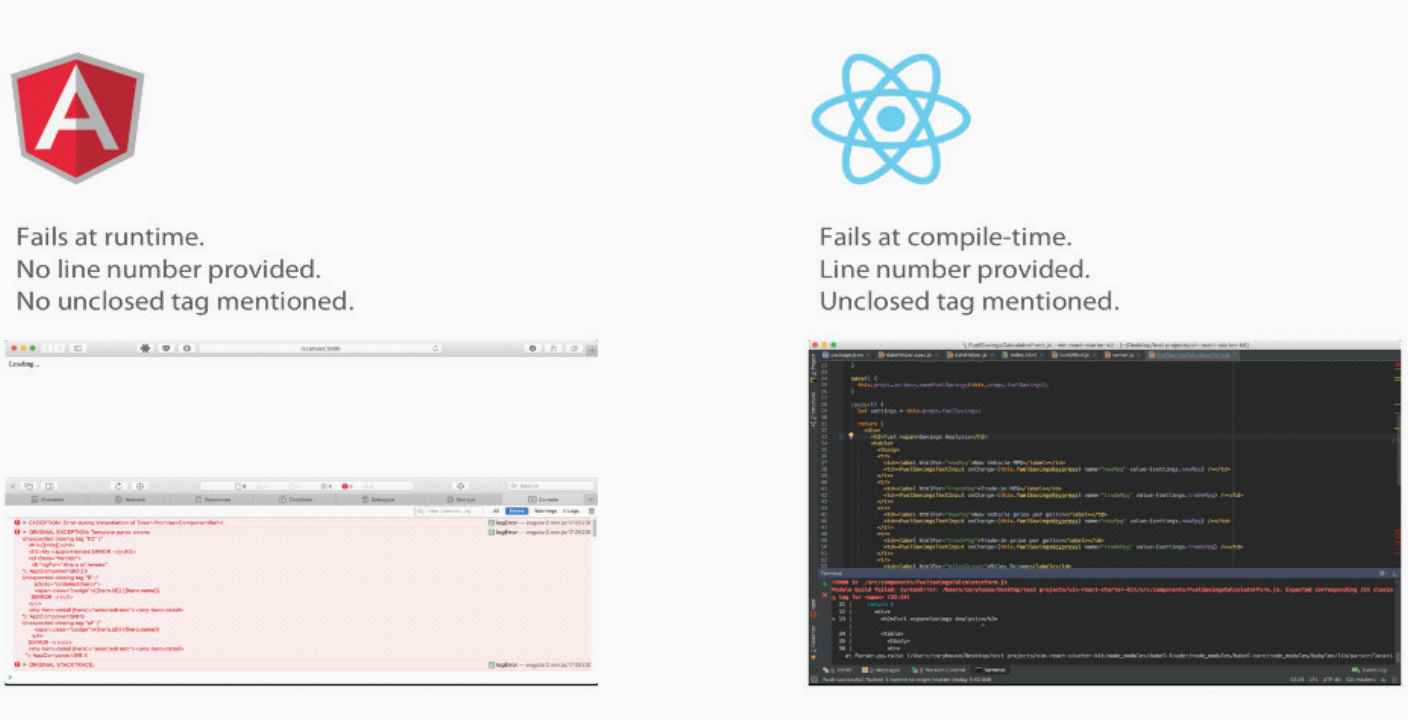
React uses **virtual DOM** for change detection. It also supports static typing that gives a better performance than Angular 2. A virtual DOM is a blueprint or abstract version of real DOM and all the changes are made to the virtual DOM because making changes to the real DOM is an expensive process. Finally, the virtual DOM is compared with the real DOM and only those parts are re-rendered where there is a difference.

Angular 2 also uses **immutable objects** to skip some parts of change detection trees for better performance, but it's not as fast as React.



Developer Experience

With respect to ease of development, there are merits and demerits concerned with both the technologies. React, being a view library, its core does not consist of important functionalities like routing, form management, state management, and so on. It is the responsibility of the developers to pick various libraries off the shelf out of various options available. This process involves lot of decision making. This is the reason you see so many starter kits for React which are pre-loaded with essential libraries.



Since Angular is a framework, it provides more options and functionality out-of-the-box.

In contrast, React wins over Angular when we consider the fact that React fails at compile time unlike Angular which fails at run time. JSX will never compile with a typo. JSX compiler indicates any errors easily by providing the line number of the occurrence of error. This saves a lot of time. Angular, on the other hand, fails at run time. The app will be up and running no matter what and then we need to troubleshoot to find where it went wrong. This slows down the overall development process.

Community and support

React wins the race over Angular 2 with respect to popularity. React is backed by around 936 contributors and 59,301 stars in GitHub community. Angular 2 has 439 contributors and 20,449 GitHub stars. React has far less open issues when compared to Angular 2.

On the contrary, using React has its own shortcomings. It runs on both ES5 and ES6. As a result, some examples of React are presented in ES5 and some in ES6. This leads to much inconsistency in syntax across various examples. So the developers need to be acquainted with both (ES5 & ES6) standards to understand every example they come across. Angular 2 is written in and supports Typescript. Even though it's not mandatory to use Typescript, all the examples in Angular 2 documentation are presented in Typescript. As the core team of Angular itself is embracing Typescript, all the related examples and open source projects are also using Typescript. Hence all the Angular examples across the web are consistent. As a result, developers won't face any issues or confusion with Angular 2.

Summary

	Angular 2	React
Author	Google	Facebook
Language	Typescript	JSX
Learning Curve	Medium	Low
Fails at	Run time	Compile time
DOM	Regular DOM	Virtual DOM
GitHub Stars	20,449	59,301
Contributors	439	936
Open Issues	939	538

Conclusion

As we can see, both Angular and React are great technologies for JavaScript that come up with their own pros and cons. So it is difficult to say one technology wins over the other. How one chooses between these two technologies is highly dependent on the requirement. Those who prefer to code in native JavaScript and do not like the idea of putting the freedom of decision making in the hands of one community, would choose React over Angular 2. Whereas, if the requirement is for a full-fledged framework with out-of-the-box functionalities already in place then, one might choose Angular 2 over React.

Apache CloudStack Cloud Services

Written by **Rajesh KP**, Member Technical Staff 2, Privilege Management

Apache CloudStack seems to be an interesting Cloud services platform using which we could manage our IaaS requirement

We all are well aware of the popular VMware cloud services based on the ESXi hypervisor. It is being used by various teams to set up the virtual machine environment of various Operating Systems required for development and testing. For better management and utilization of resources, these environments are consolidated into Cloud Services with VMware vCenter and vSphere web client. While vCenter works very well, a major issue is its licensing cost! Once the demand for VMs increases due to development and testing needs, the cost of hosting them increases.

To reduce the cost, one of the obvious choice would be by using some free, open source IaaS software. Additionally there will be a need to support existing investment in vCenter. While there are many cloud services such as OpenStack, etc. one of the interesting ones that we found was the "Apache CloudStack". In this article, I will discuss my experiments with this software and how I find it useful in setting up my own IaaS.

Introduction

Currently, we have a VMware based cloud setup with users authenticating to an LDAP server (eDirectory). We need an IaaS solution that works well with our existing investment. Additionally we wanted support for KVM Hypervisor, as our study indicated good performance and stiff competition to VMware.

When we evaluated multiple options such as Eucalyptus, OpenStack, and Apache Cloud Stack, we found that CloudStack is the only one that integrates with vCenter. When we went through the Apache CloudStack documentation, we were excited to see was that it supports various Hypervisors such as ESXi, KVM, Hypervisor, XEN, etc. Since it also supports LDAP based authentication, it rightly fit the requirements we are looking out.

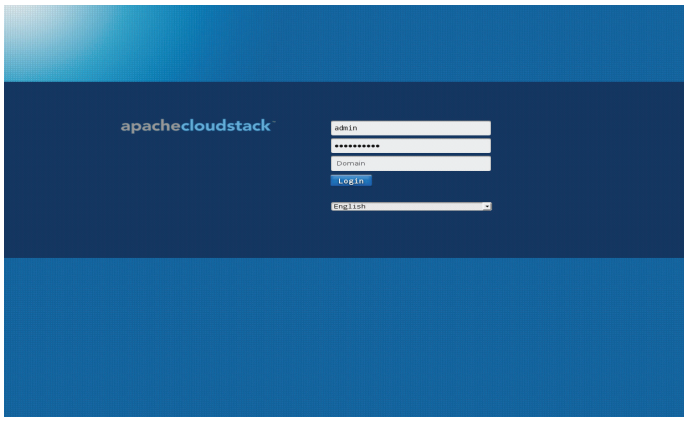
Setup

Some of the takeaways in setting up CloudStack are listed below

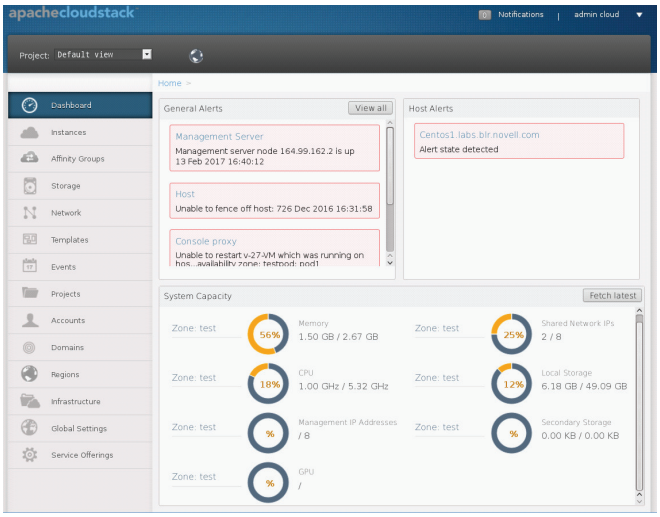
- While we setup CloudStack up on a hardware running CentOS (Install Guide), you can make a quick setup of the CloudStack environment on a VM using DevCloud details.
- While integrating with the existing vCenter cloud, what we noticed is that we cannot add an ESXi Hypervisor server directly into CloudStack but can be done by registering the vCenter server with the CloudStack. So, ESXi Hypervisor management will be done by vCenter itself.
- New Hypervisor additions like KVM can be directly added for management.

Screenshots

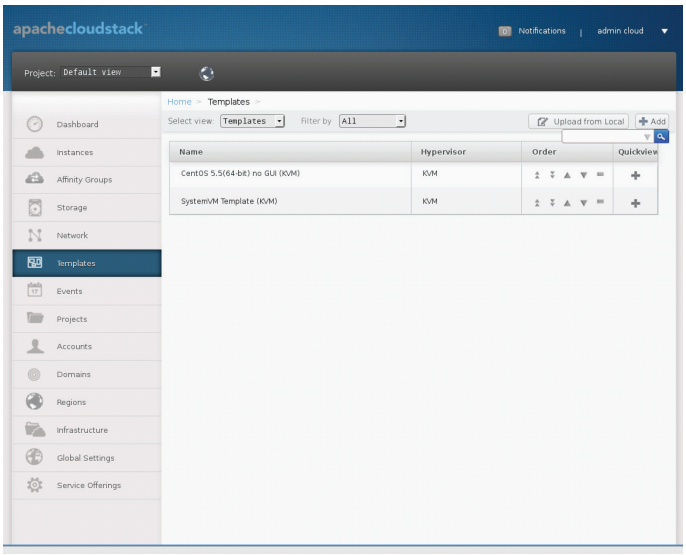
1. The Login screen looked good UI with blue background.



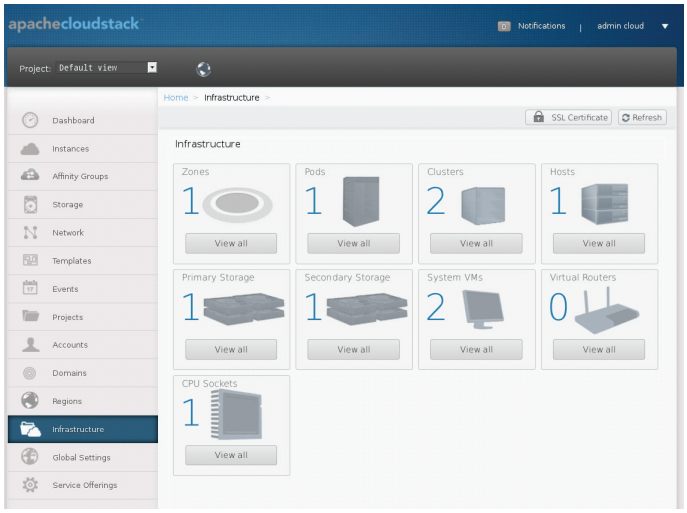
2. Dashboard giving a high level picture of the overall cloud resources.



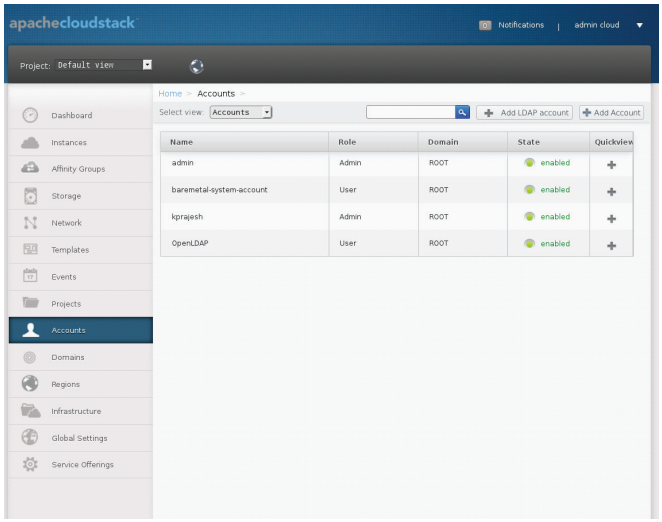
3. We could create VM templates for easier VM creation, couple of default templates are seen here.



4. The Infrastructure view is a very useful view which shows various resources available in the Cloud and we can get into each of their details by clicking the respective tile. From here we could also add/ modify them. For adding a new Hypervisor we need to install an agent on the particular Hypervisor and then add them to the Cloud resources here.



5. The Accounts tab from where the LDAP server can be configured for users. As the Cloudstack comes with some predefined configuration settings for OpenLDAP we setup an OpenLDAP Server to check the LDAP integration. Various advanced settings of the cloud can be done in the Global settings tab.



Conclusion

Apache CloudStack seems to be an interesting Cloud services platform using which we could manage our IaaS requirement.

APR Pools

Written by **Kavya Sinha**, Senior Software Engineer, Access Manager

The Apache Portable Runtime (APR) pools offers an alternate memory management model

As programmers of C, it is our responsibility to manage the resources, such as Memory Buffers and File Pointers. Often this is a tedious and challenging task. The Apache Portable Runtime (APR), a supporting library from the Apache HTTP Server, offers its own memory and resource management mechanism which helps in dealing with memory management challenges. In this paper, we will explore some parts of it. This library can be used by any C program, not necessarily webservers.

Apache uses APR Pools for memory management. Pools are the basic unit of memory management in APR. They allocate memory for the resources either directly or indirectly and ensures that memory is freed up when required.

Drawbacks of Existing Models

In existing models when a memory is allocated one need to track and ensure that the memory is freed once the requirement is over.

Two primary memory management models are:

- **Constructor/Destructor:** In constructor/destructor model, destructors are used for memory clearance. However, in case of shared or conditional resources usage of destructors becomes a tedious process for the programmers. Extreme diligence is required to prevent programming bugs in such cases.
- **Garbage Collection:** In the garbage collection model, memory management task is owned by the language itself, which prevents the program from being vulnerable to programming bugs. But on the contrary, it prevents the programmer from having the control over the lifetime of resources. In addition, it requires all the programming components to be built on same system.

Advantages

Memory Leak Prevention

Apache pools offer its own mechanism to prevent memory leaks. Developer may register any object with a memory pool having a predetermined lifetime. All the resource managed by that pool is released automatically when the pool is destroyed. Allocating memory for the objects having same lifetime in a single pool, allows you to clear the entire pool once the usage of all the objects in the pool are over. For example, objects belonging to same user session can be allocated memory from same pool and the pool is cleared once the user logs out. Programmer need not track memory associated with each object in the session individually.

Easy to test

With few routines responsible for freeing up the memory, it makes identifying memory errors simple and easy. Users can focus on the routines explicitly designed for memory management rather than tracing the entire memory allocation and deallocation process.

Performance

The overhead of allocation, freeing up of memory every other time, and mapping to and fro to virtual memory is reduced to minimal. Instead of allocating multiple small memory chunks a big chunk of memory is allocated to create a memory pool. Further, memory for required resources is allocated from these memory pools rather from the total available memory, thus boosting the overall performance. The destruction of memory-pools clears up the memory of all the associated resources in one go thereby reducing the multiple memory free up requests.

Memory Management

Memory management using APR pool is a three step process. A memory pool is like a session context. At the beginning of a session, you create a memory pool. Then, you create objects in the memory pool during the session. Finally, at the end of the session all you have to do is to destroy the memory pool.

Three basic APIs provided by the APR library that are responsible for memory management are:

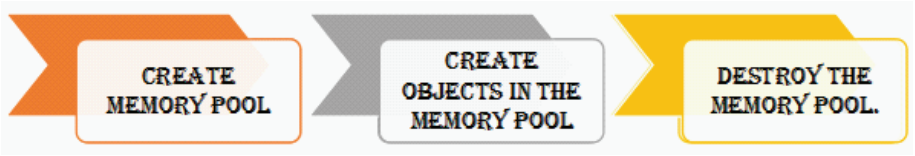
```
apr_pool_create(apr_pool_t**newpool,apr_pool_t
*parent)
```

This allocates a new memory pool from an existing memory pool parent. The memory pool is alive until you call `apr_pool_destroy()`.

```
apr_palloc(apr_pool_t *p, apr_size_t size)
```

This takes the pool and the size of the memory as arguments and it returns a pointer to the memory registered now and ready to use. `ap_pcalloc` can also be used for memory allocation, however this clears out the memory before returning the pointer.

```
apr_pool_destroy(apr_pool_t *p)
```



This destroys the memory pool referred by pointer p.

The memory allocation functions keeps a reference of all the allocated memory to destroy it later. Before memory pool destruction, Apache calls all the callback functions that are registered to the memory pool. Apache does not actually destroys the allocated memory rather it adds the memory to the free memory. You can use the `apr_palloc()` like the `malloc(3)`. Using the `malloc()` calls the need to use `free(3)` for deallocating allocated memory. However, you don't need to free each memory chunk individually in memory pool. Calling `apr_pool_destroy()` for the memory pool once frees all the associated memory.

Debugging

APR offers a list of pool debugging functions. These are implemented when the flag `APR_POOL_DEBUG` is set. Debugging for APR memory pools works by validating the ancestral relationships of data inserted. `APR_POOL_DEBUG` provides functions such as apr_pool_find (to find a pool for resources allocated in it), `apr_pool_is_ancestor` (to determine if a pool is an ancestor of another) and few more.

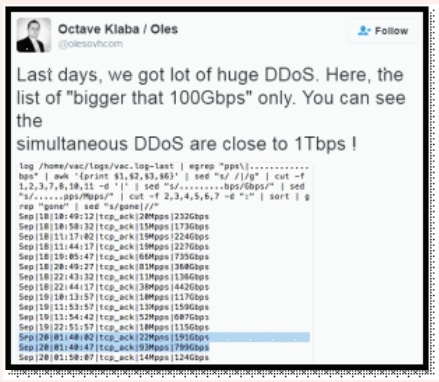
Compiling with debug flag generates a verbose memory allocation log. It is difficult to comprehend logs without tools. There is no tool available for APR till now. However, the logs can be analyzed using the logged tags after filtering the required data.

Conclusion

APR pools offers an alternate memory management model. The basic concept is that whenever you allocate a resource which requires cleanup, it is registered with a pool and the pool takes the responsibility of cleanup. Thus, removing the complexities from applications and reducing programming bugs associated with memory allocation.

World's Largest ever DDoS attack on OVH

(Arun Paul — Sr. Member Technical Staff 1, Security Group)



DDoS attack against enterprises and services has been persistent over the last many years. This includes disruptive attacks on PokemonGo, Battle.net, and so on. The game changer in DDoS attack came in September 2016, when hosting company OVH was hit by an unprecedented (from the weirdest) source. What made this attack stand out was its sheer size which peaked at 1 Gbps and involved 152,000-odd compromised LoT devices including CCTV cameras and DVRs. The tremendous growth of smart devices which includes smart A/C, Fridges, DVRs, CCTVs, smart-lighting and other smart-home devices have often compromised their security over the ease-of-use factor. Lack of proper configuration/installation, default SSL-passwords and lack of security updates makes LoT devices the new focus of emerging attacks.

Capacity Planning for Big Data Applications

Written by **Shammi Kumar Jada**, Sr Member Technical Staff 1, Security Group

Capacity planning for big data clusters in a production environment is a very critical task in IT projects

Hadoop capacity planning should identify the hardware specifications for each workload, including CPU, memory, storage, disk I/O etc. Cloudera displays all this information as charts in Cloudera manager console page.

Hadoop cluster planning describes general overview of hardware configuration needed for Master and Slave (worker) nodes in the cluster. It also helps administrator to tune Hadoop clusters to get maximum performance.

This paper describes sizing or capacity planning for big data components to be deployed in Hadoop cluster. Additionally, it discusses how to tune big data components in the cluster for better performance and scalability.

Introduction

What is the configuration of each node in your Hadoop Cluster?

Hadoop distributes data across a cluster of balanced machines and uses replication to ensure data reliability and fault tolerance. Because data is distributed on machines with compute power, processing can be sent directly to the machines storing the data. Since each machine in a Hadoop cluster stores and processes data,

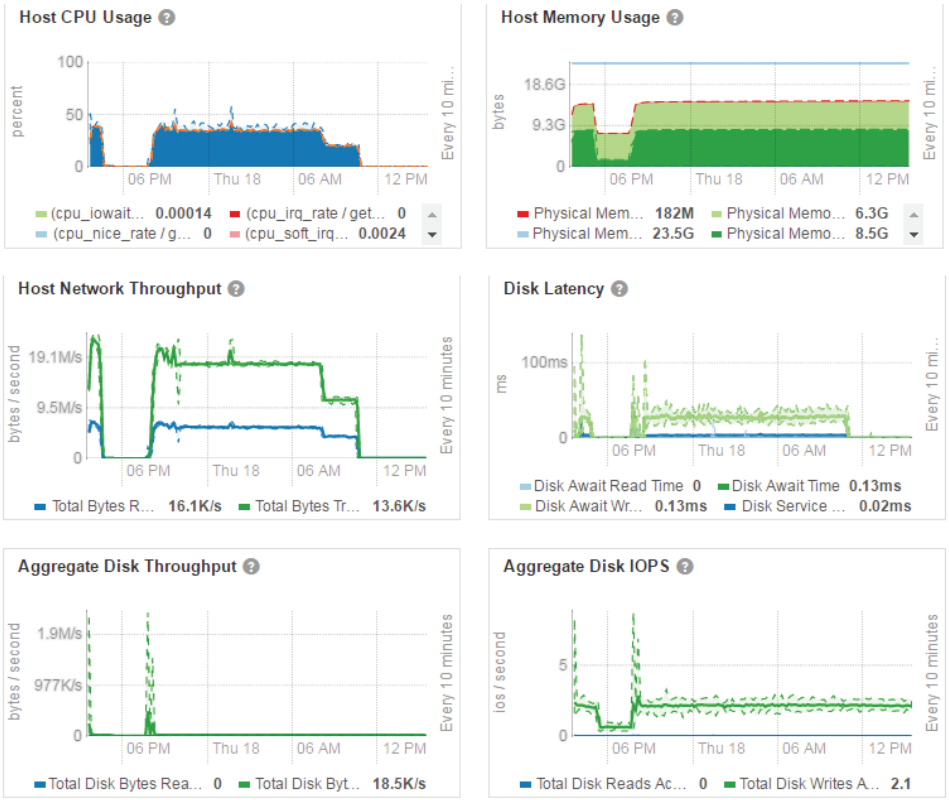


Figure 1: Resource utilization of a node in Cloudera Manager

those machines need to be configured to satisfy both data storage and processing requirements.

1. Plan Hadoop cluster sizing for your workloads:
   - Identify whether the workload is IO or CPU bound in your application
   - Count the number of cores to be used in the cluster based on the workload
   - Amount of memory used by the cluster based on the workload
   - How much data per day the application is storing in the cluster
   - Amount of read and write operations in the cluster
2. Focus on resource management of your workloads:
   - Benchmark worker jobs running on the balanced cluster to analyze how they are bound. To achieve that goal, it is straightforward to measure live workloads and determine bottlenecks by putting thorough monitoring in place.
   - Installing Cloudera Manager on the Hadoop cluster to provide real-time statistics about CPU, Memory, disk, and network load.

Choosing Hardware for CDH cluster configuration

There are four types of roles in a basic Hadoop cluster: *NameNode* (and Standby NameNode), *JobTracker, TaskTracker, and DataNode*. A node is a machine performing a particular task. Most machines in production cluster will perform two of these roles, functioning as both DataNode (for data storage) and TaskTracker (for data processing).

1. You may require to deploy the following production cluster component nodes in a big data infrastructure.
   - *Worker Nodes*: HDFS Data Nodes, HBase Region Servers, Yarn Node Managers
   - *Master/Manager Nodes*: CDH Services, Spark History Server, Yarn Resource Manager, HBase master, HDFS Name Node
   - *Messaging Nodes:* Kafka Brokers
   - *Indexing Nodes*: Elastic Search, Solr etc.
2. Configure the storage for your cluster which is one of the most important initial steps. Choose *LVM, RAID or JBOD* (Just a Bunch Of Disks) configurations for storing long term big data. *JBOD is recommended for production cluster nodes.*
3. Consider total storage capacity (number of disks) on each node based on *replication factor and high availability* of the CDH components.
4. Keep worker component node's configuration is homogeneous always and deploy rest of the components node's configuration heterogeneously in the cluster. More the nodes, performance will be better.
5. Here are the recommended specifications for Worker/ Messaging/Indexing nodes in a balanced Hadoop cluster:
   - 12-24, 1-4TB hard disks in a JBOD (Just a Bunch Of Disks) configuration
   - 2 quad-/hex-/octo-core CPUs, running at least 2-2.5GHz (enable hyper-threading)
   - 64-512GB of RAM
   - Bonded Gigabit Ethernet or 10Gigabit Ethernet (the more storage density, the higher the network throughput needed)

     Here are the recommended specifications Master/Manager nodes.
   - 4–6, 1TB hard disks in a JBOD configuration (1 for the OS, 2 for the FS image [RAID 1], 1 for Apache Zookeeper, and 1 for Journal node)
   - 2 quad-/hex-/octo-core CPUs, running at least 2-2.5GHz (enable hyper-threading)
   - 64-128GB of RAM
   - Bonded Gigabit Ethernet or 10Gigabit Ethernet

     Note: The drive count above will fluctuate depending on the amount of redundancy (replication factor)
6. When the Hadoop cluster in place, start identifying workloads and prepare to benchmark those workloads to identify hardware bottlenecks. After some time benchmarking and monitoring, we can understand how additional machines should be configured heterogeneously in the cluster.

How to tackle performance challenges in Big Data Cluster sizing

Performance is a key challenge. Whenever the volume or velocity of data overwhelms current processing systems/techniques, resulting in performance that falls far short of desired.

Following are the challenges faced:

- Variety of data, veracity of data
- Workload pattern, is it CPU or IO bound?
- OS, kernel tuning and disk swapping
- Cluster sizing
- Disk sizing and configuration, replication
- Network bandwidth between cluster nodes

Following are the two approaches to improve the performance in big data scaling magnitude:

- Scale up (vertical scaling)
- Scale out (horizontal scaling)

Scaling up: refers to purchasing and installing a more capable central control or piece of hardware. For example a scaling up

Node Type (Virtual Machines)	VM Nodes	vCPU per Node	vMemory (GB) per Node	Disks per node (3x replication)
Worker (Hdfs Data Node + HBase Region Server + Yarn Node Manager)	3	12	24	4 x 3
Messaging (Kafka Broker)	3	4	32	3 x 3
Indexing (Elastic Search)	2	8	12	4 x 3
Master/Manager (CDH Services + Yarn Resource Manager + Job Tracker + HDFS Name node + Hbase master)	1	16	32	1 x 3

Table 1: Scale out proposal – Virtual Machines (Horizontal approach)

Node Type (Bare-metal)	Hardware (Dell PowerEdge R730) Nodes	CPU per Node	Memory (GB) per Node	Disks per n (3x replica
1x Worker 1x Messaging 1x Indexing	2	24	128	12 x 3
1x Worker 1x Messaging 1 x Master/Manager	1	24	128	9 x 3

Table 2: Scale in proposal – Bare-metal Servers (Vertical approach)

Component	Recommendation
CPU	Two Intel(R) Xeon(R) CPU ES-2695 v2@ 2.40GHz
Data Storage	1–4 TB hard disks in JBOD (Just a Bunch Of Disks) configuration SSD or 15000 RPM SATA for Elasticsearch nodes 7200 RPM SATA for Kafka and HBase nodes
RAM	16–128 GB
Network Technology	Bonded Gigabit Ethernet or 10 Gigabit Ethernet

Table 3: Hardware Requirements for CDH and Elasticsearch Data Nodes

CDH component	Number of data nodes	Number of vCores (per node)	RAM (per node)	Number of disks (per node)	Recommended number of disks (including replicas)	Storage per day (1x)	Recommended storage per day (including replicas)
Worker	3	12	24 GB	4	4 x 3	365 GB	365 x 3 GB
Messaging	3	4	32 GB	3	3 x 3	570 GB	570 x 3 GB
Indexing	2	8	12 GB	4	4 x 2	260 GB	260 x 2 GB
Master/Manager	1	16	32 GB	1	1 x 1	50 GB	

Table 4: System Sizing Information for 10000 EPS in Sentinel scalablestore (8.0)

approach would be to buy a more capable server with more processing capacity (CPU, RAM and Disks). So that each node in the Hadoop cluster contains combination of services running within the same node.

Scaling out: means linking together other lower-performance machines to collectively do the work of a much more advanced one. You install multiple lower-end configuration nodes in the cluster when you create Hadoop/CDH clusters. You can later scale out the cluster by increasing the number of worker nodes and client nodes.

Plan your nodes to scale in Hadoop cluster either horizontally or vertically. For example, following are the two different hardware layout of services proposed for Sentinel product to scale 10000 events per second (EPS) with horizontal (9 nodes) and vertical (3 nodes) approaches.

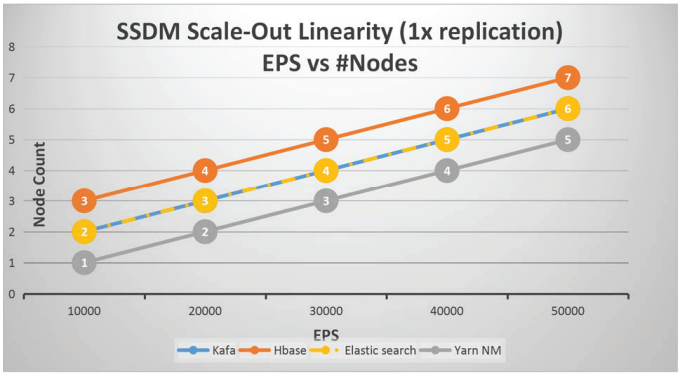


Figure 2: Events per second (EPS) vs. number of Nodes in CDH Cluster

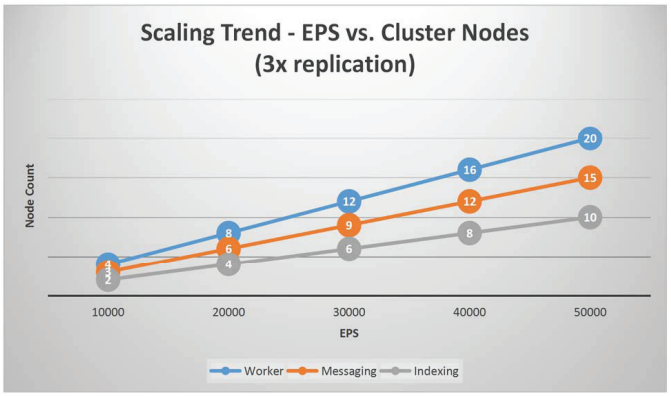


Figure 3: Events per second (EPS) vs. number of Nodes in CDH Cluster

Tuning Guidelines for CDH cluster		
CDH Component	Scaling Factor	Recommendation
Kafka	Data retention hours *log.retention.hours*	Configure the retention days based on the disk space available on the Kafka node, EPS rate, and the number of days needed to recover the system from a scalable storage outage if one were to occur. Kafka will retain the data while the scalable storage system is recovered, preventing data loss. For example, at 10K EPS rate in sentinel, Kafka needs 250 GB per day to store data.
	Data directories *log.dirs*	Each directory should be on its own separate drive. Configure multiple paths to store Kafka partitions. For example, /kafka1, /kafka2, and /kafka3. For best performance and manageability, mount each path to a separate physical disk (JBOD).
HDFS	HDFS block size *dfs.block.size*	Increase the block size to 256 MB to reduce the disk seek time among data nodes and to reduce the load on NameNodes.
	Replication factor *dfs.replication*	Set the value to 3 so that it creates three copies (1 primary and 2 replicas) of files on data nodes. More than 3 copies of files require additional disks on data nodes and reduce the disk I/O latency. The number of data disks required is usually multiplied with the number of replicas.
	DataNode data directory *dfs.data.dir,dfs.datanode.data.dir*	Each directory should be on its own separate drive. Configure multiple paths for storing HBASE data. Example /dfs1, /dfs2, and so on. For best performance and manageability, mount each path to a separate physical disk (JBOD).
	Maximum number of transfer threads *dfs.datanode.max.xcievers,dfs.datanode.max.transfer.threads*	Set it to 8192.
HBase	HBase Client Write Buffer *hbase.client.write.buffer*	Set this value to 8 MB to increase the write performance of HBase for a higher EPS load.
	HBase RegionServer Handler Count *hbase.regionserver.handler.count*	Set this value to 100. This value must be approximately the number of CPUs on the region servers.For example, if you have 6 region servers with 16 core each, set the region handler count as 100 (6 x 16 = 96).
YARN	Container Virtual CPU Cores *yarn.nodemanager.resource.cpu-vcores=14*	Increase this value to up to 80% of NodeManager's available vCPUs.For example, If NodeManager has 16 vCPUs resources, set this value to up to 14 vcores so that you leave 2 vcore for operating system utilization. You can increase or decrease this value based on the available vcores for NodeManager.
	Container Virtual CPU Cores Maximum *yarn.scheduler.maximum-allocation-vcores*	Allocate 1 vcore per Application Master container.
	Container Memory *yarn.nodemanager.resource.memory-mb*	Increase this value to up to 80% of NodeManager's memory. For example, if NodeManager memory is 24 GB, set this value to up to 20 GB.You can increase or decrease this value based on the available NodeManager memory.
	Container Memory Maximum *yarn.scheduler.maximum-allocation-mb*	For example Spark runs 3 applications: event data, raw data, and event indexer. Ensure that each AM container has sufficient memory to run these applications. For example, if YARN ResourceManager has 24 GB, allocate a maximum of 8 GB memory per AM container. You can increase or decrease this value based on the available ResourceManager memory.
Disk Latency		Whenever the disk latency goes beyond 100 ms on a data node, add more JBOD disks to the data node and configure the component causing the highest I/O load to utilize the directories where additional disks are mounted.

Case Study - Cluster Sizing for Sentinel 8.0 (scalablestore)

Following is the hardware requirements proposed for Sentinel Scalable storage Data Manager (SSDM).

CDH and Elasticsearch Cluster Configuration

The following list describes the minimum recommended cluster configuration for Sentinel scalablestore:

- Worker Nodes (Hdfs Data Node + HBase Region Server + Yarn Node Manager): 3 node cluster for 3x replication
- Messaging Nodes (Kafka Broker): 3 node cluster for 3x replication
- Indexing Nodes (Elasticsearch): 2 node cluster for 2x replication
- Master/Manager (CDH Services + Yarn Resource Manager + Job Tracker + HDFS Name node + Hbase master): 2 node cluster for high availability

Sentinel Scalable Store Data Manager (SSDM) Scale-Out Performance Results (with the hardware used in Table 1)

Performance Tuning Guidelines - CDH

The following table provides information about performance tuning recommendations that you must perform on your Cloudera setup. This tuning provides material performance benefits for configuring CDH cluster components in big data. For information about how to set these values, refer to the Cloudera documentation.

Conclusion

Purchasing appropriate hardware for a Hadoop cluster requires benchmarking and careful planning to fully understand the workload. However, Hadoop clusters are commonly heterogeneous and Cloudera recommends deploying initial hardware with balanced specifications when getting started. It is important to remember when using multiple ecosystem components resource usage will vary and focusing on resource management will be- your key to success.

This paper encourages you to tune your CDH components and effectively design/planning and sizing your hardware for big data infrastructure.

bits & bytes

IOT for Enterprise

The obvious use cases for IoT devices are energy savings, security, and convenience. Most people think IoT devices is a gimmick until they try it and then they are surprised at how useful it is. IoT is also adopted in business contexts. For example, devices with sensors and network connections are extremely useful for monitoring industrial equipment. Here are some of the enterprise examples which are adopting IoT

Internet of Whiskeys: Johnnie Walker
Making the Blue label Whiskey smart, adding connected technology to each bottle, which would send messages to smart phone when bottle is opened . . .)

Internet of Disney: Walt Disney World
Created a “MagicBand” a wristband, using which visitors can take rides, buy food etc. Disney would use this data to track the movements and provide best experience to the visitors.

Internet of Cows: BT
BT monitors the cows location in order to prevent theft.

Internet of Gas: British Gas
Enabling integration with smart energy applications, able to control heating and hot water remotely as well as a holiday mode to save power.

Cloud Access Security Brokers

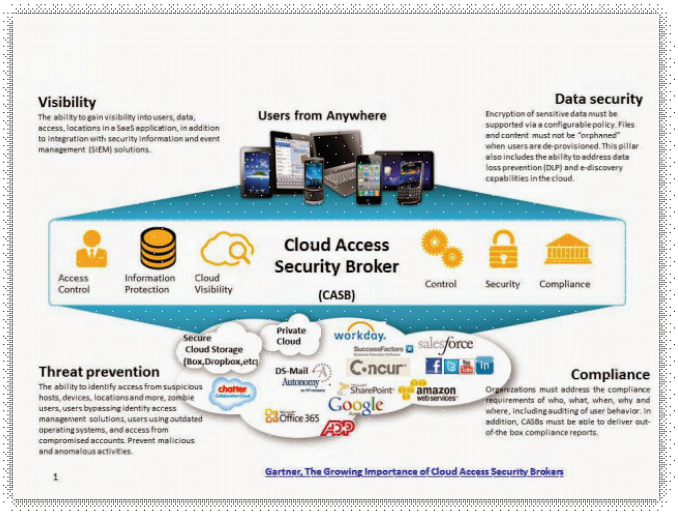
Written by **Arun Paul**, Sr. Member Technical Staff 1, Security Group

Gartner™ rates Cloud Access Security Brokers (CASB) as the number one emerging security technology of 2016. Gartner is yet to rate the various CASB competitors in their famous quadrant rating system

Cloud Access Security Brokers are on-premises, or cloud-based security policy-enforcement points, placed between cloud service consumers and cloud service providers to combine and interject enterprise security policies as the cloud-based resources are accessed – Source: Gartner™

How is CASB different from traditional security deployment?

We have been living in an infrastructure-centric network till the time Cloud-services became widely adopted. With traditional infrastructure, we had complete control over the devices in which services are deployed. Our traditional security policies and enforcement were based on such an environment. Now, that the services are moving to cloud, we still have the necessity for policy enforcement and security deployment, but we don't have a direct control over it anymore. In such a flexible-perimeter environment, we need an integration point for Security and Policy enforcement. CASB provides the same level of enforcement among unmanaged devices and cloud services. When you compare to traditional network security devices, a CASB provides the same choke-point in a cloud based deployment that a Firewall used to give in a traditional network infrastructure. CASB is so powerful that it acts like a singular gateway for an organization's multi-vendor, multi-architecture cloud-integrations requirements, simplifying the policy enforcement and management over cloud.



Identity, sign-on, credential-mapping, and other access-control technologies need to have the same granularity over cloud deployed services. CASB needs to be flexible enough to handle such offerings which includes simpler out-of-the box services and also more complex SDK based development solutions for its clients. CASB providers need to have deep understanding of the cloud deployed services of all possible scenarios and distinguish how data is handled in the cloud from the data-in-motion and data-at-rest perspective and provide distinct inline control over it.

Gartner™ defines the four pillars of CASB as

- Visibility
- Data Security
- Threat prevention
- Compliance.

More than protecting the application and the underlying network infrastructure, which can be done via cloud based firewalls/IDS/IPS, a CASB technology focuses on the 'data' and the 'users' in the cloud. This is significant as the enterprise is predominantly responsible for its data and users more than the cloud-service provider. CASBs, hence becomes one of the most important intelligence-gathering point from a SIEM-monitoring perspective.

Customer Journey Maps

Written by **Archana Tiwary**, Information Developer, Access

Understanding different types of user and providing the specific information is the crux of a successful solution: starting from a product/ service to user documentation

One of the techniques in understanding users is by creating documents that explain the role of target users and how they use the product. We call these documents as User Persona Documents. Understanding the perspectives from various User Persona Documents, you can look from a holistic viewpoint at how customers are using the product, what is their requirement, what they want to achieve, what improvements they want in the product. However, we cannot capture everything related to customer use case only by having User Persona documents. In this paper, I will highlight about one more complementary tool/technique called Customer Journey Map, which highlights the customer experiences and how it serves as a good communication tool.

Customer Journey Map is the process of tracking experiences (good, bad or indifferent) and corresponding responses that customers come across while using a product or a service. Mapping the journey builds knowledge and agreement across stakeholders. You can use the map as an artifact to create and support better customer experiences.

Using Customer Journey Maps, you can break down the journey of each specific persona into different phases and goals by putting yourself into customers' shoes. The different personas may have different goals. So, for each new feature, you need to create a separate map for each persona.

Key Terms

Let us understand few key terms we are using in this document

Touch Points: Actions and interactions of a customer with the product.

Channel: The medium of interaction. For example, through website, mobile device, or person-to-person

For example, the touchpoint could be "logging into the portal", and the channels associated with that touchpoint could be "Internet Explorer on a desktop" or "Chrome on a mobile device".

When to Create It

You should consider creating a Customer Journey Maps in the following scenarios:

- When the cost of the product is high
- When the product is complex
- When the product in not stable
- When the expectations from the product are high
- When gaps exist in your understanding of the experience that customers face

How to Create It

You can create a successful Customer Journey Map in many ways. You can find a number of templates online. You can either use an available template after customizing it based on your requirements or you can develop a template. You must create Customer Journey Maps in the initial stage of the product development.

The following are the steps that some of the teams in our organization have already implemented:

1. Establish the types of customers (user personas) of your product.
2. Identify the key journey steps (touch points) from end-to-end of a customer.
3. Identify actions/thoughts at each step.
4. Plot the expectations customers have of the product at each of the journey steps in the journey map.
5. Identify how the product would meet these expectations. Are there any gaps? Can any step be removed?
6. Identify points in the customer interaction where things sometimes or often go wrong or what are the top pain points. Describe these roadblocks or forks in the road. Identify areas in the journey that require improvement.
7. Document your customers' emotional response at every step of their journey.
8. Analyze the potential opportunities for customer service improvements. Think about what could make it better?
9. List specific actions to make these improvements and to innovate.
10. Brainstorm the map with the team internally for any additional information.
11. Get your customers feedback.

You should also state the product-specific terms and assumptions, if any, in the Customer Journey Map doc.

The following is a snapshot of the template that some of the teams have implemented in our organization:

1	User Journey:	<Product Name>					
2	Story Name	<Name of the story>					
3	Customer Information	<Persona details and motivation>					
4		Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
5	Key Journey Steps						
6	Actions/Thoughts at each step						
7	What do you want?						
8							
9	Could this step have been avoided?						
10	What could make it better?						
11							
12	Questions						
13	Notes						

A sample journey map:

2	Story Name	Handling the changes requested				
3	Customer/user Information	The administrator has to review the action items, resolve any conflicts, and and fulfill the change requests.				
4		Step 1	Step 2	Step 3	Step 4	Step 5
5	Key Journey Steps	**Gets notification that I need to intervene as a result of a change**	**Look at a list of requests that need fulfillment**	**Change profiles in collected system to match requested changes in**	**Disable or Remove profile in collected system**	**Refuse fulfillment order**
6	Actions/Thoughts at each step	Add the task section in the UI	I want to be able to see what changes were requested as a result of the review, regardless of whether I need to manually make any changes to the profiles.	If the profile was changed in the review process or a change was requested (to assign to a user), I need to be able to correct this value in the enterprise system to match the value expected from the review.	I have a request for profile removal, but I am not able to remove the profile. I need to be able to either mark the profile disabled and comment why I can't delete it.	I cannot remove a profile due to some business or technical reason. I need to provide an explanation as to why the profie was not removed, so that the requester can see the details about what happened with the fulfillment.
7	What do you want?	I need to know that there are profile changes that I need to make.	I want to be able to see profiles within my domain of authority that I need to change.		Remove or disable profile in the collected system (e.g. Active Directory) and mark fulfillment item completed.	Refuse the fulfillment order and indicate reason. This order will fall off my todo list.
8	Could this step have been avoided?					
9	What could make it better?					Should have the option to set required comments on actions to ensure that the fulfillment process follows business policies.
10	Notes			Correction of the profile in the enterprise system is a manual action.		

Who Should Create It

Senior members of the team who are customer facing such as Architect, Footprint Engineer, Product Support Lead, and Product Owner are the ideal owners of Customer Journey Maps.

However, after the first draft of the map is ready, it should be discussed/reviewed with all stakeholders.

How to Maintain It

Customer Journey Map is a dynamic document. It may change over time based on the customer feedback. You must use a content versioning tool that is accessible to concerned stakeholders for keeping the track of change history.

Conclusion

In a nutshell, creating a Customer Journey Map helps in the following areas:

- Understand the customer's experience when using our products
- Identify typical UX problems that the customer encounters
- Enable to plan for future solutions and enhancements
- Prioritize competing projects
- Enhance decision-making and content design
- Provide a foundation for the test plan of QE and content plan of InfoDev
- Bring different teams together for better customer experience
- Deliver the product at the most important time
- Highlight the areas where actions need to be taken most urgently

Driving Product Security with Continuous Integration

Written by **Kalyan Juthada**, Sr Member Technical Staff 1, Micro Focus SG Group

Security team should evaluate right set of tools and integrate as part of continuous integration process and ensure they work seamlessly together

Recent developments in DevOps led to adopting of continuous integration process within Software Development Life Cycle. Continuous Integration (CI) helps teams to create builds with ease and keep track of build jobs for testing and deployment. Common Open Source tools like Jenkins are most used by Build Engineers or Software Groups. Unit Test and Functional Test Automation suite are integrated within the CI process.

With increasing number of data breaches in IT Industry, software security is becoming prominent and critical for software development teams. Before Secure Development Lifecycle (security is considered as a priority within IT Teams only . Software development teams have now adopted secure development and testing within SDLC.

For attaining true agility and ensuring security within Agile teams, development teams should integrate security tools within CI to receive early feedback and assess software security based on the baseline derived from earlier builds in a particular release cycle.

The following workflow depicts the secure development practices adopted within continuous integration process:

Components with Known Vulnerabilities

Most Enterprise Software solutions contain third party components which might require updates to ensure existing versions are not vulnerable. Manual and automated methods are followed by development teams to review vulnerable component libraries and dependencies.

This is part of OWASP Top 10 “Using Components with Known Vulnerabilities”. Embedding vulnerable libraries is huge risk due to new exploits available once a known vulnerability exists with an embedded component. It is often easier for attackers to exploit a common third party component, than the application itself. Most of the time development teams audit libraries at the beginning of aproject and continue with development activities to ensure software is delivered on time.

With the recent increase of security research in open source libraries, Development teams should check periodically for vulnerable libraries through vulnerability scans or build time analysis. To be more effective automated review by tools can be integrated with CI.

OWASP recommends tools like OWASP Dependency Check & OWASP SafeNuGet for checking dependencies and libraries on Java or DotNet based projects. Dependency check tool is run during build time to identify vulnerable dependencies used within the project. You can use the command line utility to check target libraries on installed software.

Static Code Analysis

Static Code analysis is an automated source code review process to identify vulnerabilities during build time. The development team uses IDEs to identify vulnerabilities while developing code to detect issues early for resolution. These tools also discover best practices in coding to be followed during implementation.

Some of the common tools used in Enterprise are HP Fortify, CheckMarx, SonarQube, Find Bugs etc. Based on the budget, teams need to evaluate tool of their choice for target language types.

Few tools can be integrated directly with continuous integration. Jenkins Plugin or custom scripts can be developed to run remotely on build machines.

HP Fortify tool covers most of the software languages. Static Code Analyzer (SCA) which scans the source code and upload the results Software Security Center(SSC). SSC supports Administrator to manager users, roles and reports.

Initial run of Source Code Analysis would generate more number of issues which require security issues to be filtered out for triage (for example , HP Fortify provides list of issues by Critical/Medium and Low which can be further filtered by Security issue type, best practices etc.

Security Activities Monitored	Total Issues Discovered	New issues detected (Based on previous run baseline)	No. of False Positives	Fixed or Found Vulnerabilities
Static Code Analysis by Module OWASP Dependency Check during Build Vulnerable Libraries Network Vulnerability Scan Web Vulnerability Scan Fuzzing				

Generally static code analysis tools find more false positives which needs to be managed on target tool while triaging For example, the HP Fortify tool allows users to suppress the false positives, so that invalid issues are not reported during rescan.

Tools which do not have this functionality should maintain internal database of vulnerabilities found and mark them appropriately.

Frequent vendor updates should be performed so that new issues are discovered

Code analysis reports from continuous integration jobs should be filtered and normalized so that development team focuses on the true positives. Previous results can be used as a baseline to derive delta issues between different builds or different releases.

All Good, but say what tools you use, and how well they work!!!

Vulnerability Scans

Vulnerability Scans are run against installed software to find network and web vulnerabilities. This activity is done when build passes automation sanity.

Network Vulnerability Scan

Network vulnerability scan finds missing OS patches, insecure configurations, and vulnerable applications installed, SSL weakness etc. Common Enterprise tools are Nessus & Qualys etc.

Custom scripts should be written as part of continuous integration jobs to interact with remote vulnerability scanners in running scans on target test environments followed by automated sanity test.

Web Vulnerability Scan

Web vulnerability scan finds common web-based vulnerabilities on web console exposed by Enterprise solutions.

Common web vulnerability scanners are Burp, OWASP ZAP, and HP Web Inspect etc.

Scans are conducted in the following ways:

- Directly where tool crawls the pages from parent URL and finds out abnormalities.
- Capture the HTTP traffic (proxy based) while running automated functional tests (UI based) and run web vulnerability scan on the captured HTTP requests.

Other Tools

- BinScope Binary Analyzer – Microsoft tool to analyze binaries on a project to check compliance with Microsoft SDL, mainly that your product is compiled with the right switches to provide protection (ASLR, NX, etc).
- Fuzzing Tools – Tools to fuzz software interfaces with random or malformed data and identify any potential security issues. For example: - File fuzzing, Rest fuzzing etc.
- Attack Surface analyzer – Microsoft tool to detect changes made to operating system when target software is installed.

Reporting

Continuous Security charts can be created by consolidating and processing individual test results of security activities. Metrics derived from these charts helps any organization to assess the Security posture of the solution being developed.

Conclusion

Security team should evaluate right set of tools and integrate as part of continuous integration process and ensure they work seamlessly together.

This reduces the effort in tracking the activities separately. A release baseline is considered to find out delta issues for code analysis and vulnerability scans.

Overall an automated process will help organizations to generate software security metrics and comply with internal SDLC Process.

References

1. https://www.microsoft.com/en-us/sdl/
2. https://www.owasp.org/
3. https://en.wikipedia.org/wiki/Continuous_integration

Engineering Best Practices

Written by **Mahantesh Hongal**, Member Technical Staff 2, Security

We implemented scrum in several teams in NetIQ, it worked fairly well for us and our customers015 . . .

An engineering process involves understanding and applying the engineering methodologies in a disciplined manner. The Agile Methodologies have made a deep impact in the industry since the last decade. Though it is not formally standardized, it is globally accepted. According to VersionOne (http://www.versionone.com/) survey, the percentage of people who plan to implement agile methodology for development projects has increased by 83% in 2012.

Why are people adopting agile methodologies over traditional methodologies and what are its main paybacks?

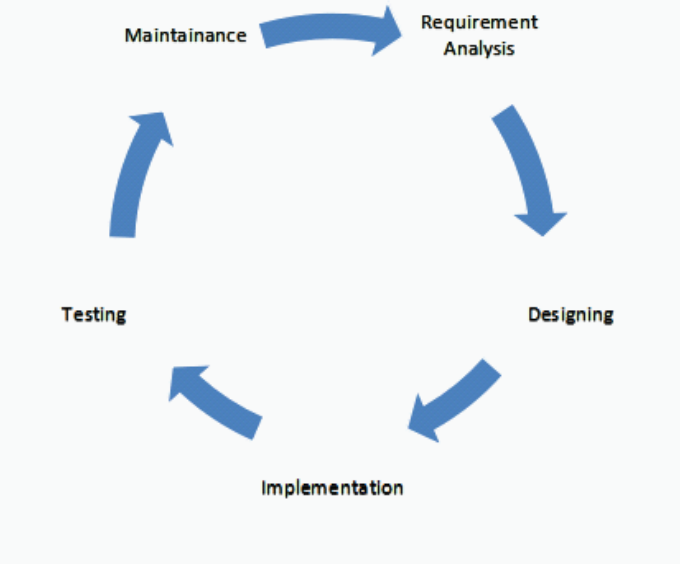
Agile methodologies not only address some significant problems with the traditional software development methodologies, but also provide the following important benefits:

- Adaptive to the change
- Increased productivity
- Effective resource utilization
- Improved visibility
- Improved team morale
- Enhanced software quality
- Reduced risk
- Faster time to market
- Improved engineering processes
- Faster process improvement

Introduction

Software engineering methodologies involve the knowledge, tools, and methods for gathering requirements, designing, developing, testing, implementing, and maintaining software. Software engineering methodologies helps the team to get a complete understanding of the entire project. Implementing and delivering large projects successfully involves teamwork and team co-ordination. We need a process which helps in team co-ordination and collaboration to deliver fine quality software as per the customer requirements.

A Software Engineer applies techniques and practices from the Computer Science and Project Management domains to improve the SDLC quality.



Problem definition

We already have numerous engineering techniques, methodologies and practices in place, then why do we need one more? Essentially, we need new engineering techniques to overcome the shortfalls of the current engineering methodologies. Most of the traditional engineering methodologies are Component centric, i.e., Component Driven Development (CDD), which leads to the following common problems of the traditional methodologies:

Freedom

Traditional methods do not give a lot of freedom to individuals in making decisions regarding selecting the tools, techniques and individual procedures. This might lead to poor quality of the software. Teams are organized according to the traditional methodologies. In traditional methods, the process is defined before we even start solving the actual problem, so the process may not suit the customer problem.

Communication and collaboration

Communication and collaboration is not part of the SDLC, but it is very important in delivering high quality products and in meeting and fulfilling customer requirements. In traditional methods, we don't see much communication and collaboration with customers except in the beginning requirements gathering stage. This gap leads to the problem in delivering the actual customer needs.

Requirement Gathering

Traditional methods always fall short in being adaptive to constantly changing customer requirements. Main reason for this could be because customers may not be able to visualize complete software at an early stage. So, the requirements will not be much clear and complete.

Another problem is sometimes customers will not be fully satisfied with the deliverables as the product is implemented totally based on the documented requirements and not as per customer's actual needs. Customers only get to see the product at the final stages or after the delivery.

Risk

A major risk will be discovered at the last stage of the SDLC, which increases the cost of re-doing and sometimes even the cancellation of project. This happens mainly because of the communication gap between the customer and the development team and the unclear requirements.

Visibility

In traditional methodologies, we don't get much clarity about the direction in which the product development is heading. In the initial cycles, we may not get full clarity and visualization of the product. Lack of internal visibility leads to the last minute risk discovery.

Adaptive to the change

In traditional methodologies, we work on a fixed plan which is not flexible to the changes and feedback. Even a small change can cause a major impact because it needs to be changed in all the phases of the SDLC.

Proposed solutions

The current market demands rapid developments and deployments which we cannot deliver with the traditional methodologies because of the issues we talked about in the above section. The Agile software development methodologies are producing the proven track record of successful rapid development and deployments since the last decade.

The Agile software development is a framework for software engineering projects. Agile promotes industries' best practices that emphasize teamwork, customer involvement and collaboration, and rapid releases. A very important feature of agile is that it emphasizes on doing little of everything rather than doing all at a time.

There are many agile methodologies in the market currently. However, following are the most successful and most adapted methodologies:

- eXtreme Programming (XP)
- Kanban
- Scrum

eXtreme Programming

The eXtreme programming methodology improves the software quality and it also addresses the problem of frequently changing requirements by the customer faced in the traditional methodologies. This methodology mainly concentrates and improves on the coding, reviewing, and testing by allowing the pair programming.

Kanban

Kanban methodology solves the problem of traditional methodologies; of long releases cycles by emphasizing on frequent delivery. Kanban limits the WIP (Work In Progress) items in each workflow state, which will not overburden the development team and also helps to concentrate and produce better quality software in a shorter time.

Scrum

Among all the existing agile methodologies, scrum is the most popular and the most adapted one. Due to its simplicity and flexibility, the scrum has become verypopular. Scrum is an iterative and incremental agile methodology. It emphasizes on the self-motivated and self-managed small teams. Scrum is the people-centric framework, which emphasizes on team co-ordination, morale, trust, respect, and commitments. Scrum creates an opportunity for customers to continuously work with the development teams to get what exactly customers want. Scrum is very flexible and open to any change at any point in time or cycle (customer can change their requirements) which delivers 100% customer satisfaction. Scrum is built based on the following Agile Manifesto:

- Individuals and interactions over processes and tools
- Working software over comprehensive documents
- Customer collaboration over customer negotiation
- Responding to the change over following a fixed plan

Scrum in detail

Let us discuss how Scrum will address significant problems faced by traditional methodologies and how it produces a fine quality software with 100% customer satisfaction.

Freedom

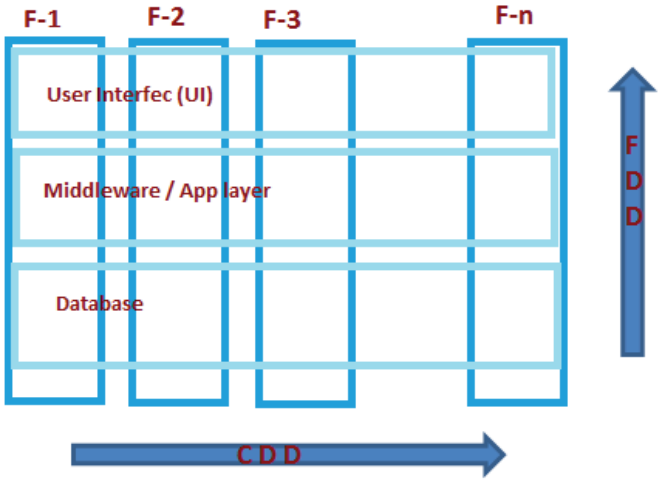
Scrum gives 100% freedom to the team to decide what works well for them and finally the whole team commits for the final delivery. Every individual in scrum are self-managed or self-organized. Everybody knows all the work items in scrum and an individual has the full freedom to choose the work item of their interest.

Communication and collaboration

The main building block of the success of scrum is the communication and collaboration. Communication and collaboration helps the team to clearly understand and commit on what customers need and it helps in discovering the challenges, risks, unknowns, impediments in the early stages. This will address the problems of traditional methodologies like last minute risks, rework, and customer negotiation.

Communication plays a vital role in providing clear visibility on the project schedule and the software quality. On every sprint, scrum demonstrates a working piece of software to the customer and is open for feedback and confirmation.

The team meets every day to discuss what they achieved yesterday and what they are going to work on today. The team also



discusses about risks or impediments if any, and resolves them as a team.

Requirement Gathering

In the traditional methodologies, we gather all the requirements in the beginning and freeze it. After the requirements are frozen, we start writing software as per the gathered requirements. If there are any changes in the requirements or customer needs, we need to change it in the designing, implementing and testing phases. But, scrum addresses this issue by using the Feature Driven Development (FDD) and opening the discussion with customer in each sprint to seek feedback.

Risk

In the traditional methodologies, many risks will be discovered during the last stages because traditional methodologies work in Component Driven Development (CDD) fashion.

Customer can see the working software only after developing the complete product which postpones all the potential risk and feedback to the last cycle. Scrum addresses major technical risks using the Feature Driven Development (FDD) which is explained in the above diagram.

With scrum, even in 100% worst cases only one sprint's work will get affected not the complete product unlike traditional methodologies.

Visibility

Visibility is one of the major reasons why people have adopted scrum. Scrum provides a clear visibility at all the stages to the development team, management and customers. At every sprint end, the working software will be demonstrated to the customer. If the customer has any feedback or change in the requirements, those will be addressed in the upcoming sprints and the iteration continues.

Adaptive to the change

Scrum is flexible enough to absorb the changes even at the late Feature Sprint also. Unlike the traditional methodologies, scrum won't work on the fixed plan and requirement. Scrum understands that the customer may not fully visualize the product in the early cycles and there is a chance of change in requirements. Scrum collaborates scrum teams and stakeholders (customers) together to seek feedback and confirmation on what is done till date.

Business benefits

Following are the major business benefits of the scrum methodologies over traditional non-agile methodologies.

- Adaptive to changes
- Increased productivity
- Effective resource utilization
- Improved visibility
- Improved team morale
- Enhanced software quality
- Reduced risk
- Faster time to market
- Improved engineering discipline
- Faster process improvement

Summary

Looking at all the facts discussed in the proposed solution section, we know that scrum is a proven methodology for the current market needs. We also looked at how the scrum is helping organizations and customers in delivering what customers actually need than what he asks. Scrum not only helps customers or organizations but also helps development team in improving their efficiency, effectiveness, morale, confidence and trust.

Call to action

We implemented scrum in several teams in NetIQ, it worked fairly well for us and our customers. We received a positive response about Scrum from all the stakeholders and team members. Scrum is easy and flexible to adopt and implement. One has to implement it to see the benefits of scrum.

bits & bytes

Google's Android Device manager, which allows you to do things like locate, lock, or wipe a lost device is a nifty tool with an unfortunate name – it doesn't tell you much about what it actually does. But at Google I/O, the company has sneakily changed the name into something more sensible: Find My Device.

Along with the name change, Google has added battery information, which should help you know how long you have to locate your device before it dies out, as well as the name of any WiFi network its currently connected to.

I can't tell you how many times someone I've known has lost their Android device without being aware that Google has a built-in tool for finding it. Compare that to my iOS-toting friends, all of whom seem to know about Find my iPhone, whether or not they use it. It's a small change, but one that will hopefully mean less phones permanently lost.

Exploring the Exploratory Testing

Written by **Raju Korti**, Member Technical Staff 1, SG Group

Exploratory testing, is all about discovery, investigation and learning. It is a hands-on approach in which testers are involved in minimum planning and maximum test execution

What is Exploratory Testing?

It is definitely not:

- Using test cases or planning
- Working in an agile environment
- List of problem solving procedures you must use

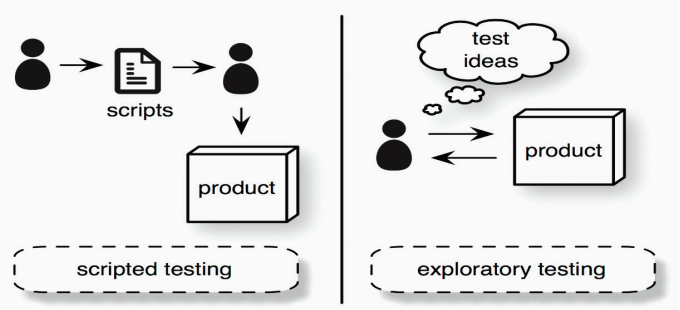
It certainly is:

- A software testing technique in which testers do not follow any specific test cases, plan or approach
- An opportunity to meet people that helps you to know people on a team and what they can do
- The idea of exploration, study, research and investigation.
- About asking questions, experimenting, and gaining knowledge about the software or application which tester are going to test
- Is highly efficient. By brain storming you gain a better understanding of the software
- Don't end up using outdated documentation
- Providing quick feedback and thereby helping everyone on the team to do a better job by using more up-to-date information about the product

Issues with **Exploratory Testing**

How to manage:

- What is being tested?
- How thorough the testing is?
- What defects are detected?



- How much risk remains within the application space in terms of existing and undiscovered issues?

Benefits of Exploratory Testing:

- Independent of the chosen development model. It can be applied to any situation, even outside of software development.
- Creates engagement and encourages people to think out of box.
- Identifies complex defects in a system much in advance
- Provides user-oriented feedback to developers and business analysts

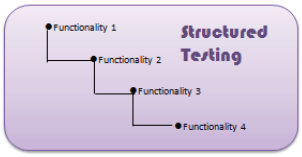
When to Use Exploratory Testing?

In general, exploratory testing is called for in any situation where it is not obvious what the next test should be, or when you want to go beyond the obvious tests. More specifically, you can use exploratory testing in any of the following situations:

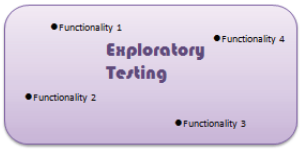
- Provide rapid feedback on a new product or feature.
- Learn the product quickly.
- Diversify the testing already tested using scripts.
- Find the single most important bug in the shortest time.
- Check the work of another tester by doing a brief independent investigation.
- Investigate and isolate a particular defect.
- Investigate the status of a particular risk, in order to evaluate the need for scripted tests in that area.

Exploratory Testing Best Practices:

- **Tester should have prior experience in software testing:** Typically exploratory testing needs a greater



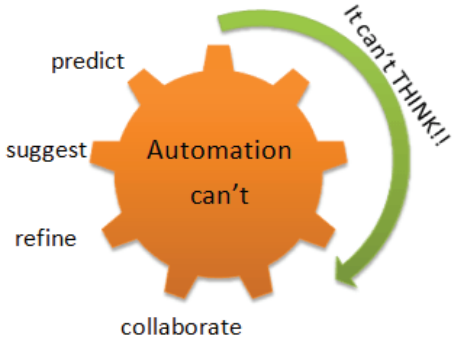
Functionalities are checked in a structured manner



Functionalities are checked in a ad-hoc manner

level of testing skill and prior experience than other testing techniques.

- **Need to focus on target area in terms of functionality:** Exploratory testing helps you exercise a system like a user while actively looking to identify bugs. Focus on these goals to maximize the value of your tests. Remember that exploratory testing can complement other testing methods that examine systems in different ways.
- **Have plan for this activity but don't restrict yourself:** You cannot do exploratory testing if you restrict yourself by following any guidelines. However, exploratory testing doesn't mean testing without control or good practice. You need to plan your testing in advance. Planning helps you to clarify specific aspects of a system that you want to examine, including special data requirements or system needs.
- **Avoid having too many features under test:** The aim of exploratory testing is not coverage. It is to find the defects and issues in a system that you won't find through other forms of testing. Typically these defects arise through edge cases of testing, but that doesn't mean they are low impact.
- **Time Boxing:** An exploratory test session can be limited to a set timescale, a practice also known as time boxing. This will be a focused test effort of fixed duration. This can help concentrate on the project's specific goals and scope, rather than drift into unfocused exploration.
- **Know the software you are testing from both the business and technological points of view:** What problems is it supposed to solve, who are the stakeholders, and how does it work?
- **Use test ideas to guide testing rather than test cases to define it:** It is important to ask questions and to look at problems from different angles, to use new tests instead of old ones.



Disadvantages of Exploratory Testing:

- Tests invented and performed on-the-fly can't be reviewed in advance and therefore it will not catch errors in code and test cases. It can be difficult to show exactly which tests have been run.
- Depends on the testers experience and skills
- Need in-depth domain or application knowledge
- Not suitable for long execution time
- Difficult to reproduce the defect as test cases are not available

Conclusion

Exploratory testing is performed to overcome the limitations of scripted testing. It helps in improving test case suite. It emphasises on learning and adaptability.

Thinking While Testing!!

Though the current trend in testing is to push for automation, exploratory testing is a new way of thinking. Automation has its limits.

bits & bytes

Machine Learning Writes Better Emails

Last year, google launched Smart Reply, a feature for Inbox by Gmail that uses machine learning to suggest replies to email. Now they have created a new version of Smart Reply for Gmail. This version increases the percentage of usable suggestions and is more algorithmically efficient, according to research.googleblog.com.

Inspired by how humans understand languages and concepts, they used hierarchical models of language, an approach that uses hierarchies of modules, each of which can learn, remember, and recognize a sequential pattern.

According to research.googleblog.com, the content of language is deeply hierarchical, reflected in the structure of language itself. A hierarchical approach to learning is well suited to the hierarchical nature of language. They have found that this approach works well for suggesting possible responses to emails.

Smart Reply relies on machine learning to scan the subject line and body of an email and make suggestions based on what it sees. The company said it has built up a huge bank of anonymized customer messages and response decisions to help accomplish this. It is also designed to remember your individual preferences

According to Kurzweilai.net, Google says the machine-generated replies already account for 12 percent of emails sent; expect that number to boom once everyone with the Gmail app can send one-tap responses.

Go Unsafe with C#

Written by **Koustov Maitra**, Member Technical Staff 2, Privilege Management

A volcano emits smoke. A knife glides through the night air. These things are dangerous and unsafe. In C#, unsafe things access memory directly and yes this is pointer

When you develop an application in C#, your code is well managed and there is no need to write explicit code to perform garbage collection or any other memory management. Memory Management and Garbage Collection are handled by CLR.

However, there are situations when you want to control certain blocks of memory. For example, to interface your application with an operating system or to use win API functions in your code. In such cases, using pointers will be of more help. However, pointers disturbs memory directly and it is not recommended by CLR. But still you can accomplish it by writing unsafe or unverifiable code in C#. This means CLR cannot guarantee the code safety.

So it is about walking beyond the comfort zone of C#

Why unsafe?

Being a wide ranging language what C# was missing is pointer which gives flexibility to access memory directly. Though pointers are not very common in C#, still it can be very helpful in handful of situations.

- Direct working on disk
- COM or Platform Invoke scenarios
- Performance improvement : It reduced number of instructions which improves performance
- It helps improving experience in some scenarios: Like get rid of array bounds.

The use of unsafe context in other situations is discouraged. Specifically, an unsafe context should not be used to attempt to write C code in C#.

Inside unsafe code

Consider an example of Square method which finds square of a given number.

The basic method would be expected as

```
static void Square(int* n)
{
    *n = *n * *n;
}
```

You can notice the list of error marks. So in C# user needs to declare the block of unsafe code that needs to be compiled with pointers. It can be block, method, class etc. So the rectified method is

```
static unsafe void Square(int* n)
{
    *n = *n * *n;
}
```

```
int nInput = 10;
Square(&nInput);
```

Now can you consume it?

Error again. So, just like earlier when we use pointer, the block needs to be declared as unsafe.

```
unsafe
{
    Square(&nInput);
}
```

This is a very basic example of using unsafe code.

Code Compilation

You might have guessed by now, .NET doesn't allow to compile the unsafe blocks due to security reasons. To compile code with unsafe blocks, you need to inform the compiler to allow unsafe code using command line argument of /unsafe. Developers who use "MS VS IDE" to compile their project can actually turn on the option in the Build tab of project properties

And the Result

As expected.

Advantages

- It is fully supported by .NET CLR
- Improves performance
- Helps in optimizing the complex structure
- Improves compatibility with binary code
- When it is required to manage a large blob of memory the unsafe code really becomes very handy

Cautions

- The unsafe context cannot be verified for code safety: Code can be run from user trusted context. So if you want to run the snippet anywhere else then it fails.
- With unsafe CLR housekeeping will be compromised: You bring back the native memory issues like dangling pointers, buffer over run.
- Pinned managed objects: If you want to access managed objects from your unsafe code then the objects are pinned which will cause GC to skip those objects.
- Reduce readability when you have a mix of managed and unsafe code floating around your application
- The unsafe context cannot be verified for code safety
- Using unsafe code requires manual treatment of memory, we might end up with decreasing performance instead of improving it.

Instance

Let's talk about one real life scenario where unsafe code is really useful. In image processing you need to deal with millions of pixels and process them. Going with simple matrix data structure approach in C# would take a while to get desired result from an image and you might be dealing with thousands of them. In scenarios like this, if you can get access to the memory and directly work, it will definitely improve the performance.

Here is an example that compares managed and unsafe processing.

Linear array access
  00:00:07.1053664 for Normal
  00:00:07.1197401 for Unsafe *(p + i)
Random array access
  00:00:42.5559436 for Normal
  00:00:40.5632554 for Unsafe
Random array access using Parallel.For(), with 4 processors
  00:00:10.6896303 for Normal
  00:00:10.1858376 for Unsafe

Closing bits

- Unsafe concept is beyond managed territory
- The concept of unsafe remain same across other .NET languages
- User should take care of memory clearance
- Unsafe is very useful in improving performance for large data processing

bits & bytes

Google has big plans for the virtual reality (VR) and the augmented reality but the company believes that the world of all things VR is still waiting for its iPhone moment. Clay Bavor, the VP of VR and augmented reality at Google, said at the Google I/O 2017 that he was very pleased with the progress the company was making VR but the big breakthrough that would popularise the dorky headsets was still awaited.

"In the last six months after launching Daydream (Google's VR platform) we are very pleased with the progress we have made," said Bavor. "But we are long way from reaching the iPhone moment in the VR."

Bavor's comments came in response to a question about the perceived slow progress of the VR and related gadgets. A number of companies, including Facebook that now owns Oculus and HTC that has made Vive VR headset, are working on virtual reality. Microsoft too has something in the works under the HoloLens program, although it is more into the domain of augmented reality. While the technologies are exciting, so far the VR products have failed to catch the fancy of consumers.

Improve Java performance using Memoization

Written by **Binod Suman**, Sr Member Technical Staff 1, Security

Memoization decreases the computation time of the optimized algorithm as it reduces the amount of calculation that is required by storing the calculated data into a data structure such as an array or treemap

Many functions perform redundant calculations. Within a single function invocation, several sub functions may be invoked with exactly the same arguments, or, over time in a system run, a function may be invoked by different users or routines with the same or similar inputs. This observation leads to the conclusion that in some cases it is beneficial to store the previously computed values and only perform a calculation in situations that have not been seen previously. This technique is called "memoization."

Memoization consists of caching the results of functions in order to speed them up when they are called several times with the same argument. The first call implies computing and storing the result in memory before returning it. Subsequent calls with the same parameter imply only fetching the previously stored value and returning it.

What to memorize:

1. A method costly to execute.
2. A method always returns the same output form the same input.
3. A method that is called many times with the same input.
4. Results from database queries (Like hibernate level cache)

How to implement:

```
Integer costlyFunction(Integer x){
    // some expensive calculation done here
   return someInteger;
}
```

Corresponds to:

```
private Map<Integer, Integer> cache = new
HashMap<Integer, Integer>();
Integer costlyFunction(Integer x) {
  if (cache.containsKey(x)) {
    return cache.get(x);
  } else {
    Integer result = // some expensive
calculation done here
    cache.put(x, result) ;
    return result;
  }
}
```

We can improve the above code to using ConcurrentHashMap to prevent two threads from attempting to simultaneously compute the same input values. Note the new method `java.util.Map`. computeIfAbsent which takes a function as its second argument - it was added specifically for this use case.

```
private Map<Integer, Integer> cache = new
ConcurrentHashMap<>();
Integer costlyFunction(Integer x) {
  if (cache.containsKey(x)) {
    return cache.get(x);
  } else {
    Integer result = // some expensive
calculation done here
    cache.put(x, result) ;
    return result;
  }
}
```

The above code could be simpler and advance using computeIfAbsent method.

```
private Map<Integer, Integer> cache = new
ConcurrentHashMap<>();
Integer costlyFunction(Integer x) {
  return cache.computeIfAbsent(x,<Java_Class_Name>::
myAlgo);
}
static Integer myAlgo(Integer x) {
```

```
    Integer result = // some expensive calculation
done here
    return result;
}
```

We can enhance and make it more cleaner using Lambda expression (Java 8)

```
private Map<Integer, Integer> cache = new
ConcurrentHashMap<>();
Integer costlyFunction(Integer x) {
return cache.computeIfAbsent(x,y -> myAlgo(x));
}
static Integer myAlgo(Integer x) {
    Integer result = // some expensive calculation
done here
    return result;
}
```

In java 8, using java.util.function class, we can further improve this code and make it very generic.

```
private Map<Integer, Integer> cache = new
ConcurrentHashMap<>();
Integer costlyFunction(Integer x) {
Function<Integer, Integer>functionResult =
memoize(Java_Class_Name:: myAlgo);
return functionResult.apply(x);
}
public static <T, R> Function<T, R>
memoize(Function<T, R> fn) {
   Map<T, R> map = new ConcurrentHashMap<T,
R>();
    return (t) -> map.computeIfAbsent(t, fn);
}
static Integer myAlgo(Integer x) {
    Integer result = // some expensive
calculation done here
    return result;
}
```

What if we want to memoize a recursive function? Two options suggest themselves:

1. Use the non-threadsafe memoization function given above, and make sure it's never called from more than one thread at once.
2. Wrap the non-threadsafe memoization function with a re-entrant lock. This is thread-safe, but a lot less efficient than using ConcurrentMap, since the latter locks on individual hash buckets rather than globally locking the entire map during updates.

Here is a thread-safe and recursion-safe implementation using a re-entrant lock:

```
public static <I, O> Function<I, O>
memoize(Function<I, O> f) {
    Map<I, O> lookup = new HashMap<>();
    ReentrantLock lock = new ReentrantLock();
    return input -> {
        lock.lock();
        try {
            return lookup.computeIfAbsent(input, f);
        } finally {
            lock.unlock();
        }
    };
}
```

Memoization benefit:

- Memoization decreases the computation time of the optimized algorithm as it reduces the amount of calculation that is required by storing the calculated data into a data structure such as an array or treemap. For example, the original runtime of the fibonacci sequence according to big-O notation is exponential time (O(n^n)) however with memoization the algorithm runs at linear time as we are returning O(n) in many cases.

Memoization Drawback:

- Memoization requires more space as another data structure is needed to store the values already calculated, if you take example of fibonacci sequence not much more space is used however in many cases memoization takes up too much extra space when used to be practical in case where big computation would be there.

In essence, when you use memoization you are trading time for space, you decrease the computation time of your algorithm but increase the amount of space in memory that your program uses.

I used to below code to test the above memorization:

```
import java.util.Map;
import java.util.concurrent.ConcurrentHashMap;
import java.util.function.Function;


public class Memoization {
       //private Map<Integer, Integer> cache = new
HashMap<Integer, Integer>();
       private Map<Integer, Integer> cache = new
ConcurrentHashMap<>();

       public static void main(String[] args) {

              Memoization demo = new Memoization();
              System.out.println(demo.
doubleValue(5));
       }

       /*Integer doubleValue(Integer x) {
              return x * 2;
       }*/

       /*Integer doubleValue(Integer x) {
              if (cache.containsKey(x)) {
                return cache.get(x);
              } else {
                Integer result = x * 2;
                cache.put(x, result) ;
                return result;
              }
            }*/

       Integer doubleValue(Integer x) {
              // return cache.computeIfAbsent(x,Mem
oization::getDoubleValue);
              //return cache.computeIfAbsent(x,y ->
getDoubleValue(x));
            Function<Integer, Integer> g=  memoize
(Memoization::getDoubleValue);
            return g.apply(x);
       }

       static Integer getDoubleValue(Integer x) {
              return x * 2;
       }

       public static <T, R> Function<T, R>
memoize(Function<T, R> fn) {
              Map<T, R> map = new
ConcurrentHashMap<T, R>();
              return (t) -> map.
computeIfAbsent(t, fn);
            }

}
```

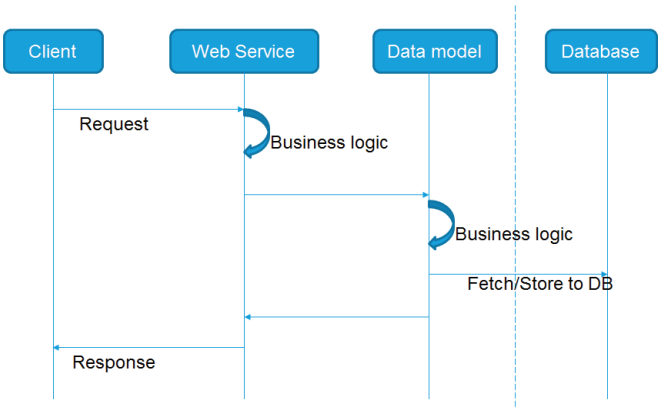
Integration Tests? Why?

Written by **Vallish Kumaraswami**, Sr Member Technical Staff 1, ZENworks

. . . gives focus on the class under test without worrying on the remaining classes. The advantage of this type of test is that it gives very quick feedback to the engineer on the code that is getting modified

We all are aware of unit tests. They are tests written to cover a unit. In object oriented languages like Java, this would be a single class. To test a given class, we mock all the classes that this class depends on. This gives focus on the class under test without worrying on the remaining classes. The advantage of this type of test is that it gives very quick feedback to the engineer on the code that is getting modified. But the main drawback of unit tests are that they focus only on one class in isolation and hence do not exercise the interactions between two or more classes. Interactions between two classes become particularly important when in-house written code interacts with 3rd party library over which the engineer has no control.

We are also aware of end-to-end tests. They are tests that cover a workflow in the exact same way as an end-user uses the system. This needs all the systems, environment and test beds to be in place. The advantage with these tests are that they simulate the exact user conditions and hence are very near to production environment. They provide very good feedback of the feature. End-to-end tests are considered to be the most ideal form of tests because of the ability to catch most of the bugs during the product development life cycle. But end-to-end tests take a long time to complete due to their dependency on complex environment, need for complete build and 3rd party automation tools. Due to these dependencies, end-to-end tests also tend to be flaky and need time to become stable. What this means is that the feedback to the engineer is delayed thereby increasing the turn-around time for a given feature or a bug fix.



Is there a middle ground? Is there a way that developers can enjoy quick turnaround time of unit tests while being able to test interactions across multiple classes that form a logical group? Is there a way to test with as less simulations of complex environment as possible without skipping those tests in time-crunch periods?

At EPM, we attempted to solve this problem for web services. Why web services?

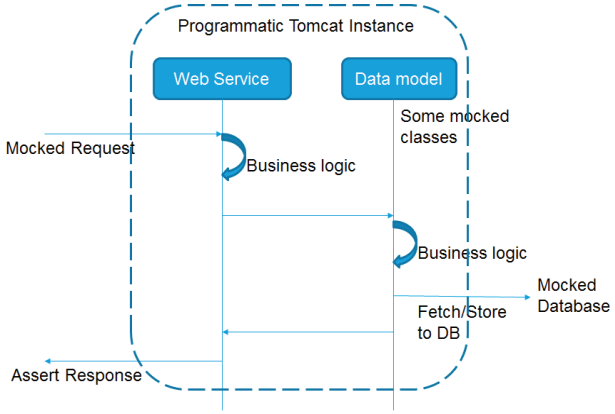
Web Services are a typical example of what we described above. They usually need a consumer/client and hence are tested end-to-end. The following is an example of a typical web service:

Here, the aim is to test the business logic in two layers as well as the wiring/flow of data between the layers. This includes serialization/de-serialization of request and response and actual hosting of the web service in a tomcat instance. We are not interested in storing or retrieving from an actual database since data does not change at rest. The client itself is tested separately as well. Thus, in effect, the following needs to be achieved:

How did we achieve it?

This involved the following steps:

- From the WSDL file that defines the web application's services, generate client stub classes using any WSDL to java converter



- Instantiate java Tomcat and deploy the webapp programmatically (see code snippet below)
- Make requests to web service using generated client classes (mock classes if required)
- Assert responses using TestNG framework

```
public class TomcatLauncher
{ static Tomcat tomcat;
public void launchTomCat(String webappName, String
warFile) throws Exception
{
    File rootFolderHavingWarBuilt = getRootFolder();
    tomcat = new Tomcat();
    URL customPath = (new File("target/classes")).
toURI().toURL();

    String dir = System.getProperty("java.
io.tmpdir");
    dir = dir+"/tomcattemp";
    System.setProperty("ZENWORKS_HOME", dir);
    tomcat.setBaseDir(dir);
    tomcat.getHost().setAppBase(dir);
    tomcat.getHost().setAutoDeploy(true);
    tomcat.setPort(Integer.valueOf(8081));
    File webContentFolder = new File(rootFolderHaving
WarBuilt,"/"+ warFile);
    ctx = tomcat.addWebapp("/"+webappName, webCon-
tentFolder.getAbsolutePath());
    tomcat.start();
    //tomcat.getServer().await();
   }
}
```

Advantages of this approach of integration testing:

- Avoids longer feedback loop and can be run on development engineer's box
- Tests tying up of all layers of service and business logic
- Removes testing dependency on platform tools/automation and is Jenkins-able
  - Hence, the other tests termed as large tests
- Helps in separation of service layer from database access code
  - Cannot test this way if war contains code referring to classes that needs to be mocked. For example, Initializing from file system
- Can do scale tests on development box!!!

At EPM, we call these types of integration tests as "Component" tests.

bits & bytes

World's First Underwater IoT Network

A text message is sent from a boat floating about 800 meters off from another one carrying a group of journalists in the waters about 10 kilometers west of the Port of Incheon on Tuesday.

"I am hungry!"

The message was successfully received by a hydrophone installed 25 meters underwater by a research team led by Ko Hak-lim, professor of oceanic marine IT convergence technology at Hoseo University.

It took about 20 seconds to transfer the message through the long term evolution network operated by South Korea's leading mobile carrier SK Telecom.

What we do here is to realize a so-called underwater IoT network, and Korea will be the first in the world to establish an adaptive underwater wireless network," said Ko. "Once we build the network here in the West Sea of Korea, which is infamous for its turbulent sea currents, the technology will be applicable to any part of the oceans, and will endure strong currents."

The scholar's team is in the third year of working on this underwater project with SK Telecom, developing a communications system connecting an underwater station with an offshore buoy. The project was commissioned by the Ministry of Oceans and Fisheries.

The buoy is supposed to control the underwater station and send the signals from the waters to SKT's onshore network.

In the current stage, the joint team has succeeded in sending and receiving text messages, low-resolution pictures and real-time data about water temperatures, sea currents, salinity, submarine earthquakes and tsunamis, which are collected from underwater sensors via a pilot network.

The team aims to embark on construction of a test bed for the underwater network this October and complete by 2020.

"We plan to install a number of underwater stations with a diameter of 1 kilometer in multiple areas that are in need of communications for economic and security purposes," the professor said. "The underwater LTE network can be widely applied to issuing early warnings for oceanic disasters such as tsunamis, protecting fisheries, monitoring the submarine environment and detecting glitches in deep-water oil drilling, and so forth."

There have been active studies and research on submarine communications technologies in countries like the United States, Europe, China and Canada. However, most of them have been limited to fixed-line networks, according to SKT.

"SKT is the only firm in the country with the technologies to design LTE networks for public protection services, railroads, maritime facilities and submarine resources," said Park Jin-hyo, head of network technology institute of SKT.

Location Aware Access Techniques

Written by **Ravi Kiran Jayanthi**, Member Technical Staff 1, Access

Geo-location authentication uses IP, which can be spoofed . . .

With the increased threats from hackers and the recent cyber-attacks to web mail providers, like Yahoo, have been forcing end users and administrators to think of a stronger and secure access, while keeping in mind, important factors like, convenience to end users and cost effective deployments.

Apart from device based authentications, which are detrimental to cost and convenience, another way of keeping resources more secure is, with context-based authentication. Rules can be made to take into account the context parameters like location of origin, risk profile assessment and a decision to allow or "step up" to two factor authentication.

Client's geographical location as an important authentication factor to assess the risk of an authentication request, in a way enhances security of applications.

GeoLocation

Similar to factors like, "what you have", what you are", a new factor "somewhere you are", geolocation. This location capability is commonly performed by isolating a host system's IP address from a packet header and compared against GeoLocation database to verify for a match. If a match is not found, the request is put to step up authentication or deny.

GeoLocation database matches such assignments to the location the network has registered. There are a number of free and paid subscription geolocation databases, ranging from country level to city or state level. These databases typically contain IP address data. Organizations can ensure, the authentication request's source IP address comes from approved locations (countries etc..) Or IP ranges associated with their employees or customers.

If a user from India is having his account accessed from somewhere in Middle East, you may want to throw up additional challenges like security questions, after they have provided the correct answers. Now if the user is rightfully on a Desert Safari, they will be able to answer the challenges and your system will note this new location down a possible location, that user uses.

Geo-location authentication uses IP, which can be spoofed, and is not geographically accurate. But one of the important contextual factors to assess the risk profile of the request.

This is not a very accurate method and may be completely wrong if you are using a VPN to tunnel all your Internet traffic.

Geo-fencing – balances user experience and security, with an end user authentication using a GPS device. Defines a virtual boundary, or geo-fence, around a specific physical area, outside which access is not allowed. This area could be very small, within the perimeter of a certain building. It helps better management of BOYD devices. An administrator decides who can access what within that barrier, based on GPS coordinates. A tracked device that moves into or out of the fenced area triggers an event.

If the computer running the browser has built in a GPS hardware like most smart phones today, the geo-location can be narrowed down pretty well.

On the flip side, the data collected, such as the duration of the geo-fence's existence, etc. can be intrusive, on end user's privacy.

Geo-velocity

Using a user's geo-location and login history together can also help prevent malicious login access. This factor is based on the location and time of the end-user's last successful login event, as compared to the time and location of the current login attempt. if the time span between the last successful login event and the current login attempt is less than the travel time between the two locations, then request will be denied to proceed to actual authentication.

For example, if a user logged in at 2 p.m. IST in Bangalore, it is reasonable to consider a logon attempt at 5 p.m. IST from Delhi, but not from a farther location, like HongKong etc. So the same can be made as "improbable travel event" and deny the request.

Conclusion

There is a range of authentication methods available today, to deny hackers or attackers being accessed, but the challenge is to make use existing parameters, on the level of assurance and accountability you need for different scenarios. For example, combining both contextual authentication with biometric or any strong authentication could eliminate the need for a password or token altogether.

Context-based authentication can be tailored to ones organization's risk tolerance, to balance security with a better user experience. Users are unaware of the context-based authentication processes and are not burdened by two-factor authentication unless a login is deemed to involve a certain level of risk.

Metasploit- Securing the enterprises via securing the software's

Written by **Neeraj Vijay**, Member Technical Staff 2, Access Group

All powerful things comes with responsibilities, Metasploit is very powerful framework has lots of features, capabilities and tools for destruction

With the development of devices and software, it became a necessity to secure the information and devices. Though there is different ways to strengthen the security, one of the key process is to have a strong security plan in place and penetration tool to ensure we achieve it.

In NetIQ Access Manager (NAM), robust security testing is followed. This is achieved through the set of guidelines and test tools. Metasploit addition strengthen the overall Security testing and protect from known vulnerability.

This article includes:

- Metasploit Pro Framework Overview
- Metasploit Pro Workflow
- How Metasploit Pro fits into overall security coverage.

Metasploit Framework Overview

The Metasploit Framework is an open source penetration testing and development platform that provides the latest exploit code for various applications, operating systems, and platforms. Pen-testers can leverage the power of the Framework to create additional custom security tools or write your own payloads for new vulnerabilities.

Installation of Metasploit is simple and supported on multiple versions of Linux and windows Operating system. Key framework components are:

Services: Metasploit Pro runs the following services:

- PostgreSQL runs the database that Metasploit Pro uses to store data from a project.
- Ruby on Rails runs the web interface
- Pro service, or the Metasploit service bootstraps Rails, the Metasploit Framework, and the Metasploit RPC server.

Web Interface

A web interface is available for you to work with Metasploit Pro. To launch the web interface, open a web browser and go to https://localhost:3790.

Command Line Interface

The Pro Console enables you to interact with Metasploit Pro from the command line.

Key terms use with Metasploit Pro

Following are some basic terms and concepts that is essential to work effectively with the tool:

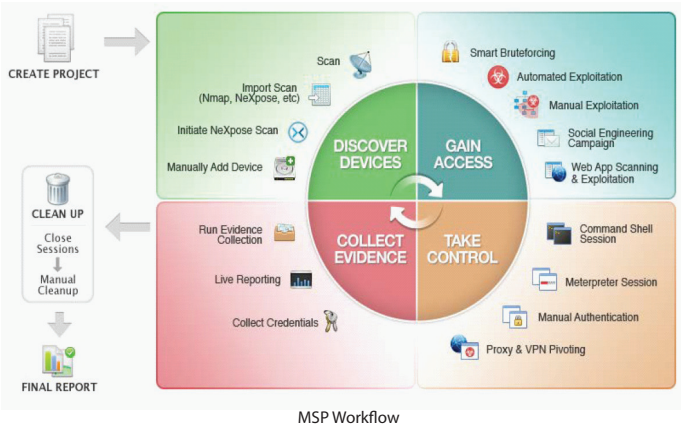
Project: A container for the targets, tasks, reports, and data that are part of a penetration test. A project contains the workspace that you use to create a penetration test and configure tasks. Every penetration test runs from within a project.

Framework: A module can be an exploit, auxiliary or post-exploitation module. The module type determines its purpose. For example, any module that can open a shell on a target is considered an exploit module. A popular exploit module is MS08-067.

Meterpreter: An advanced multi-function payload that provides you an interactive shell. From the Meterpreter shell, you can do things like download a file, obtain the password hashes for user accounts, and pivot into other networks. Meterpreter runs on memory, so it is undetectable by most intrusion detection systems.

Module: A standalone prepackaged piece of code that extends the functionality of the Metasploit

Post-exploitation module: A module that enables you to gather more information or to gain further access to an exploited



MSP Workflow

target system. Examples of post-exploitation modules include hash dumps or application and service enumerators.

Exploit: A program that takes advantage of a specific vulnerability and provides an attacker with access to the target system. An exploit typically carries a payload and delivers it to a target. For example, one of the most common exploits is MS08-067, which targets a Windows Server Service

Listener: A listener waits for an incoming connection from either the exploited target or the attacking machine and manages the connection when it receives it.

Payload: The actual code that executes on the target system after an exploit successfully compromises a target. The payload enables you to define how you want to connect to the shell and what you want to do to the target system after you take control of it.

Metasploit Workflow

Workflow Summary

Metasploit workflow is defined based on penetration testing standards and procedures. Typical Metasploit workflow consist of:

Creation of Project: For any penetration testing, first you need to create a project. This is required to collect data from security testing.

Gather Information: This is called reconnaissance in penetration testing terminology. The idea is to gather as much information possible from the application under test. In this phase, Metasploit Pro discovery scan is used to gather information.

Also, there exists an option to import already available data from other tools like Nessus, burp or other commercial penetration testing tools.

Exploit: Once the information is gathered, Metasploit provides an option to perform auto-exploitation or manual exploitation of target to gain access. Multiple ways available such as, web app scanning, Web app exploitation and social engineering to gain access of the remote host.

Post exploitation: Once the access is gained on target hosts, additional tools and modules are available to exploit and gain further level of access. e.g. Using these steps, the pen-tester can get information at system level by increasing their privilege, collect system data, passwords and download sensitive information from the remote host.

Brute force: this method is used to find the password in systems across network or in application. This can be used to find valid login credentials.

Clean-up Open session: Ensure that no footprint is left on the target system. These steps restores original setting on target host.

Generate Report: Metasploit provide custom as well template based report. Template can be selected based on the requirement. E.g. you may choose custom reporting when you have more than 1 report to include for particular test. All the captured data and information can be included in final report.

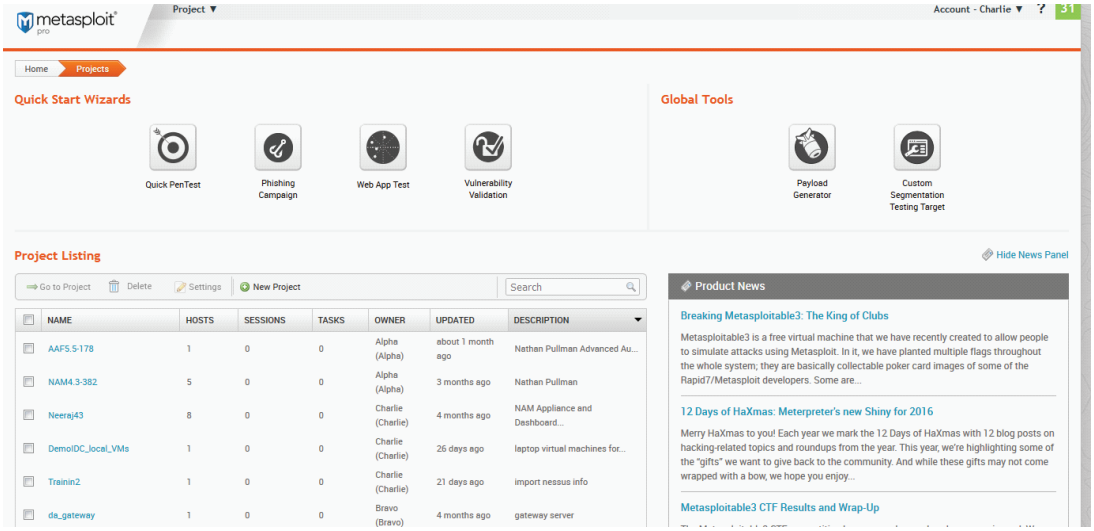
After the test execution identified vulnerability are captured and displayed in the dashboard:

How Metasploit fits into overall security test coverage?

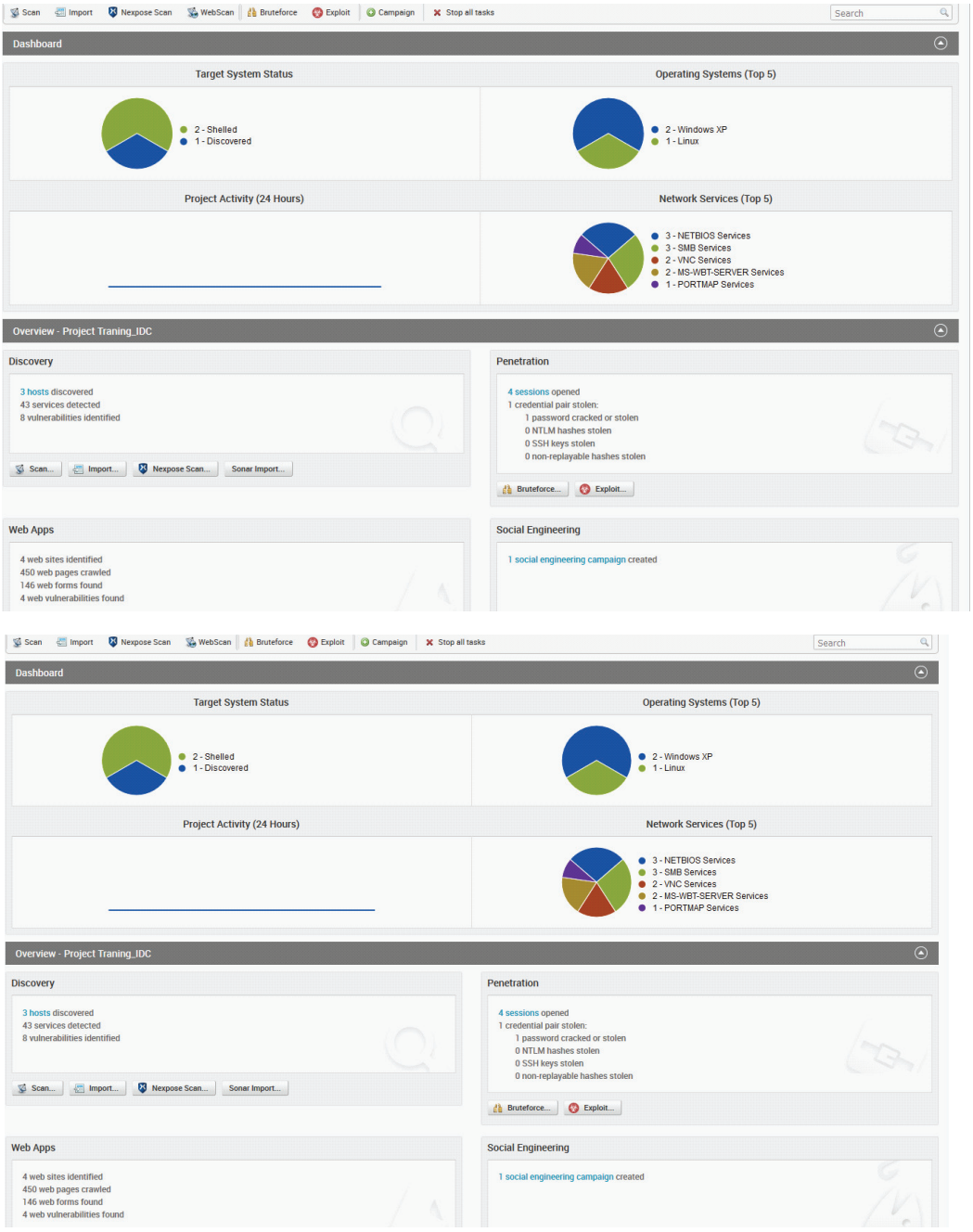
So far, we have discussed about the Metasploit framework and its workflow. The usage of Metasploit highly depends on the application under test. It is very useful for the appliances, network products and Network administrators. On the other side, it can be tweak to create custom payload based on the application under test.

One of the recent test performed to identify WannaCry (ms17-010) vulnerability existence on the Network hosts. Since Metasploit has got required module to identify, it was easy to point and validate the host running.

Since NAM supports the Appliance and Non-appliance version, it is important to test product to validate all possible weakness in the system. It could be vulnerability in services, OS or web application.



Metasploit Dashboard



Project Dashboard after the test performed

We do a lot of reconnaissance with all the available pen-test tool to identify weakness against known vulnerability.

In NAM, here is the typical work flow we follow:

- We perform Nessus scan on a set of targets and gather information, this is saved as a data file for future uses
- Second level of test performed using Burp. The information is captured and saved as xml for future use
- Metasploit Pro is executed on same set of targets, to identify the known vulnerability against the components using the workflow discussed in previous section. Since it gets updated frequently, this helps to identify any known vulnerability
- Additional information from Nessus and Burp is used in Metasploit for active exploitation and identify the applicable exploits. The difference is recorded.
- All the reports are gathered and proper response/mitigation plan is discussed.

This whole process is created as tasks in Metasploit Pro and triggered whenever required.

Conclusion

In this article, I have focused on the Metasploit Pro GUI and how we can include it in our security testing to ensure that we covered all aspect of penetration testing and exploitation before we ship our product. Metasploit is very powerful framework has lots of features, capabilities and tools for destruction. Whenever pen-test is performed on the product, you should ensure that you adhere to Pen test code of ethics and have permission to test using this tool.

References

1. https://community.rapid7.com/docs/DOC-1570
2. https://community.rapid7.com/docs/DOC-1567
3. https://community.rapid7.com/community/metasploit

.Net CLR for SQL Server

Written by **V. Chandra Mohan Reddy**, NetIQ AppManager

The CLR provides services such as automatic garbage collection, security support, and runtime type checking

As we know it can be difficult to write queries in SQL Server for the complicated tasks, with T-SQL, and it would be easy with a traditional language. Microsoft has given a way to get the benefits of programming languages in the SQL Server by using the CLR.

The Common Language RunTime (CLR) is the core engine of the .Net framework which manages the execution of programs written for .Net framework regardless of the programming language. The CLR provides services such as automatic garbage collection, security support, and runtime type checking.

SQL Server 2005 and later versions host the CLR in the Database Engine. This is called *CLR integration*. With this integration you can create database objects such as functions, stored procedures, triggers, user-defined types (UDTs), and user-defined aggregate (UDA) functions in programming languages supported by the CLR. Managed code running in SQL Server-hosted CLR is referred to as a *CLR routine*.

Generally, we should use SQL Query when the code primarily performs the data access. CLR routines are best for CPU-intensive calculations and for supporting complex logic that would be difficult to implement using the SQL Queries.

The methods written in C# for SQL routines must be annotated with attributes such as SqlProcedure, SqlFunction, SqlTrigger, SqlUserDefinedType . . .

How to create CLR for SQL Server in C#

From Visual Studio 2010, follow the below path and create the project for SQL Server (SQL CLR is not supported for Azure SQL DB.

New Project > Database > SQL Server > Select .Net Framework > Select "Visual C# SQL CLR Database Project > Add Database Reference (Cancel it if you want to deploy manually)

Create the required SQL routines from the "Add" option of the project. Suppose if the user select the Stored Procedure the following code is added with the annotation method SqlProcedure.

```
public partial class StoredProcedures
{
    [Microsoft.SqlServer.Server.SqlProcedure]
    public static void StoredProcedure1()
    {         // Put your code here     }       };
```

How to Enable CLR in SQL Server

The ability of SQL Server to execute CLR code is off by default, So SQL routines will not execute in SQL Server unless the "clr enabled" option is enabled in SQL Server by using sp_configure as shown in below.

```
sp_configure 'clr enabled', 1
      GO
      Reconfigure
      GO
```

After *sp_configure* and *Reconfigure* are executed-the server does not need to be restarted.

Deploy the CLR in SQL

CLR can be deployed either by Auto Deployment or Manually.

For the Auto Deployment, In Visual Studio user has to add the database reference where you want deploy automatically .Deploying a SQL Server Project in Microsoft Visual Studio registers an assembly in the database that was specified for the project. Deploying the project also creates CLR functions in the database for all methods annotated with the supported annotations.

Manual Deployment in SQL Server

1. Register the C# assembly in the SQL Server as shown below.

```
CREATE ASSEMBLY [<AssemplyName_In_
SQLServer>] FROM  <Path_of_the_DLL>
WITH PERMISSION_SET = SAFE
```

2. Create the Releavant SQL Server object in SQL Server. If assembly is the stored procedure follow as below.

```
CREATE PROCEDURE [dbo].[<Procedure_Name>]
WITH EXECUTE AS CALLER AS EXTERNAL NAME
[<AssemplyName_In_SQLServer>].[<ClassName>].
[<Method_Name>]
GO
```

3. Execute the above stored procedure

```
exec dbo. [<Procedure_Name>]
```

Real Time Example of SQL CLR with example

SQL Server doesn't support full regular expressions, – sometimes, the LIKE operator can be an option, but it lacks the flexibility that regular expressions provides. But this can be achieved using the SQL CLR. The below example explains the complete c# code and usage of Regular Expression in the SQL Server.

1. Code in C#:

```
public partial class StoredProcedures
{
    [Microsoft.SqlServer.Server.SqlProcedure]
    public static void regex_
validation(string input_string, string
regex_String, out string result)
    {
        Regex rgx = new Regex(regex_String);
        result = (rgx.IsMatch(input_string))
? "Regular Expression Matched" : "Not
Matched";
    }
};
```

2. Registering the Assembly in SQL Server.

```
Create ASSEMBLY Regex_Assembly FROM 'C:\temp-sql_
clr\SQL_Without_Database.dll'
WITH PERMISSION_SET = SAFE;
```

3. Creating the Stored Procedure in SQL Server.

```
CREATE PROCEDURE [dbo].[Regex_Validation_Sproc] @
input nvarchar(max)  ,  @regex nvarchar(max)  ,  @
result nvarchar(max)  out
WITH EXECUTE AS CALLER AS EXTERNAL NAME [Regex_
Assembly].[StoredProcedures].[regex_validation]
```

4. Executing the Stored Procedure:

```
declare  @result nvarchar(max)
exec [Regex_Validation_Sproc] 'MemPhysUsage^^%',
'^(Mem)?PhysUsage\^\^%$',  @result output
select  @result 'Result_From_SQLCLR'
```

After executing the above stored procedure we will get the result as "Regular Expression Matched".

We can also view the list of assemblies registered in SQL Server with the query "select * from master.sys.assemblies"

Now we have solved the Regex validation in the SQL Server using the C# with the help of the CLR which is not possible with the SQL Server. Of course, SQL CLR can be used to do much more than this!

bits & bytes

Google presents Cloud Spanner as a happy medium between two common database needs that often prove incompatible. A database can be highly scalable and distributed (the NoSQL approach), or it can be transactionally consistent (the conventional database approach). Cloud Spanner aims to be both.

As laid out in a 2012 research paper, one key to accomplish this is a time synchronization mechanism for actions that need to be kept consistent between nodes—such as globally consistent read operations, which people expect from a transactional database.

This sync mechanism takes into account the potential differences between timestamps provided by different machines in the cluster and can "wait out" the differences if they are too large. But the system also tries to keep uncertainty to a minimum by drawing on multiple time sources to increase clock accuracy. As a result, it's easier to get operations spread across multiple nodes (for example, MapReduce) to agree on when something was achieved and to deliver consistent results.

pCloudy – "Simplifying Mobile App Testing"

Written by **Aruna Kumari P**, Member Technical Staff 2, Security Group

pCloudy helps developers and testers by providing real time access of various mobile devices, where a signed users can access over the internet via a browser

We all know about the mobile boom; from advertising to app development, to technology enhancement, virtually everything revolves around mobiles and hand-held devices.

Well, apart from the fact that the technology world is now centering on mobile devices, the demand for providing perfectly functional apps, with the right targeted devices is becoming a primary challenge to the industry.

Challenges foreseen by Engineer in Testing Mobile Application

- Traditional App testing is time consuming
- Don't have in-house testing environment
- Don't have the right tools to test
- Don't have the device readily available
- Don't' have the right testing Process/Method
- No mobile testing experts
- Not enough time to test on all devices
- We don't do mobile testing

That is exactly where pCloudy pitches in and makes our jobs easy. It helps developers and testers by providing real time access of various mobile devices, where a signed users can access over the internet via a browser.

pCloudy "Simplest Mobile App Testing"

- Realtime, Anytime, Anywhere
- Public Cloud
- Various Real Android and iOS devices
- across multiple networks

pCloudy provides remote access to hundreds of real Android and IOS devices - using just a browser along with many tools for testing and debugging Web app. The information is protected and secure over web sockets.

Features and Benefits of pCloudy

Browser Support

pCloudy devices and the entire test environment can be accessed through the browser using registered user. It helps to ensure that your apps work on latest OS versions.

Location Testing

One can remotely connect to any device from anywhere in the world to launch the testing platform. There are no location restriction for creating the test environment. You can use any network and any device location to simulate live or customer environment.

Variants of Devices

pCloudy has 100's of Android and IOS devices deployed in the cloud, and it keeps adding the new devices on weekly basis. Through pCloudy one can access these hundreds of devices and debug their application quickly. This remote cloud testing solution reduces project cost, management overheads and optimizes device sharing among development and testing teams. These various devices are classified based on OS/OEM/Screen Size/ Network and Device location for ease of use.

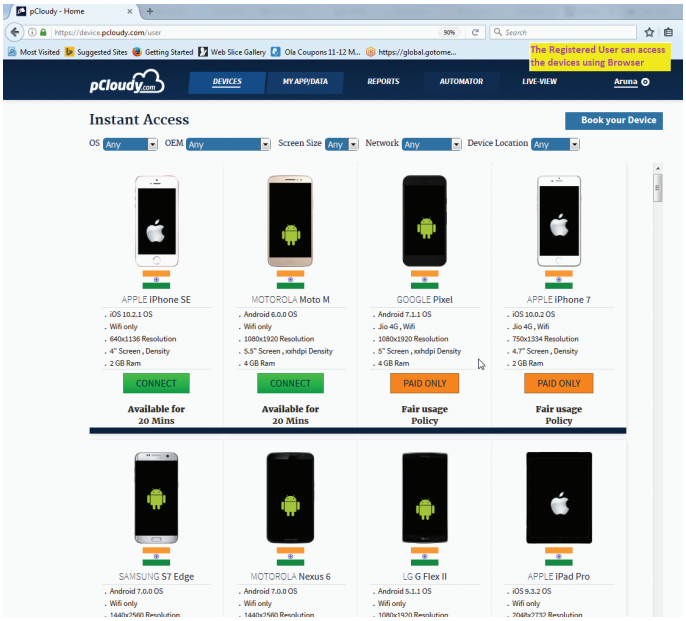


Fig 1 Web access of the pCloudy environment using HTTP Browser

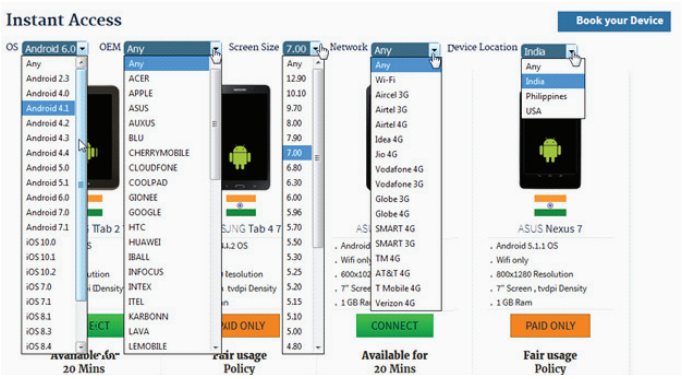


Fig 2 pCloudy where one can select the required IOS or Android from the available list

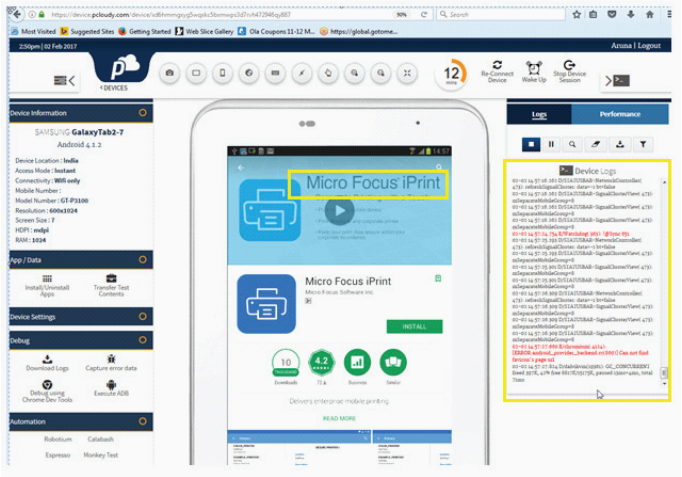


Fig 3 Monitoring Live Device-Logs while installing iPrint App

Performance Metrics Tools

pCloudy provide intelligent tools like CPU and memory profiling and network usage monitoring. In addition, it provides live performance metrics through real-time access and test automation. It has the ability to analyze performance bottlenecks, CPU usage, memory consumption, network usage monitoring, and critical paths on popular real devices.

Full-Resolution Screen Shots

It allows quick and easy capture of app screenshots or responsive web layouts. You can get full resolution, i.e. 1:1 pixel, screenshots anytime during the use of the device. With this one can instantly share and report problems or capture app layouts with any platform.

Cloud Drive

pCloudy, provides a large Cloud Drive space to manage all apps and test data (around 10GB Cloud Storage for test data, reports and more). One can store logs, screenshots and videos on the cloud and share it effortlessly with other users. Ease of collaboration makes teamwork efficient.

Secure Devices

pCloudy does not employ any hacks or tamper any data or the devices. One can test their app on market deployed devices which are connected to WiFi and cellular networks on demand. It has **Public cloud, Private cloud and on-premises** cloud, one can decide which cloud environment they want to use to deploy and test their App as per their business need.

Test Automator

pCloudy, automation service allows to test the app on multiple devices simultaneously. It support to run the tests using Robotium, Calabash and Appium. Each test cycle generates powerful graphical reports along with debug logs, screenshots, and session videos. It provides fastest way to automate testing, test runs and get reports from multi-devices to one's inbox. Test on all devices and get report to a single inbox through automation, helps testers to cover more devices in lesser time.

Weinre Debugger pCloudy installs the app and debug through weinre, which is a remote debugger for hybrid mobile apps, it provides deep access into the internals of installed application. Weinre helps to efficiently track down layout issues and get insights for code optimization, networks stats, much more and facilitate to fix the issues.

Chrome Dev Tools pCloudy provides in depth debugging for apps and websites through Chrome Dev Tools. Chrome Dev tools provide web/app developers deep access into the internals of their application to efficiently track down layout issues, set JavaScript breakpoints, and get insights for code optimization, networks stats and much more.

Debug Logs pCloudy provides direct access to the logs and debug information. You can install the app and debug via LogCat logs. Set custom filters or get crash dumps and stack traces.
All logs are saved automatically on Cloud Drive which helps to avoid reproduction of the issue multiple times and can be shared easily with the team members for further analysis, which helps in reduction of bug cycle.

Crash Reporting pCloudy instantly reports crash in any installed application with stack trace and one can further analyze the crash.

Session Videos pCloudy provides high resolution video of the device access session. One can record and share those videos for easy reproduction and resolution of the issues seen in the application. One can get complete video of the device usage to share and analyze installed app behavior step-by-step.

pCloudy is a mobile application testing platform designed to increase application testing coverage while saving significant time and cost.

Conclusion

pCloudy helps to test or certify mobile apps on real multiple Android and iOS devices using an intuitive web interface in the cloud environment. It provides the real-time streaming of a device and any action performed on it. It has many tool sets which will be useful for automation, debugging and data generation, it provides direct access of crash logs, video filing and other data. pCloudy provides an analytics-driven platform for manual and automated testing of apps on real devices, thereby providing a one-stop shop for all product app testing.

"pCloudy has the capability to become India's first mobile application testing platform on cloud. This has been a huge pain area for Indian application developers and testers as they need to test the compatibility of their apps across a variety of mobile devices and certify it."

Link which one can register themselves and explore free trail: https://www.pcloudy.com/.

Can find few more similar products at https://datafox.com/competitors/pcloudy-com.

Predictions

Written by **Keshavan Santhanam**, Sr Member Technical Staff 2, ZENworks

Data Science is a very diverse field that involves analyzing data and making predictions. We need to have expertise in Statistics, Machine Learning, and Programming, and still be incapable of predicting properly

Predictions are used to determine the outcome of an activity without executing it. There are two ways to predict – either by logical deduction, or by statistics. The latter is branched under Data Science.

To statistically determine an outcome, we need to first find out what data to collect, and methods to collect it. However, we do not need to do much to get data, if the problem we are trying to solve is based on our experiences. This is pretty much an unexplored field because geeks are busy breaking their personal records and those of machines.

Once we get relevant experiences in a problem domain, what we do next is to experience them over and over again. Beyond a point, we stop focusing on getting more experience and get into familiarity zones and stagnation. This results in a situation where we start living with our problems instead of solving them, and then we start lagging behind others and become specialists in negotiating certain kinds of problems. Finally, we get outdated.

Where did we go wrong? A simple analogy can explain it. Between earning money, and spending it, there is a whole domain called Finance Management. Yet, most people care about only earning and spending money. We do something similar with our experiences.

What else can we do with experience other than trying to be the most experienced guy in market?

Data Science says that we can also use our experiences to predict events.

Data Science is a very diverse field that involves analyzing data and making predictions. We need to have expertise in Statistics, Machine Learning, and Programming, and still be incapable of predicting properly. It looks like a tough field. However, it does not mean that all predictions are difficult to make.

The steps to predict are:

1. Create models to represent our experiences.
2. Share the models.
3. Use models to predict.

Models need not be accurate in all scenarios. Sometimes, it is not worth the effort to be accurate.

The purpose of the article is to develop an interest in developing predictive models for experiences.

This article shares predictive models for two interesting problems - Abuse and Social Media Addiction. The respective models can be called as '*Mentality for Abusers*' and '*Mentality for Social Media Addicts*'.

Mentality for Abusers

An Abuser is abusing a resource if it is used for purposes other than what it is meant for. A model would be presented that predicts mistakes before they are done.

We have plenty of choices before mistakes are done. It helps, if we can highlight our choices and pick the proper ones when we have the luxury. After the mistake is done, the focus is more on damage control rather than the actual problem. This way, we end up having more processes & rules that prevent problems, rather than processes which surprise us with success.

The model is very simple. It is a language agnostic but vocabulary richness helps a lot.

The Predictive Model is a formula: '*Using ? is not ?-ing*'

If '*using* ?' is one activity based on a word ?, '?-*ing*' is a related activity that involves using the result of the first. For example, Using Money is not Spending, or Using Experience is not Experiencing.

The magic is in how one of the activity that came from the above formula always turns out to be some kind of mistake in the real world that is categorized as abuse. Call it the practicality of human languages.

So, the model predicts a next possible mistake.

An interesting axiom is - 'A mistake is always an abuse of something'. This means that to rectify a mistake, we just need to find out what we abused. Mistake rectification is more actionable this way.

We were planning a study group. Following are some mistake predictions we could make:

1. Using study group is not studying in groups. Is it not the obvious activity of a study group? The model says that it is a mistake.
2. Using study is not studying. Studying somebody else's output is not studying. People should start with same inputs & get

their own outputs. Everybody's output needs to be compared with each other and merged - to create standards and solve compliance issues.

3. Using experience is not experiencing. We started above that we need to use our experience for something other than experiencing.
4. Using interest is not interesting. Proper use of interest in an activity is not very interesting.

While half of applications using the model give use-abuse type of related activities, the rest gives a use-create type of related activities. Let us rephrase the model slightly as - 'Using ? is not creating ?' for such words.

1. Using Tool is not Creating Tools. We prioritize tools based on usability, or ease of use. As per this model, this will not lead to the creation of useful tools.

We could go on and on, predicting all sorts of mistakes.

People may say that such conclusions are silly. The reason could be because we are used to realizing mistakes after they are done. It is worth noting that these conclusions can be made before a mistake is done using a dumb formula & they might not sound impractical at that time.

Mentality for Social Media Addicts

If you have ever tried developing & managing personal projects at home, you would find yourself very lacking. It is easy to blame lack of remuneration, environmental conditions, noise etc. for our failures, but the problem is deep.

With relatively more experience in industry than most other employees, I miss having managers who can re-prioritize my work to be more important than life. That just means that, I have to do that unpleasant task myself. Sometimes, it is health or family that is more important. Sometimes, it is responsibility as a citizen, or responsibility as some other identity, which people generally demonstrate on social media. The trouble starts when social media responsibilities become more important than health, family, and work.

No doubt, the problems that are discussed nowadays on social media are of very high priority, and things we just cannot ignore. So important that governments frame policies, and people can predict the future based on social media inputs. We cannot help, get the feeling that our calling has come when we type what we feel.

Because of social media, our priorities have changed from something static – '*What work is most important*?' to something negotiable – '*What work becomes most important*?'. Media always had the power of creating perceptions. Social media makes it worse. Our health, work, family are all unimportant compared to our responsibility to let people know what we feel on social media.

All of this is common knowledge. But, here is a model for social media addiction.

Which work becomes more important?

The one which offers more number of intelligent acknowledgments is more important.

I could add that relevant context, which is also important, but that is just being unnecessarily accurate.

With this information, I can predict why some work has become so important for me.

The basis for this mentality is as follows. We live in a world where we get plenty of unsolicited, un-understandable & curt acknowledgements. The motor of a vehicle or generator wants to acknowledge that it has got its fuel, and does so with in an unjustifiable tone, with very lengthy words & sentences. The driver wants to acknowledge his impatience with others ahead of him, by broadcasting his honking. The kids in house want to acknowledge all this disturbance by taking it to a new level. Our poor ears are not designed for such abuse. Why does everybody have to send their acknowledgements to us? We hardly have any context for the acknowledgements we get, and we did not ask for it either. This creates a deep longing for work that involves intelligent acknowledgements. Social media provides an escape from this harsh reality by connecting people with intelligent acknowledgments and relevant contexts. That is why people love to work on social media.

They could be currently trending Q&A (as in Quora), or whatever is on the mind (as in Facebook). We had ORKUT in the past which failed because it restricted contexts and acknowledgments to within groups. Google's answer to Facebook failed because the overall feel is complicated and artificial - hardly a solution to our complicated problem.

In our work, acknowledgment generally means praise or certification for work already done. In-progress work does not count as acknowledgeable. We need to allow for in-progress acknowledgments, if we want to avoid our work from getting de-prioritized by social media.

Obviously, this is not our manager's responsibility. Let us see how we can acknowledge ourselves:

1. We can prioritize work that is very intellectually satisfying to be done first. This just means doing work that gets a lot of self-acknowledgment, and then, hope that we are satisfied enough to do other tasks which are less interesting.
2. Interestingly, TDD acknowledges in-progress work. We prepare our acknowledgments in advance, and they dispassionately certify our work, and therefore on completion! TDD is one answer to social media addiction at work.

Given the importance of TDD, we need to fix somethings in our TDD implementation. One of an important aspect of TDD is the emphasis on identifying test cases before development. However, we never feel justified doing this because it does not seem to make much of a difference. We are missing out a lot if we do not do it. No doubt, the test cases we identify will be incomplete or sometimes useless. But that is because we have only tried predicting test cases without using models. Identifying models that will predict test cases would be an interesting enhancement to our TDD implementation.

Conclusion

One of the biggest blocks to changing status quo is our inability to use our memory faculty effectively. We cannot use our experiences properly if we do not know how to use our memory in the first place.

Too much of delegation of remembering to tools like Editors (where mistakes are no longer permanent blots), IDEs (where searching is easier than remembering), Google Search (because of which we do not have to remember anything), Event Managers, TODO lists etc. are literally damaging our career.

The below situation presented by somebody on Quora aptly describes a really bad case of this problem.

https://www.quora.com/I-get-disturbed-for-small-things-which-are-not-happening-the-way-I-want-How-to-convince-my-mind-to-be-Ok-with-the-way-things-are-happening/answer/Keshavan-Santhanam

Project Majenta

Written by **Ashwin Venkatesha**, Software Engineer, Access

An open source project from Google based on TensorFlow, aimed at bringing together the contributions of artists and machine learning enthusiasts to create unique tracks

An intriguing question to begin with, have you ever imagined a computer pick a Beatles song and combine that with a guitar solo to create a unique track? Well, the answer is 'Project Majenta', an open source project from Google based on TensorFlow, aimed at bringing together the contributions of artists and machine learning enthusiasts to create unique tracks.

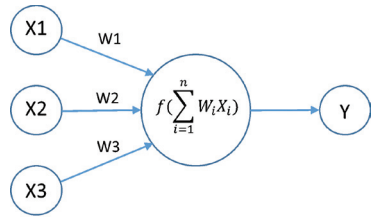
In this article I will try to elaborate about the fundamentals of neural networks and the modelling techniques used by Project Magenta, and how one can use it to produce melodies.

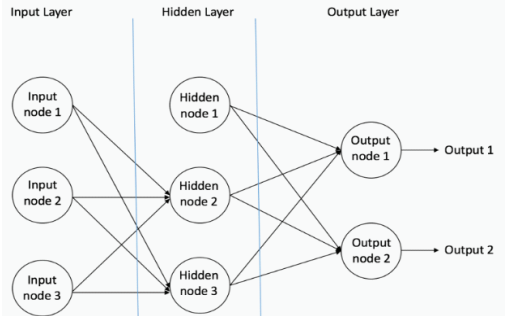
Neural Network

First, let us understand the concept of Neural Network which is a computer model that is inspired by the way a biological neural network in the human brain processes information. The basic unit of computation in a neural network is the neuron, often called a node or unit. It receives input from some other nodes, or from an external source and computes an output. Each input has an associated weight (w), which is assigned on the basis of its relative importance to other inputs

Feedforward Neural Network

The feedforward neural network was the first and simplest type of artificial neural network devised. It contains multiple neurons





(nodes) arranged in layers. Nodes from adjacent layers have connections or edges between them. All these connections have weights associated with them.

An example of a feedforward neural network is shown in the above figure.

In a feedforward network, the information moves in only one direction – forward – from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.

Recurrent Neural Networks (RNN)

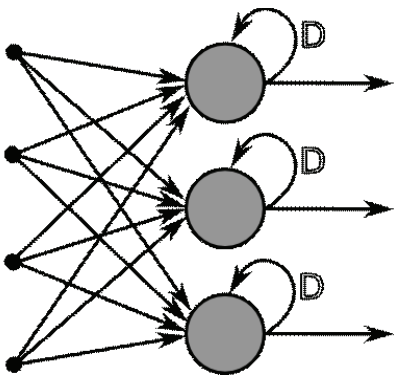
Humans don't start their thinking from scratch every second. As you read this article, you understand each word based on your understanding of previous words. You don't throw everything away and start thinking from scratch again. Your thoughts have persistence.

Traditional neural networks can't have persistence, and it seems like a major shortcoming. For example, imagine you want to classify what kind of event is happening at every point in a movie. It's unclear how a traditional neural network could use its reasoning about previous events in the film to inform later ones.

Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist.

LSTM Networks

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term



dependencies. Remembering information for long periods of time is practically their default behaviour, not something they struggle to learn. To have a deeper understanding about LSTMs, I strongly recommend going through http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Project Magenta's use of LSTM

Remember the task at hand is to generate music given an input. Music is generated using scores of notes and there is a dire need for context based knowledge than just persisting information that is recently accessed, this is where traditional RNNs fail to make the cut. Majenta uses LSTM to address the memory problem. LSTM is known for, when to forget its current input and when to remember it, it does not struggle to learn.

Majenta uses Melody RNN (LSTM) which comes with three configurations:

- Basic - This is the baseline for melody generation with an LSTM model.
- Lookback RNN-In contrast to Basic RNN, this model introduces custom labels, which enable model to remember what is only required.
- Attention RNN- This model can access past information without having to store it, making it the ideal candidate for long term dependency problems.

The first four notes of Twinkle Twinkle Little Star, fed to these models and the team at Google were surprised that a sensational track was composed. This track was later improvised by Jason Nyugen, on the đàn bãu and Alex Koman, on guitar to create a unique man and machine collaboration (https://www.youtube.com/watch?v=Aq3370Prbi4)

Let us look at how to use Magenta to generate melodies,

- Magenta environment needs to be set up – all necessary dependencies need to be downloaded.
- Pre-trained models to generate music- basic rnn, lookback rnn or attention rnn models that have been trained on thousands of Musical Instrument Digital Interface (MIDI) files have been made available.

```
melody_rnn_generate \
--config=${CONFIG} \
--bundle_file=${BUNDLE_PATH} \
--output_dir=/tmp/melody_rnn/generated \
--num_outputs=10 \
--num_steps=128 \
--primer_melody="[60]"
```

This generates a melody starting with a middle C note. '—primer_melody' can be replaced with '–primer_midi' to prime the model with a melody stored in the filesystem.

- Training our own models to generate music
  - Initially MIDI files have to be converted to a fast and efficient data format called NoteSequences
  - Next, SequenceExamples are generated by extracting melodies from NoteSequences. Each SequenceExample is fed as an input to the model during training and evaluation.
  - Once the model is trained and evaluated, tensorboard can be used to view the evaluation and training data.
  - To generate music a primer melody such as, --primer_melody="[60, -2, 60, -2, 67, -2, 67, -2] can be used that primes the model with first four notes of Twinkle Twinkle Little Star.

References

1. http://colah.github.io/posts/2015-08-Understanding-LSTMs/
2. https://magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn/
3. https://github.com/tensorflow/magenta
4. https://github.com/tensorflow/magenta/tree/master/magenta/models/melody_rnn

bits & bytes

Dont get mislead about WannaCry

its the ETERNALBLUE/DOUBLEPULSAR that needs to worry you

All of us must have read about WannaCry ransomware that took majority of the media share in the last few days. While it is important to be vigilant about this and protect our systems from this particular virus, most of the important information is getting lost in the details. Imagine if you have a house and it has multiple doors with numbered locks. You never knew that these locks had a secret code other than your PIN to open it. A mischievous gang knew about this secret PIN and posted this in their twitter. By then, the poor manufacturer knew about this and announced a public recall with a free upgrade to a better lock. Some used the benefit and installed the upgraded lock while most people either ignored or were unaware. In the mean time, a robbery gang used this secret PIN to enter houses and started installing a micro drone that can take remote instructions. They started leveraging the drone to do what ever they want in the impacted households. In this story, the latch having problem is ETERNALBLUE while the drone is DOUBLEPULSAR ; WannaCry is just one gang. There can be many such gangs planning how to create havoc with their version of drone in your house.

What is important? Upgrading the latch, searching for the drone in your house or telling everybody you are not impacted ? Isn't it common sense?

This is far from over unless everybody fixes their problematic locks and survey for any suspicious drones in their houses.

Ransomware – The most profitable malware in the history of cybersecurity!

Written by **Arun Paul**, Sr Member Technical Staff 1, Security

Ransomware was looming around as a threat for years, it has recently been elevated to the most profitable malware in the history of cyber security

Ransomware, Phishing and Spyware has been the most trending cyber-attack in the recent times. Although Ransomware was looming around as a threat for years, it has recently been elevated to the most profitable malware in the history of cyber security. Primarily working based on data-encrypting malware, there are many variants of this threat – CyptoLocker, CryptoWall, Locky to name a few.

Definition of Ransomware

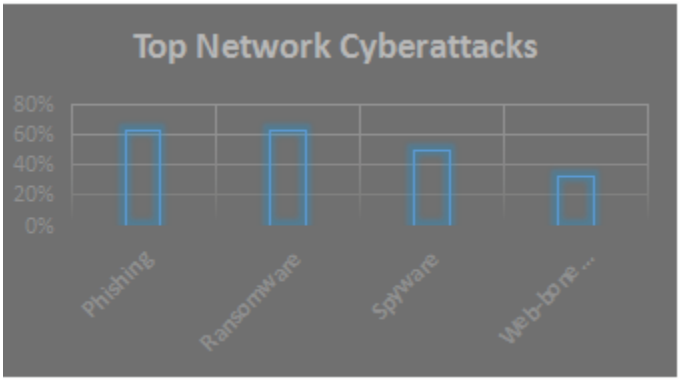
Ransomware is computer malware that installs covertly on a victim's computer, executes a cryptovirology attack that adversely affects it, and demands a ransom payment to decrypt it or not publish it. A ransomware can encrypt the computer's Master File Table (MFT) or the entire hard drive. Thus, ransomware is a denial-of-access attack that prevents computer users from accessing files since it is intractable to decrypt the files without the decryption key. Ransomware attacks are typically carried out using a Trojan that has a payload disguised as a legitimate file. (Source: Wikipedia)

Why Ransomware Flourished

Following are few of the top reasons why Ransomware has flourished in a short time:

1. **Ease of Operation** – One of the primary reasons for the tremendous success of ransomware is the ease at which the threat actor can execute a ransomware attack.
2. **Ease of Payment** – With ransoms being paid via BitCoins and TOR for anonymity, the extortion mechanism has become easier for the attacker.
3. **Ease of Mutation** – Designing new variants is much easier in Ransomware due to the various methods in which the attack can be made. This also helps the attacker evade early detection from typical security measures.
4. **End-Point-Security/DLP slippages** – There has been a great lot of improvement in DLP solutions but still it focuses at data leaving the network and not at Data-at-Rest. Ransomware on the other hand works on how the Availability of data/asset can be disturbed.
5. **Human emotions at play** – The attackers know that in most of the cases the human emotions can be used to their advantage. This includes the urgency of getting the information back, the loss of reputation or goodwill, which gives the attackers an added advantage.

Ransomware netted attackers $209M in Q1, 2016 (source: FBI)



6. **Time to Recovery** – For critical services like Healthcare, the time to recovery carries more weightage than the financial cost.
7. **Cost of Ransom Vs Recovery Cost** – Some times the cost of Ransom will be cheaper and faster than the cost of attempt to recover the data, which includes disruption of operation, risk of hiring 3rd party professionals to do the job and other business complications, especially in cases where the sheer volume of the data to be recovered is huge. The average compensation for ransomware is about $300, as of 2015 (*source*: Kaspersky).
8. **Prioritization of HIPAA over Cyber-Security in Healthcare Industry** – The HIPAA is a mandatory requirement for Healthcare industry and it has reached a certain level of maturity. However, cybersecurity in this industry has always taken a second stage and hence the Healthcare industry has become one of the easy and primary targets for Ransomware actors (*source*: ISC2).

Attack Vectors in Ransomware

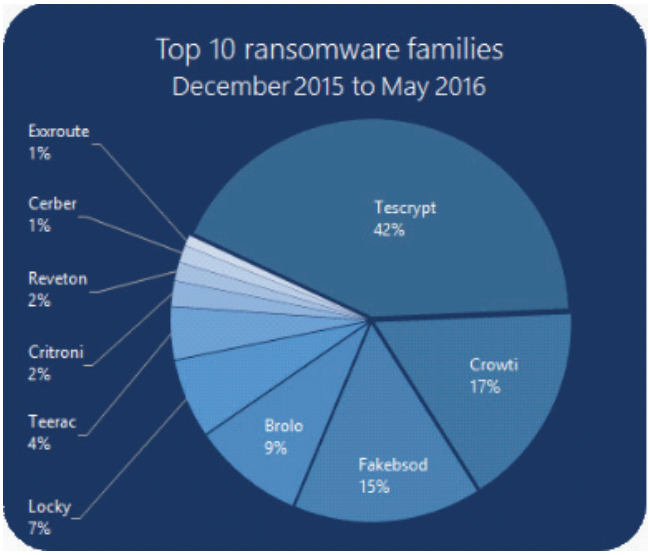
There are three Attack Vectors tactics used in majority of the Ransomware scenarios. This is the first step in a Ransomware infection

1. **Exploits** – Exploits are usually in the form of pdf file, webpage, designed to leverage software flaws, bugs in application and eventually compromise the end point.
2. **Macros** – The second type of attack vector are Macros. Usually embedded in MSoffice documents, xls. Once activated, they download additional malware or the ransomware itself and compromise the endsystem.
3. **Executable files** – A range of variants here from the .exe extension onward to harmless looking pdf files. Once the code is run, it will cause the end point to be infected.

Safeguarding against Ransomware

Following are few standard practices that can be followed to minimize the risk of a Ransomware scenario happening in your organization.

1. **Strategy for Defense** – Humans are always the weakest link in Information Security. There is a famous saying that "There is no patch for Ignorance". This bring the focus back on how critical use awareness trainings are with respect to compliance and regulatory requirements and also as a first step in the organizational defense strategy.
2. **Strong Backup Policy** – Backups are the foundation of your recovery strategy and is a crucial part in the BCP DR plan. One of the considerations in Backups form the Ransomware perspective is to avoid keeping the backup within local access to the data-sources (eg server). This is to avoid the infestation of the backup itself, following the compromise of a data-source.
3. **Strict policy enforcement on use of Macros** – Most of the software have evolved to disable auto-execution of macros in their code. Most of the malwares written in macros will prompt for the user response to enable macros. This is a part of user training and awareness programs.



4. **Strict policies on Email and associated file attachments** – More than defining a proper guideline and procedures for IT best uses, the users need to be educated on how to handle an event of a suspected attachment in a corporate email. We can opt to white-listing or black-listing the file types per department.
5. **Least Privilege enforcement** – The lease privilege is a best practice that need to be extended to all the Authorization policies across all functions in the organization. Privileged User Monitoring and Privilege Escalations are also areas which need proper monitoring policies in the organization.
6. **Patch Management** – Patch early and Patch often. As discussed before, we need to lessen the gap between the declaration and availability of a patch and its installation, which need to be continuously monitored and improved with respect to automation, policy and procedure and enforcement.
7. **File Server Monitoring** – Most of the business critical files in the organization will be saved on file servers or cloud services. We can monitor the spike in volume of this event based on the known baselines. This is one typical behavior of a Ransomware attack.
8. **System Hardening** – Few of the easy and must do system hardening types are mentioned below –
   - Disable macro execution –if it's a file downloaded from the internet.
   - Harden the browser by group policies. Eg plugins are not to be executed automatically, Plugin should not be enabled if it is outdated.
   - Software updates – Strict monitoring software patches and vulnerabilities and enforcement of the same to reduce the attack surface.

Ransomware Simulator

RanSim is a tool that can simulate the actual behavior and attack tactics of a typical Ransomware. This tool can be used to check on your end-point security software in its capability to detect, prevent and report a ransomware attack in real-time. This is a controlled Ransomware stimulator which can be installed to test your system.

URL: https://knowbe4.zendesk.com/hc/en-us/articles/229040167#TEST

Top 10 Ransomware families

Image courtesy: Microsoft

Rise of the Machines

Written by **Girish Kambhampati**, Sr Member Technical Staff 1, WPG

. . . Internet of Things may gradually evolve into Internet of Everything, in future

Each recent century is marked with a scientific innovation that changed the lives of people – in areas like transportation, electricity, industries etc. The invention of computer and further developments in computer technologies and the innovations in networking and telecommunications happened independently for some period. Once they started merging together and gave birth to Internet, the lives of people changed dramatically and this single innovation out-performed all the other innovations that happened in the previous centuries. Till recent years, Internet knowledge has been confined primarily with computers and only people having computer knowledge were able to take more advantage out of it. With the Internet encompassing mobiles and gradually spreading into many other things at home, office or outside - we are going to see drastic revolutionary changes.

While machines have been gaining intelligence from more than a century, generally machines of olden times either operate based on the way the machine had been programmed to operate or based on limited ways of taking instructions or inputs. With the advancement in modern day computation and machine learning capabilities, machines are getting more intelligent and are capable of taking decision themselves. More interestingly, With IoT, machines are gaining outside world knowledge and they can also provide knowledge to the outside world. Machines gaining ability to act based on outside knowledge had brought tremendous benefits from convenience of remotely controlling utility devices at home to the improvement of quality of life of huge population of smart city by smartly managing the assets of the city. Internet traffic generated by things (non-PC devices) has become about 40% by 2014 and there are some estimates that predict that the traffic could reach about 70% by 2019.

Present and Future

The developments happening in the domain of IoT opened up many areas to be explored for research communities and also opened up many potential business opportunities for business organizations. Two areas in which IoT was successfully utilized are the "Smart Cities" and "Home automation". Certain countries utilized the technology of IoT to manage city assets effectively such as schools, libraries, transportation systems, hospitals, power plants, water supply management, waste management, law enforcement and other such services. Barcelona established a number of projects that utilize IoT – a sensor technology in irrigation system that transmits real-time data to gardening crews about the level of water required for plants, implementation of smart traffic lights such that buses run on routes designed to optimize number of green lights, emergency vehicles transmit the emergency route information to traffic lights and traffic lights adjust to ensure less traffic obstacles to emergency vehicles.

There are many other areas that can utilize IoT based technologies to come up with new products that can improve efficiencies and communication.

- Like with any other network based technologies, privacy and security remain the primary concerns for expansion of IoT. As the security and privacy gets enhanced more and more, IoT become more and more ubiquitous.
- Google started a company named "Waymo" in 2016 after successfully developing a driver-less car and successfully driving on public roads in 2015. They are working to make it completely safe and easy for people and things to move around. The company targets to make self-driving cars available to public by 2020.
- Advanced medical diagnosis and remote patient medical monitoring are possible thru IoT. Many medical companies are doing researches in developing products based on IoT.
- Industrial automation has got a huge potential for IoT technologies to help them in manufacturing, process controls, optimizing plant safety and security and other such areas.
- Since devices that implement IoT can be of different types, the communication protocols that suit different types of devices are being developed. Standardizations are yet to happen after interoperability requirements get more clarity in future.
- Since IoT generates huge amount of data, the data storage technologies have to adopt to meet the specific requirements of IoT generated data.

Conclusion

IoT turned many fictional things of the old fantasy movies into reality and this is definitely a good "Rise of Machines" without any reason for the fear of "Terminator3" becoming reality. Internet of Things may gradually evolve into Internet of Everything, in future.

Reference

https://www.internetsociety.org/sites/default/files[ ]/ISOC-IoT-Overview-20151014_0.pdf

RxJava

Written by **Md Majid Jahangir**, Software Engineer, Access Group

RxJava not only familiarizes one with reactive programming but also with several concepts of functional programming

Sometime ago, I had an opportunity to work on the Access Manager Analytics Server where I worked on optimization of event data aggregation from Sentinel for graph visualization. Here, I got introduced to a reactive programming model for composition of asynchronous callbacks from our component.

RxJava is a JVM implementation of Rx (Reactive Extensions), an open source project of Netflix, which strikes out the problems that plague software development like multithreading, exception handling, concurrency, event driven model, and so on. It also helps in making the code more compact and easy to extend. It is a library that is extensively used in android development and developing asynchronous and event driven programs by using observable sequences. It extends the observer pattern and provides a collection of operations to filter, map, transform and compose observable sequences. It uses thread safety, low-level threading and many aspects of threading and concurrency.

The main building blocks of RxJava are **Observables** and **Subscribers**. The Observable data type is equivalent to Iterable with one major difference. An Observable imitates push model whereas Iterable is based on pull model. In Iterable, the consumer pulls value from the producer thus blocking the call until the data is produced, unlike in Observable, where the producer pushes the value to consumer when the data is ready, thus making it more feasible for event driven applications. So, in short, an Observable emits an item with a Subscriber ready to consume it.

```
Observable<String> myObservable = Observable.create(
    new Observable.OnSubscribe<String>() {
        @Override
        public void call(Subscriber<? super String> sub) {
            sub.onNext("Hello, world!");
            sub.onCompleted();
        }
    }
);
```

```
Subscriber<String> mySubscriber = new Subscriber<String>() {
    @Override
    public void onNext(String s) { System.out.println(s); }

    @Override
    public void onCompleted() { }

    @Override
    public void onError(Throwable e) { }
};
```

Figure 1 (a) Observable emitting data, and (b) Subscriber consuming it.

In the example below an observable is created to emit “Hello world”, and a subscriber is created to consume data.

The first half of the example focuses on creating an observable using create method. The create method will receive an implementation of Observable.onSubscribe interface. The implementation defines what action will be taken when a subscriber subscribes to the observable. However, the action will be invoked only when someone subscribes to it. Inside the action, we define the events that needs to be pushed to the subscriber using onNext() and onCompleted(). In the example, we push the text “Hello, world” to the subscriber via onNext event.

The second half of the example focuses on creating a subscriber that will respond to the events being emitted by the observable. It is an abstract class and the implementations of the three basics events of observables namely onNext, onCompleted and onError are subscribed. In the example, since “Hello,world” is being pushed using onNext event and the goal of snippet is to print it, our Subscriber implementation prints the text inside onNext implementation.

As shown in Figure (1), any observable calls onNext() for each item it emits, onError() for any errors, and onComplete() on no data availability from the observable.

Important: Unlike observer pattern, observable does not start pushing items unless someone explicitly subscribes to them. Hence the following snippet is important:

To make this snippet more compact and easy to maintain, let us use

```
myObservable.subscribe(mySubscriber);
// Outputs "Hello, world!"
```

Java 8 lambdas and remove the entire boilerplate to the following one liner:

```
Observable.just("Hello, world!")
    .subscribe(s -> System.out.println(s));
```

In addition to this, RxJava provides loads of operators to modify the items emitted by observables like maps and filters before they get operated on subscribe (reference GitHub or wiki page). These operators can be applied to observables in the chained form which resemble mostly like builder pattern.

The most crucial aspect of RxJava is the ability to handle concurrency. You need to consider the following before you start working on RxJava:

1. The thread that declares the subscription (invocation of subscribe()) is the thread that pushes the emission. So, the above "hello world" example is run entirely on main thread. To overcome that, before subscribing the observable, we can invoke a subscribeOn() on a thread pool or thread to let the emission and operations on subscription to happen on different thread.
   In the following example, the snippet will print the emitted text on main thread.

```
public static void onClick(String[] args) {
    //this is UI thread
    Observable<String> source = Observable.just("I just got clicked");

    source
        .subscribeOn(someThread)    //this moves exec to other thread for network call
        .doOnNext(networkCall())
        .observeOn(uiThread)        //coming back to UI thread when the network data is ready
        .subscribe(i -> System.out.println("Received " + i + " on thread "
                + Thread.currentThread().getName()));
    Thread.sleep(3000);
}
```

   Whereas, if you want to move the execution to another thread, you can use subscribeOn to move it to other computation threads like the one below:

```
public static void main(String[] args) {

    Observable<String> source = Observable.just("My software never has bugs. "+
                        "It just develops random features.");

    source
        .subscribeOn(Schedulers.computation())
        .subscribe(l -> System.out.println("Received " + l +
            " on thread " + Thread.currentThread().getName()));
    Thread.sleep(3000);
}
```

2. Similarly, after the emissions start on a separate thread, one may want to switch further operations to another thread to offload some intensive task that may block the emission thread. This can be achieved using observeOn(). For example, you have a UI thread that has a setting which performs network call, gets the result, and returns it to UI thread.
   The below snippet explains it better,
3. The most important point - Achieving Parallelization:
   The most common mistake that anyone makes is - using subscribeOn() will make Observable emit items on multiple threads. Using subscribeOn does move the task to a separate thread. All emissions from observable will emit on a single separate thread in succession. So how do we overcome it? Well, perform a flatMap() operation on the observable, create an observable and then perform all operations within flatMap.

So, in the below example one can see an observable emitting 10 integer values which will be eventually passed to networkCall(). In order to perform all these 10 network calls on 10 separate threads, for each item inside flatMap, we create another Observable and perform a subscribeOn to schedule it for separate threads.

```
Observable<Integer> vals = Observable.range(1,10);
vals.flatMap(val -> Observable.just(val)
            .subscribeOn(Schedulers.computation())
            .map(i -> networkCall(i))
).subscribe(val -> System.out.println(val));
```

RxJava not only familiarizes one with reactive programming but also with several concepts of functional programming.

bits & bytes

Messaging Protocol for Internet of Things: MQTT

MQTT is a publish/subscribe-based lightweight messaging protocol for Machine to Machine (M2M) communication, on top of the TCP/IP protocol. The protocol provides telemetry technology, and MQTT developers are working to connect the evolving internet world, which is expected to produce even more diverse smart devices. The first version of the MQTT protocol was authored by Stanford-Clark, IBM, and Arlen Nipper.

Why MQTT?

MQTT has been used by Facebook for their messenger application, which needed a persistent connection to their servers without killing battery life. It requires a low network bandwidth and has a small code footprint. It transmits data over widely distributed and sometimes intermittent networks. These features translate as advantages for remote devices with little memory and processing power.

Other notable features of MQTT are:

- It's open source, royalty free and therefore easy to adopt and adapt
- It follows a publish/subscribe model for one-to-many distribution
- Small message headers
- Multiple Quality of Service levels
- Simple command messages
- Data type agnostic
- Retained messages
- Clean sessions and durable connections
- Last Will and Testament (LWT)

Secure by Design

Written by **Harippriya Sivapatham**, Sr Member Technical Staff 2, Access Manager

This article is a concise list of various Security Principles that would help to build a secure product

Most organizations have a well-oiled machine with the sole purpose of building functional software. Security is rarely a criteria while building a product. However, the increasing concerns/business risks associated with insecure software has increased the awareness for integrating security into the development process. *Secure Development Life Cycle* (SDL) is gaining a lot of traction in the industry and also within Micro Focus.

Security has always been an afterthought and SDL intends to change that. Architects and developers play a vital role in the success of SDL. It is very efficient and easy to rollout SDL if products are already built with security in mind. This article is a concise list of various *Security Principles* that would help to build a secure product.

Secure Design

Security Architecture is one component of the product architecture. It addresses where and how the security of the components must be handled. Some generic high level security principles to consider while designing are:

Modularize and secure access: Modularize your system and secure access to every module. This loose coupling helps with reuse of modules and also introduces multiple layers to security.

Separation of Duties: It is a classic security method to prevent fraud and from giving too much power to one individual. For example, an administrator should not be able to change the audit settings for himself.

Implement CSRF prevention: Cross Site Request Forgery (CSRF) is one of the top 10 OWASP vulnerabilities. An infrastructure must be put in place to support random CSRF tokens to be managed and validated for each request.

Reuse proven security implementations: Do not implement your own security mechanism like encryption algorithms, cryptography, etc.

Implement layered security: Avoid single point of failure. Support multiple levels of security, so that it is not easy to exploit a system.

Secure Development

Validate Input

- Input validation mitigates some of the Top 10 Vulnerabilities listed by OWASP:
  - Injection attacks – Command Injection, LDAP Injection, SQL Injection, etc.
  - Cross Site Scripting (XSS)
  - Directory traversal attacks
  - Un-validated forwards and redirects
- Input validations must be performed on the server.
- Use whitelists -- Validate against a "whitelist" of allowed input.

For example, prevent Directory traversal attacks by allowing only accepted paths or filenames, instead of checking for unaccepted character sequences like "../../".

Sanitize Output

- Sanitization of data removes harmful characters and then encodes the data.
- Sanitize on the server side and encode all data returned by the server.
- Prevents XSS attacks.

Secure sensitive data

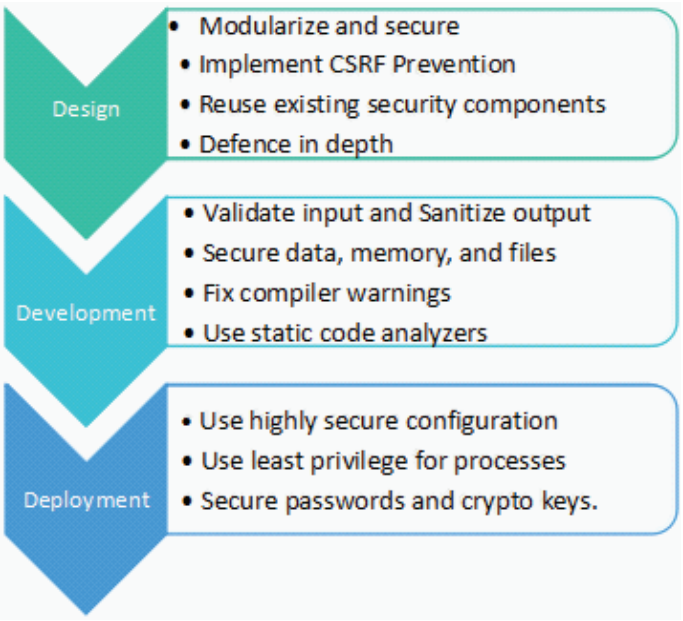
- Do not disclose session IDs, passwords, actual file path, etc. in error messages, debug logs or URLs.
- Do not display debug or error stack traces on the UI.
- Customize error pages to display generic messages.
- Remove unnecessary response headers with information about OS, application framework, web server, etc.

Secure Memory

- C++ : Guard against buffer overflow
- Java: Close resources explicitly. Also, do not use String variables to store passwords. These could be obtained by doing a java memory dump. Use byte arrays instead.

Secure Files

- Save uploaded files in a location outside of the web context.
- Turn off execution privileges on the file upload directories



- Do not accept filenames for well-known files. Use IDs instead.
- Do not send absolute file path to client.

Miscellaneous

- Fix compiler warnings
- Run static code analyzers and fix vulnerabilities reported by them.
- Do not hard code passwords.

Secure Deployment

- Configure to use higher TLS version, stringent ciphers, better key size, etc.
- Run all processes with least privilege possible.
- Properly secure passwords and crypto keys.

Conclusion

This article is designed to be a quick reference for the security principles to be considered during various phases of a product life cycle. It is a collection of insights and best practices gathered from various sources and from hands-on experience of fixing various issues reported by security testing tools like Burp, Nessus and Metasploit.

Security is a vast topic. Open Web Application Security Project (OWASP) is a good place to start. It lists the top security vulnerabilities and provides various projects to help mitigate them.

bits & bytes

Why it's time to start planning your marketing around machines?

Algorithms have taken over who we connect to, what we buy, where we go, how much we earn and even who we marry. And marketers who don't start hatching a plan for how they're going to tap this new phenomenon are at risk of giving competitors an unassailable advantage, according to We Are Social.

Facebook, Google Maps and dating apps such as Tinder are all powered by algorithms today, while on the marketing front, programmatic advertising and social media marketing are both examples of algorithms in action.

Algorithms have been helped on their way by the proliferation of connected devices, the pair said. The reasons why we're growing so reliant on them is because they help us manage choice, improve our confidence, increase efficiency by automating lower interest tasks, provide a source of inspiration, and make difficult decisions on our behalf.

As algorithms grow in scope, marketers will spend less time trying to persuade people to do things, and more time trying to influence the algorithms influencing the people.

As an example, voice-controlled devices in the home such as Amazon's Alexa and Google Home. These could start to impact the product and services we choose and challenge the brand status quo.

The device will ask the consumer; it will fall back on recurring choices and the brand you bought last time; it will select a brand based on social recommendation engines; or consumers will just let the platform decide based on a number of algorithmic inputs, such as IoT sensor information, what they read and ask about, behavioural attributes and Facebook likes.

Shielded Virtual Machines

Written by **Raju Thimmappa**, Sr Member Technical Staff 1, Data Center

Protecting any virtual machine (VM) from malicious software is must . . .

Today everything is in the virtual world. Virtual machines help you in building applications and infrastructure.

Protecting any virtual machine (VM) from malicious software is must, at the same time it's also required to protect the same VM from a compromised host.

Your VMs are attached through the fabric of a hypervisor whether it's Hyper-V, VMware, Xen, KVM or any other.

A VM is just a file, it's easy to migrate, backup and replicate; so your VM can be easily copied and carried outside the network through the means of something like a USB stick!

Virtualization requires every one of us to think differently about the security of our VM infrastructure and applications

Shielded VM is the way to protect your VM on hypervisor. Windows Server 2016 Hyper-V has come up with Shielded VMs (Generation 2 VMs) using the combination of Secure Boot, BitLocker encryption, virtual Trusted Platform Module (vTPM) and the Host Guardian Service.

Shielded Virtual Machine

The attacker can steal your VM images and run them on their own hypervisor, so it is very much important to shield your VM.

It's not only 'any user' from which your VM is protected but also from the administrator! An administrator can do everything on the system. Anything you do to encrypt or protect a VM, the administrator can easily undo.

Example: You created a VM and provided vTPM. With virtual TPM, the administrator could still find those keys in memory and decrypt the VM.

This is possible on all platforms, VMware, Hyper-V, Xen, KVM, etc. What is the solution then?

- Safeguard the VM in such a way that it can run only on that hypervisor where it belongs to
- Bind the VM to base fabric ('fabric' is nothing but the resources of the hypervisor with which its built ,including the base hardware)
- Protect the VM from rogue admins

Windows Server 2016 introduces Shielded VMs to provide all the above solutions.

Shielded VMs protect virtual machines from compromised or rogue administrators in the fabric and encrypt disk and state of virtual machines so only VM or tenant admins can access it.

In addition to the above Microsoft is also protecting the fabric using the new feature the *Host Guardian Service*.

Whenever a shielded VM is powered on, the Host Guardian Service (HGS) first checks to see if the host's fabric is allowed to run the Shielded VM or not. HGS does the job of verifying whether the hosts (hypervisors) are allowed to run a shielded VM and/or they are in a perfect condition to run it.

Shielded VMs require Windows Server 2012 or Windows 8 or later, and they will not run unless the Hyper-V host is on the Host Guardian Service. When you start any VM using the Windows Server 2016 Hyper-V manager, the Hypervisor itself undergoes rigorous health attestation process.

New Shielded VM can be created within the Azure Pack management portal and also Microsoft is giving the option to convert existing VMs to shielded VMs.

Actually Microsoft has utilized the mechanism of software licensing concept used by software licensing manufacturers to identify the licensed VM on any hypervisor. The licensing software mechanism first gathers the information of the VM and its bare metal (hypervisor). Then a unique bare metal ID is created so that the license assigned to a VM cannot be misused by any means (but still the licenses are getting cracked!)

Microsoft has leveraged a similar hardware finger printing mechanism as that use by software licensing protection. The fingerprint includes information about the VM and the Hypervisor it is running on.

In case of software license protection mechanism, the information of the VM and its bare metal (hypervisor) is collected first and then a unique bare metal ID is created so that the license assigned to a VM cannot be misused by any means.

Now **Microsoft** has taken a new directional all together with this Shielded VMs in VM Security race. **VMware** has vSphere 6.5 virtual Unified Extensible Firmware Interface (UEFI) secured boot for hosts and guests, plus encryption for VMs in motion. Currently only VSphere can do TPM, but VM can't do virtual TPM of guest (VM) is not possible. This gives Microsoft an edge over VMware.

Conclusion

Microsoft has improved the security aspects of VMs, to protect the workflow and build secure business environments. At this point of time with Windows Server 2016 Host Guardian Service it seems to be difficult to fabricate a VM even as the Administrator!

TensorFlow

Written by **Khadija Chowdhry**, Associate Software Engineer, EPM

. . . where researchers meet developers in the machine learning world

A brief history: Every single Google product uses machine learning in some way, be it Google Voice Search, Google Images, speech recognition, recommendations etc. Google needed machine learning to take advantage of their vast amount of data sets and TensorFlow came into being to help in research, development, and deployment of machine learning models.

Originally developed by the Google Brain team for Google's research and production purposes, TensorFlow was later made open source. This library of algorithms originated from Google's need to instruct neural networks to learn and reason similarly as humans do, so that new applications can be derived which can "think" and "behave" like a human. TensorFlow has sought out to bring this research into a single platform so that the same tool sets can be used to collaborate on projects and improve efficiency.

The library: TensorFlow provides multiple APIs. The lowest level API--TensorFlow Core-- provides complete programming control. TensorFlow Core is recommended for machine learning researchers and others who require fine levels of control over their models. The higher level APIs are built on top of TensorFlow Core. These higher level APIs are typically easier to learn and use than TensorFlow Core. In addition, the higher level APIs make repetitive tasks easier and more consistent between different users. A high-level API like tf.contrib.learn helps to manage data sets, estimators, training and inference.

How it works: In this programming system, numerical computations are represented using data flow graphs. Nodes in the graph (called ops) represent mathematical operations, which can be as simple as addition/multiplication to some complex mathematical computations. Graph edges represent the typed multidimensional data arrays called tensors.

TensorFlow:

- Represents computations as graphs.
- Executes graphs in the context of Sessions.
- Represents data as tensors. Only tensors are passed between operations in the computation graph (more on tensors covered later).
- Maintains state with Variables. For example, the weights for a neural network can be stored as a tensor in a Variable. During training this tensor is updated automatically by running a training graph repeatedly.
- Uses feeds and fetches to get data into and out of operations in the graph.
- A feed temporarily replaces the output of an operation with a tensor value. It is fed as an argument to a run() call.
- To fetch the outputs of operations, the graph is executed with a run() call on the Session object with tensors passed to it.

Tensors: The central unit of data in TensorFlow is the tensor. A tensor consists of a set of primitive values shaped into an array of any number of dimensions. A tensor's rank is its number of dimensions. Here are some examples of tensors:

```
3 # a rank 0 tensor; this is a scalar with shape []
[1. ,2., 3.] # a rank 1 tensor; this is a vector with shape [3]
[[1., 2., 3.], [4., 5., 6.]] # a rank 2 tensor; a matrix with shape [2, 3]
[[[1., 2., 3.]], [[7., 8., 9.]]] # a rank 3 tensor with shape [2, 1, 3]
```

The Computational Graph

TensorFlow Core programs consist of two discrete sections:

1. Building the computational graph.
2. Running the computational graph.

A *computational graph* is a series of TensorFlow operations arranged into a graph of nodes.

Example 1: Creating an adder which adds two input variables and a constant.

We build a simple computational graph using placeholder nodes that accept external inputs and a constant node. A placeholder is a

```
>>>
>>> import tensorflow as tf
>>> a = tf.placeholder(tf.float32)
>>> b = tf.placeholder(tf.float32)
>>> c = tf.constant(6.5)
>>> adder_node = a + b + c
>>> sess = tf.Session()
>>> print(sess.run(adder_node, {a: 3.4, b: 4.5}))
14.4
>>>
```

Figure 1 A basic program showing how nodes and created and executed.

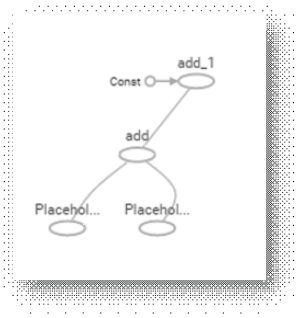


Figure 2 Graph constructed by running example 1.

promise to provide a value later. A constant takes no inputs, and it outputs a value it stores internally.

Figure 1 shows how to create two placeholders, *a* and *b*, and *a* constant *c* (also tf.float32 implicitly). The addition operation is done in another node, *adder_node*.

Notice that to actually evaluate the nodes, we must run the computational graph within a session. A session encapsulates the control and state of the TensorFlow runtime.

The following code creates a Session object and then invokes its run method to run enough of the computational graph to evaluate *adder_node* by running the computational graph in a session, and printing the final value as follows:

```
>>> sess = tf.Session()
>>> print(sess.run(adder_node, {a: 3.4, b: 4.5}))
```

The entire computational graph will look like this Figure 2.

Variables: In machine learning we will typically want a model that can take arbitrary inputs, such as the one above. To make the model trainable, we need to be able to modify the graph to get new outputs with the same input. Variables allow us to add trainable parameters to a graph. They are constructed with a type and initial value.

Example 2: Creating a basic model that finds out the loss between actual data and a training model.

```
W = tf.Variable([.3], tf.float32)
b = tf.Variable([-.3], tf.float32)
```

```
# Import the data set of images and labels
from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)

import tensorflow as tf

# -- Create the graph by defining nodes and tensors

# x is a placeholder conatining images of 28x28 = 784 pixels flattened
x = tf.placeholder(tf.float32, [None, 784])
# Weight matrix W to multiply the 784-dimensional image vectors by it to produce
# 10-dimensional vectors of evidence for the different classes.
W = tf.Variable(tf.zeros([784, 10]))
# b has a shape of [10] so we can add it to the output.
b = tf.Variable(tf.zeros([10]))
# y = softmax(xW + b)
y = tf.nn.softmax(tf.matmul(x, W) + b)

# -- Define the training model using cross-entropy function and
#    Gradient Descent optimizer
y_ = tf.placeholder(tf.float32, [None, 10])
cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))
train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)

# -- Initialize the variables created
init = tf.global_variables_initializer()
sess = tf.Session()
sess.run(init)

# -- Training the model by running the steps 1000 times
for i in range(1000):
  batch_xs, batch_ys = mnist.train.next_batch(100)
  sess.run(train_step, feed_dict={x: batch_xs, y_: batch_ys})

# -- Evaluate the model
correct_prediction = tf.equal(tf.argmax(y,1), tf.argmax(y_,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
print(sess.run(accuracy, feed_dict={x: mnist.test.images, y_: mnist.test.labels}))
```

Figure 3 A training and evaluation model for basic MNIST classification.

```
x = tf.placeholder(tf.float32)
linear_model = W * x + b
```

Constants are initialized when we call tf.constant, and their value can never change. By contrast, variables are not initialized when we call tf.Variable. To initialize all the variables in a TensorFlow program, we must explicitly call a special operation as follows:

```
init = tf.global_variables_initializer()
sess.run(init)
```

Since x is a placeholder, we can evaluate linear_model for several values of x simultaneously as follows:

```
print(sess.run(linear_model, {x:[1,2,3,4]}))
```

Output:

```
[ 0.          0.30000001  0.60000002  0.90000004]
```

We've created a model, but we don't know how good it is yet. To evaluate the model on training data, we need a y placeholder to provide the desired values, and we need to write a loss function.

A loss function measures how far apart the current model is from the provided data. We'll use a standard loss model for linear regression, which sums the squares of the deltas between the current model and the provided data. linear_model - y creates a vector where each element is the corresponding example's error delta. We call tf.square to square that error. Then, we sum all the squared errors to create a single scalar that abstracts the error of all examples using tf.reduce_sum:

```
y = tf.placeholder(tf.float32)
squared_deltas = tf.square(linear_model - y)
loss = tf.reduce_sum(squared_deltas)
print(sess.run(loss, {x:[1,2,3,4], y:[0,-1,-2,-3]}))
```

Output:

```
23.66
```

We can go ahead and train the model we just created to get the correct values of parameters W and b using a gradient descent optimizer (the complete tutorial is available on the TensorFlow website).

```
>>> with open("mnist.py") as f:
    c = compile(f.read(), "mnist.py", 'exec')
    exec(c)

Extracting MNIST_data/train-images-idx3-ubyte.gz
Extracting MNIST_data/train-labels-idx1-ubyte.gz
Extracting MNIST_data/t10k-images-idx3-ubyte.gz
Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
0.9197
>>>
```

Figure 4 Output depicting the model's accuracy with the test data.

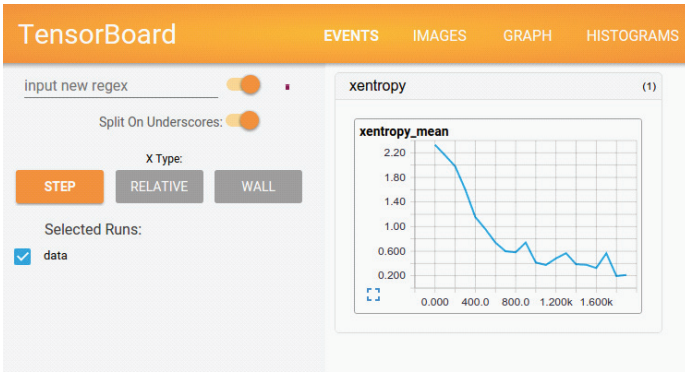


Figure 5 MNIST TensorBoard. (source : https://www.tensorflow.org/images/mnist_tensorboard.png)

Example 3: A hands-on on the basic MNIST tutorial of TensorFlow. This "Hello World!" problem of machine learning world is actually an image classifier problem that gets a handwritten image and finds out the digit from it. Here's the complete code showing how nodes, sessions, tensors, and variables can be used to create an algorithm that is able to train 55000 sets of data and test our model in just a few lines of code!

A brief overview of the computation flow

To compute anything, a graph must be launched in a Session. A Session places the graph ops onto Devices, such as CPUs or GPUs, and provides methods to execute them. These methods return tensors produced by ops as NumPy* ndarray objects in Python, and as tensorflow::Tensor instances in C and C++. The variables are used to maintain state across executions of the graph and are initialized by init() call. Placeholders are used to "feed" the model with data and finally a session is run giving us an output.

That's not all: TensorFlow also provides a visualization of the graphs we constructed using TensorBoard. The computations that we do in TensorFlow - like training a massive deep neural network - can be complex and confusing. To make it easier to understand, debug, and optimize, TensorBoard provides a suite of visualization tools which can plot quantitative metrics about the execution of the graph, and show additional data like images that pass through it.

Conclusion

TensorFlow is an easy to install framework for experimenting with our training models and datasets without having to worry about boilerplates for creating basic setups to work on. It provides the tools and libraries we need so the primary focus is truly development. Optimization algorithms like Gradient Descent are included in the library for us to import and use. Also, the programs can be easily deployed into production systems via TensorFLow Serving, which is a flexible, high-performance serving system for machine learning models, designed for production environments. It makes it easy to deploy new algorithms and experiments, while keeping the same server architecture and APIs. And lastly, although it is primarily used for conducting machine learning and deep neural networks research, the system is general enough to be applicable in a wide variety of other domains as well.

TensorFlow can run on CPUs, GPUs, 64-bit Linux or Mac OS X desktop or server systems, as well as on mobile computing platforms, including Android and Apple's iOS.

**NumPy* is an extension to the Python programming language with the core functionality being "ndarray" (an n-dimensional array or data structure). These are efficient multi-dimensional container of generic data. Numpy has ndarray support, but doesn't offer methods to create tensor functions and automatically compute derivatives (+ no GPU support) like TensorFlow. But it acts as a good data structure for passing between computations in a TensorFlow graph.

bits & bytes

The answer to the question of what makes deep learning different from traditional machine learning may have a lot to do with how much data you're working with.

"When you start getting into true big data, that's when you can really get into deep learning," said Alfred Essa, vice president of research and data science at New York-based publishing company McGraw-Hill Education.

Driven by advances in analytics technologies, deep learning processes became a more widely discussed topic last year. Since then, what constitutes deep learning vs. machine learning has been up for debate. They involve a lot of the same tools and techniques, after all.

But despite some similarities, the two are unique disciplines, Essa said in a presentation at the Business Analytics Innovation Summit in Chicago this week. For example, he pointed out that conventional machine learning algorithms often plateau on analytics performance after processing a certain amount of data. The reason, he said, is that when an algorithm is directed to look for correlations among specific variables, those correlations become apparent fairly quickly. There's only so much it can learn.

The performance of deep learning algorithms, on the other hand, tends to improve exponentially when they're given more training data to analyze, according to Essa. This is partially because they're less directed than machine learning algorithms. They take a neural-network approach to look for patterns and correlations that can be more subtle than what machine learning turns up, and that become clearer only with the use of more data.

The basics of face recognition and its applications to security

Written by **Gulshan Vaswani**, Sr Member Technical Staff 2, Privilege Management

Facial recognition systems extract landmarks and unique features from a face, and try to match them to a facial database

Face recognition has gained much attention in recent years and has become one of the most successful applications of image analysis and understanding. Its importance is increasing in areas like social security to identify troublemaker from a crowd in public places. Face recognition is the process of *detecting* faces in images and videos and *identifying* person in the image, whose face it belongs to. The aim of *face detection* algorithms is to determine the size and location of human faces in digital images, ignoring anything else, like trees or other objects in the image. This can be achieved either by using a reference image containing known faces, or by finding key features of a face, like the eyes, the nose or the mouth. In face recognition, the faces are matched bitwise, meaning that any slight alteration in the processed image (like a different facial expression) can cause the matching to fail. After we found the face in the image, we can use facial recognition to determine who that face belongs to. *Facial recognition* systems extract landmarks and unique features from a face, and try to match them to a facial database.

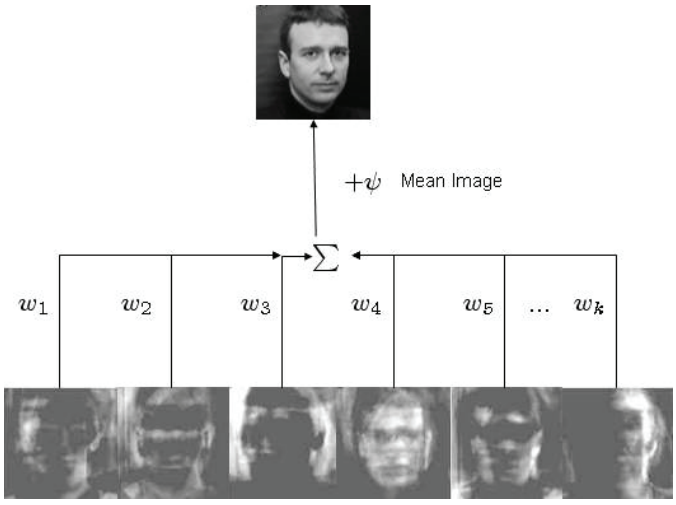
Face recognition process

The process of recognizing a face in an image has two phases:

- Face detection – detecting the pixels in the image which represent the face. There are several algorithms for performing this task, one of these widely used method is “Haar Cascade face detection”.
- Face recognition – the actual task of recognizing the face by analyzing the part of the image identified during the face detection phase.

Face recognition system is expected to identify faces present in images and videos automatically. It can operate operates in either or both of two modes:

- Face verification (or authentication): involves a one-to-one match that compares a query face image against a template face image whose identity is being claimed.
- Face identification (or recognition): involves one-to-many matches that compares a query face image against all the template images in the database to determine the identity of the query face.



Methods and algorithms

Appearance There are three major categories of face recognition methods.

- **Statistcal methods:** Face recognition works by comparing a given image with a stored template and measuring the statistical distance between them. This method is called appearance based statistical methods method. These are methods which use statistics to define different ways of how to measure the distance between two images. In other words they try to find a way to say how similar two faces are to each other. There are several methods which fall into this group. The most significant are:
  - Principal Component Analysis (PCA) – described in this article.
  - Linear Discriminant Analysis.
  - Independent Component Analysis.
- **Gabor Filters:** Filters simplify facial features for easier comparision. Gabor filters are commonly used in image processing that have a capability to capture important visual features. These filters are able to locate the important features in the image such eyes, nose or mouth. This method can be combined with the previously mentioned analytical methods to obtain better results.
- **Neural Networks:** They simulate the behavior of human brain to perform machine learning tasks such as classification or prediction.

Eigenfaces algorithm and PCA for face recognition

Eigenfaces algorithm and PCA for face recognition. Eigenfaces is probably one of the simplest face recognition methods and also rather old. While it is simple it works quite well. The eigenfaces algorithm follows the pattern

1. Compute the distance between the captured image and each of the images in the database.
2. Select the example from the database, which is closest to the processed image (the one with the smallest distance to captured image).
3. If the distance is not too big – label the image as concrete person.

In the above figure, a face that was in the training database was reconstructed by taking a weighted sum of all the basis faces and then adding to them the mean face.

Where, mean face is the average of all trained images in the database.

It transforms the face images into a set of basis faces, which essentially are the principal components of the face images.

When we find the principal components or the Eigenvectors of the image set, each Eigenvector has some contribution from each face used in the training set. So the Eigenvectors also have a face like appearance. These look ghost like and are ghost images or Eigenfaces. Every image in the training set can be represented as a weighted linear combination of these basis faces.

For example,

```
Image1 = Mean Image + 10% Eigenface 1 + 4% Eigenface
2 + … + 1% Eigenface 5
```

For recognizing an input face, the face is projected onto the Eigenfaces and its distance using methods like SVM, Euclidean Distance etc. is calculated. The image is said to be recognized when the distance from a particular image is minimum and within threshold limit.

Applications of Face recognition application to security

Prevent banking fraud: The technology could be used as a security measure at ATMs. In addition to using a bank card or PIN, the ATM would capture an image of the customer's face, and compare it to the account holder's photo in the bank database to confirm the customer's identity. If the face is not detected / matched with the account holder's face, then additional step up authentication can be requested to the user. If that also fails, it's a good chance that the user is fraud and his image can be shared with the banking security or city security organization like police.

Early detection of crime: The system can be installed in public places like airports. Here, the face recognition can be used to detect the faces of pre-known criminals and detain them for further enquiry.

Ease authentication in enterprise software: Mandating secondary authentication in enterprise software is good but tedious for the users. Compared to biometric based authentication, face recognition gives benefits where application users are not mandated to enter secondary authentications like OTP or fingerprint. Rather they are automatically secondary authenticated with the help of cameras.

Future

New approaches to improve the accuracy of face recognition are being developed. One of such areas is 3D face recognition. In this three dimensional images are captured using multiple cameras. In this approach features such as depth of nose, eyes, lips and chin are also considered for face detection.

Libraries

Few libraries that can be used for face detection are openCV, openFace, tensorFlow etc.

References

1. http://blog.octo.com/en/basics-face-recognition/
2. https://onionesquereality.wordpress.com/2009/02/11/face-recognition-using-eigenfaces-and-distance-classifiers-a-tutorial/
3. https://en.wikipedia.org/wiki/Facial_recognition_system
4. http://pages.cpsc.ucalgary.ca/~marina/601/Week6_FaceDetection.ppt
5. http://www.vcl.fer.hr/papers_pdf/Appearance-based%20Statistical%20Methods%20for%20Face%20Recognition.pdf
6. http://uni-obuda.hu/users/kutor/MOI%20K+F%20II.%202014%20tavasz/Hallgat%C3%B3i%20prezent%C3%A1ci%C3%B3k/MOI%20K+F%20II.%202014.cikkek%20prezent%C3%A1ci%C3%B3k/Matusinka%20Roland/Matusinka%20Roland%20-%20Basics%20of%20face%20detection%20and%20facial%20recognition.docx

The Game Should Change

Written by **Vamsi Krishna**, Distinguished Engineer, SMG

The Game should change. We should become predictable at quality and eliminate the "guess work" in our testing models

I have been personally associated with a lot of releases and if I look back we have come a long way in terms of adopting changes in our development practices. More notably, we are now predictable at our release dates. However, we are not so at our quality.

Right now, we as an organization are embarking an endeavor to host our software in Cloud. This is a significant milestone and changes a lot things. Hosting means we are responsible for SLAs, downtime , health, reliability, security and many more attributes of the software we host. All of this means, we need to embrace a good DevSecOps model to minimize the impact for businesses that run on our Cloud. Given that context, I want to share some thoughts on what should change in 2017 in our practices. (*Disclaimer*: In doing so, I may be stereotyping the processes we follow and that is not intentional but to address and connect with majority).

- 2012
  - Some random Sprint starting day
  - First week daily standup
  - Second week daily standup
  - Final week daily standup
  - Sprint Demo
  - Retrospective
  - The Game should change
- 2017
  - Some random day before Sprint starting
  - Sprint Planning
  - First week
  - Second week
  - Final week
  - Sprint Demo
- Conclusion

First lets have a look at what used to happen in 2012.

2012

Some random Sprint starting day

We are an agile organization and we have already embraced Scrum. Each Scrum team consists of members of Dev and QA.

- *Product Owner* wants to build a new "Feature" and would like to make sure it is part of the next release. The story is aptly prioritized so that the team can estimate.
- The Self Organized Team looks at the story and has a meeting with Product Owner to discuss the requirement. Questions are asked and some white board flows happen. The Product Owner draws what he wants on the board. Team feels confident they can build it and play *Planning Poker*. The estimate is X story points and it will take Y sprints to build it.
- Development Architect of the Team works on the design and presents it to all team members. QA architect requests many clarifications and consensus is reached on what will work and what won't.
- The entire team feels they are ready to build.
- The Development team estimates it will take 2 weeks for them to "build and unit test" the software. They can give a drop after 2 weeks for it to be "tested".
- The QA team has a backlog of previous sprint work to complete and they will finish that part before moving on to the new "Feature"
- During the second week, it is decided that the QA team will complete the Test Strategy for certifying the "Feature"
- The Plan is in place

First week daily standup

- The Self Organized Team meets daily at 10 AM for their standup. Six of them attend. 4 Dev and 2 QA.
- Each developer explains their problems and the impediments they are encountering in building the feature. The Dev Architect parks them for a detailed discussion at the end of the standup and is confident of solving them.
- Since the QA engineers are busy with their previous sprint work, are noting the keywords in the standup that can assist them in ensuring the test strategy covers these impediments. Senior QA asks more questions to ensure things are not going out of control.
- The meeting ends and moves to Parking Lot. QAs are allowed to leave as they have "Important and Urgent" previous sprint work to be completed and closed.

- Dev Architect cracks all the problems in the next 15 minutes and gives directions to the dev engineers on what to accomplish before next standup.
- The continues for the first week.

Second week daily standup

- The QA team is up to speed this week. They have completed every task of the previous sprint. Now, they are fully "focused".
- Standups become more interactive with Dev showing some initial shape of the "Feature". The QA are happy as the "Feature" somewhat looks similar to what they have imagined it to be.
- Where required, more questions are asked about unhandled negative cases and it is decided to have a detailed discussion after the standup to handle negative cases.
- During Parking Lot, QA feels the feature developed lacks "Must Haves" for it to be picked up for testing.
- Some negotiations later, the Self Organized Team decides to balance at a mid path and a revised plan to ensure the "success" of sprint is achieved.

Final week daily standup

- The initial drops given by the Dev to QA are not in acceptable quality. QAs indicate many basic things do not work, but Dev argue the basic things were never part of the requirements. More discussions happen in Parking Lot to come up with a fresh plan to make things "happen"
- The plan is adjusted such a way that some parallelism can be achieved by Dev and QA on a daily basis
- As the Sprint end approaches, the priority shifts to "Absolute Musts" for testing and postponing "Exploratory" , "Reliability" and "Non-functional" tests to next sprint
- Basic things start working and we have one more day to go
- More and more things are getting fixed by last day and "Validation" by QA is the only pending activity of the Sprint. Dev is already celebrating and preparing for Sprint End party
- QA pulls off by working hard and the setup is ready for demo

Sprint Demo

- Product Owner approves most part of the Demo and asks for few changes in the workflow based on feedback in Demo. The team records all the feedback and pushes things that they cannot fix in the reminder half day of the Sprint.
- A negotiation follows and a new story is added to backlog to "capture" the enhancements suggested

Retrospective

- Some enthusiasts suggest we have done a great job, but we need to do a better "plan" from next sprint. It is decided that everybody sticks to the "plan" and callout of impediments should happen more clearly and loudly
- Team is unhappy with the amount of test case automation and it is decided to "try" how that can be changed
- QA engineers raise concerns they are always in the catch up mode due to previous sprint activities and since they are not involved "early" enough they are not able to "focus"
- The meeting is "time boxed" and the next Sprint is about to start

The Game should change

The above story is completely hypothetical and a work of fiction. But, if you have related to this story and can see that your team follows some of this, its a matter of concern. For, there are multiple smells and flaws in this mode of adoption of Scrum and Agile practices. According to me, this is not Scrum. This is Waterfall in the name of Scrum. The reasons are clear and here are some of the smells

- The Design is only owned by the Devs
- There is a backlog of work for QAs at the beginning of every sprint
- First week is not producing anything Testable
- There is no such thing as Dev drop to Tests. Testing has to be continuous and drive the Development. Refer T*est Driven Development*. "Prevention better than Cure" not practiced.
- Exploratory , Reliability and Non-functional testing deemed unimportant for Demo
- The testing is largely fear-based and *Black Box*. No visible movement towards *White Box* Testing.
- Importance is given to "basic things" that fail and fixing them creates a false perception of "completeness"
- Automation Testing is not adequate
- *Test Pyramid* is not even discussed

The Game should change. We should become predictable at quality and eliminate the "guess work" in our testing models. The answer lies in developing code that is "White Box Testing" enabled. Speaking/blogging this is very easy but very difficult to practice. But having tried this personally, it is worth every penny and every second you spend on it. The more you practice you will feel you are on the right path. The world around is moving, so we are not alone. We ,as a company, make lot of money selling Testing Tools and shouldn't use them in-house?

Here are my proposals for the change

- Dev is not from Mars and QA not from Venus. We are all part of one team that is aligned to self organize. Every Dev has an inherent ability to build things and every QA has an ability to visualize how they break. Both of them doing them together ,at the same time, makes the structure rigid and solid. I call this Dev-QA Pairing. We should attempt this.
- Dev and QA together MUST own the design. This is not optional. Catching bugs early and preventing them in the first place is the new normal.
- There should be strict adherence of Test Pyramid. If you have a bug found due to end to end testing, it basically means you have a missing unit test. More about it below.
- Testing is continuous development activity and there is no such thing as "test drop" or a "dev drop".

Lets welcome 2017.

2017

Some random day before Sprint starting

- Product Owner wants to build a new "Feature" and would like to make sure it is part of the next release. The story is aptly prioritized so that the team can estimate.
- The Self Organized Team looks at the story and has a meeting with Product Owner to discuss the requirement. Questions are asked and some white board flows happen. The Product Owner draws what he wants on the board. Team feels confident they can build it and play Planning Poker. The estimate is X story points and it will take Y sprints to build it.
- Development and QA Architects of the Team work on the design

Sprint Planning

- A high level presentation of the design happens by the Dev and QA Architects. Questions are asked and clarified. The team members get an understanding of high level design and get

ready to plan the Sprint based on the historic velocity of the team

- There are no milestones in the plan. We are done when all the Test cases are passing which are yet to be identified.
- The last couple of days are reserved for Exploratory Testing of the feature

First week

- The whole team meets for a Test Strategy Discussion. The High Level Design is drawn on the board and test strategy is arrived it. Now, all the team members go through an exercise of rationalizing the Test Strategy. During this step, every "end to end" test is questioned if it can be covered at much lower levels of testing i.e integration/component/unit tests. Many permutations and combinations of test inputs are easily achieved at unit test level. Hence they are "demoted" to lower levels in the Test Pyramid.
- All the failing tests are added into the system so that the "Continuous Integration" build system picks up the build and marks 100% failures
- Dev and QA team members pair up to build components simultaneously. It is preferable to try out formal Pair Programming techniques
- As teams meet up for standup, the progress is measured only by the amount of Test cases passing day over day

Second week

- About 50% of test cases are passing. QA members are also able to build end to end automation through simulated tools.
- In the standup, many impediments are discussed and the Self Organized Team has to adjust some code to suit the new requirements that popped up. The QA analyzes the impact and suggest changes to the test strategy required.
- Newer test cases are added and the team makes sure existing ones are passing.
- The progress dips, but the team knows it can be brought back to normal. The team knows , with nothing to be repeated manually, we are on track.

Final week

- The QA team demos the finishing feature to Product Owner to get inputs using their Automation Suite. More changes are suggested. The team adds failing test cases and the progress dips again.
- The team meets to discuss the "new changes" and estimates what can be absorbed in the current sprint now that they have most of the feature, passing already accepted tests.
- The complex changes are recorded to be picked up as a future story
- Exploratory testing is done for the last two days and the bugs are attended as they arrive, of course by writing failing test cases first.

Sprint Demo

- The Demo is given and 100% passing test cases indicate completeness. No perception, but pure data.

Conclusion

Of course, my 2017 is a depiction of perfection and everything-falling-in-place and I fully know it is difficult to meet the picture as stated. However, this is not impossible. All around the world, teams are beginning to get more and more closer to this depiction. We should try and not stay behind. We should become predictable at quality.

Happy New Year 2017!!

bits & bytes

In its continued efforts to make Azure a platform that appeals to the widest range of developers possible, Microsoft announced a range of new features at Build, its annual developer conference.

Many of the features shown today had a data theme to them. The most novel feature was the release of Cosmos DB, a replacement for, or upgrade to, Microsoft's Document DB NoSQL database. Cosmos DB is designed for "planet-scale" applications, giving developers fine control over the replication policies and reliability. Replicated, distributed systems offer trade-offs between latency and consistency; systems with strong consistency wait until data is fully replicated before a write is deemed to be complete, which offers consistency at the expense of latency. Systems with eventual consistency mark operations as complete before data is fully replicated, promising only that the full replication will occur eventually. This improves latency but risks delivering stale data to applications.

Document DB offered four different options for the replication behavior; Cosmos DB ups that to five. The database scales to span multiple regions, with Microsoft offering service level agreements (SLAs) for uptime, performance, latency, and consistency. There are financial penalties if Microsoft misses the SLA requirements. The company describes Cosmos DB as "schema agnostic," performing automatic indexing of data regardless of how it's structured and scaling to hundreds of trillions of transactions per day. Cosmos DB is already being used by customers such as online retailer Jet.com.

Many applications still call for traditional relational databases. For those, Microsoft is adding both a MySQL and a PostgreSQL service; these provide the familiar open source databases in a platform-as-a-service style, removing the administrative overhead that comes of using them and making it easier to move workloads using them into Azure.