**Project Doc**

**Words Dictionary:**

**Mellanox overview**

Mellanox Technologies is a leading supplier of end-to-end InfiniBand and

Ethernet interconnect solutions and services for servers and storage.

Mellanox offers a choice of fast interconnect products: adapters, switches,

software and silicon that accelerate application runtime and maximize business

results for a wide range of markets including high performance computing,

enterprise data centers, Web 2.0, cloud, storage and financial services.

**Mellanox Call Center**

Mellanox Call Center provides permanent manned support services for Mellanox customers throughout the year (7/24/365) based on the appropriate support contract.

The Call Center is the customer’s point of contact (mostly for emergency cases) for initiating and advancing technical related and support administrative issues.

Call Center activities are designed to accomplish the following targets:

1. Provide systematic and efficient call handling services to Mellanox the customer.

2. Maintain information flow according to developing situations .

3. React to cases according to predefined procedures.

Technical support

**Mellanox's online Customer :**

Mellanox's online Customer Resource Management (CRM) system – SalesForce,

provides a comprehensive online tool to manage all of the customer's support

issues in one place.

1. Complete case management including reporting support issues and tracking their progress.

2. A searchable knowledge database to find solutions, best practices and worthy information

3. Access to documentation and drivers/firmware/software downloads

4. Built-in RMA request and tracking system.

**Mellanox Care – Proactive Care Platform:**

24/7 fabric management services provided by Mellanox networking experts

Mellanox Care services use a combination of advanced monitoring software with a 24/7 human expert service of Mellanox personnel

Mellanox Care will identify, alert and address hardware failures, non-optimal configuration, service degradation issues and more. Above and beyond keeping the fabric healthy, Mellanox Care identifies and addresses the more complex performance issues and bottleneck scenarios, which are impacting application performance.

**InfiniBand (abbreviated IB):**

A computer-networking communications standard used in [high-performance computing](https://en.wikipedia.org/wiki/High-performance_computing), features very high [throughput](https://en.wikipedia.org/wiki/Throughput) and very low [latency](https://en.wikipedia.org/wiki/Latency_(engineering)). It is used for data interconnect both among and within computers. InfiniBand is also utilized as either a direct, or switched interconnect between servers and storage systems, as well as an interconnect between storage systems.

**Ethernet:**

Ethernet is the most widely installed local area network (LAN) technology. Ethernet is a link layer protocol in the TCP/IP stack, describing how networked devices can format data for transmission to other network devices on the same network segment, and how to put that data out on the network connection. It touches both Layer 1 (the physical layer) and Layer 2 (the data link layer) on the OSI network protocol model. Ethernet defines two units of transmission, packet and frame.

**Open Ethernet:**

The Open Ethernet initiative is based on a complete separation between the switch hardware and the switch software. In simple terms, Open Ethernet allows any software to be run on any hardware and allows either the hardware or the software to be changed without changing the other components.

**RDMA over Converged Ethernet (RoCE):**

Remote Direct Memory Management (RDMA) is the remote memory management capability that allows server to server data movement directly between application memory without any CPU involvement. RDMA over Converged Ethernet (RoCE) is a mechanism to provide this efficient data transfer with very low latencies on lossless Ethernet networks. With advances in data center convergence over reliable Ethernet, Mellanox’s NICs with RoCE uses the proven and efficient RDMA transport to provide the platform for deploying RDMA technology in mainstream data center application at 10GbE, 40GbE and 56GbE link-speed. The Mellanox NICs with its hardware offload support takes advantage of this efficient RDMA transport (InfiniBand) services over Ethernet to deliver ultra-low latency for performance-critical and transaction intensive applications such as financial, data base, storage, and content delivery networks.

**Python vs Java:**

Java unit testing Junit, TestNG and webdriver.

pyhton's unit testing frameworks like PyUnit, nose

Speed:

Python is an interpreted language hence it's execution is slower.

Usability:

Python is easier to program in than Java.

Java programs are longer than their equivalent python program

Resourceful:

Java's documentation is generally better older and is more developed.

Python programs can run from sources and Java must be compiled first.

**JAVA**

statically typed

In Java, all variable names (along with their types) must be explicitly declared. Attempting to assign an object of the wrong type to a variable name triggers a type exception.That’s what it means to say that Java is a statically typed language.

Java container objects (e.g. Vector and ArrayList) hold objects of the generic type Object, but cannot hold primitives such as int. To store an int in a Vector, you must first convert the int to an Integer. When you retrieve an object from a container, it doesn’t remember its type, and must be explicitly cast to the desired type.

**PYTHON**

dynamically typed

In Python, you never declare anything. An assignment statement binds a name to an object, and the object can be of any type. If a name is assigned to an object of one type, it may later be assigned to an object of a different type. That’s what it means to say that Python is a dynamically typed language.

Python container objects (e.g. lists and dictionaries) can hold objects of any type, including numbers and lists. When you retrieve an object from a container, it remembers its type, so no casting is required.

**exception:**

In your application, method A calls B calls C calls D calls E calls F. You discover that F must throw exception SpecialException, and it must be caught by A.

|  |  |
| --- | --- |
| JAVA | PYTHON |
| You must throw SpecialException in F, and catch it in A. **and** You must add “throws SpecialException” to the signatures of methods B, C, D, E, and F. | You must raise SpecialException in F, and catch it in A.Exceptions will propagate upward automatically; there is nothing more that you must do. |

**Test softwares:**

A software test is a piece of software, which executes another pierce of software and validates if that code results in the expected state (state testing) or executes the expected sequence of events (behavior testing).

http://www.vogella.com/tutorials/JUnit/article.html#whataresoftwaretests

**Unittest :**

The unittest unit testing framework was originally inspired by JUnit and has a similar flavor as major unit testing frameworks in other languages. It supports test automation, sharing of setup and shutdown code for tests, aggregation of tests into collections, and independence of the tests from the reporting framework.

test fixture

A test fixture represents the preparation needed to perform one or more tests, and any associate cleanup actions. This may involve, for example, creating temporary or proxy databases, directories, or starting a server process.

test case

A test case is the individual unit of testing. It checks for a specific response to a particular set of inputs. unittest provides a base class, TestCase, which may be used to create new test cases.

test suite

A test suite is a collection of test cases, test suites, or both. It is used to aggregate tests that should be executed together.

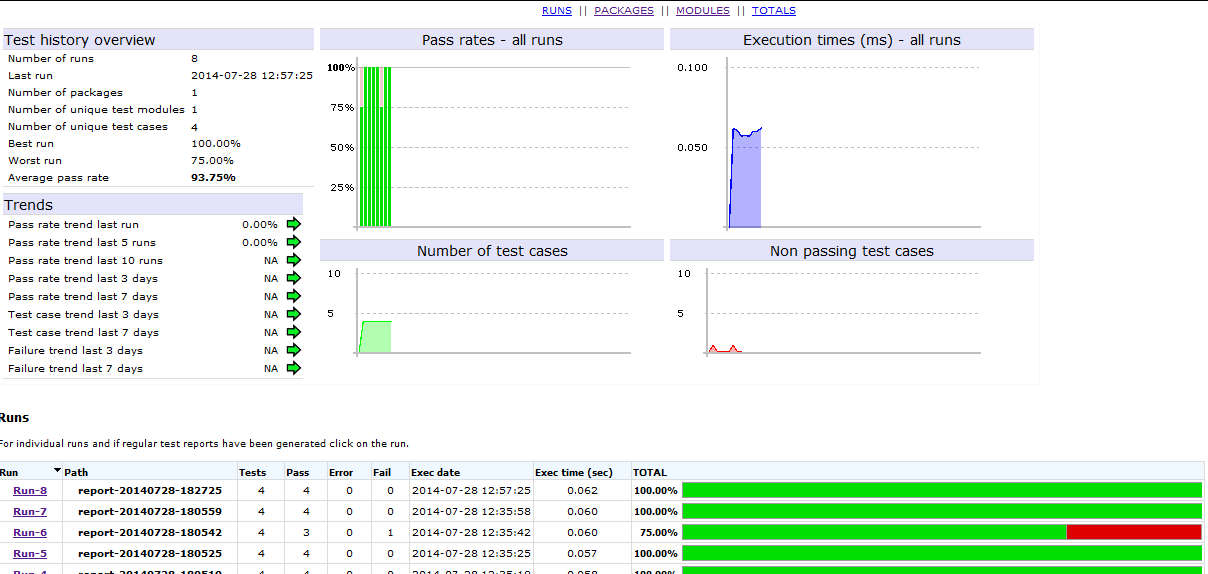
test runner

A test runner is a component which orchestrates the execution of tests and provides the outcome to the user. The runner may use a graphical interface, a textual interface, or return a special value to indicate the results of executing the tests.

<https://docs.python.org/3.6/library/unittest.html>

**Junit:**

JUnit is a simple framework to write repeatable tests. It is an instance of the xUnit architecture for unit testing frameworks.



**Knowlage Center:**

Machine learning is a subfield of computer science[1] that evolved from the study of pattern recognition and computational learning theory in artificial intelligence.[1] Machine learning explores the study and construction of algorithms that can learn from and make predictions on data.[2] Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions,[3]:2 rather than following strictly static program instructions.

**Machine learning**

Machine learning is closely related to computational statistics; a discipline that aims at the design of algorithm for implementing statistical methods on computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms is infeasible. Example applications include spam filtering, optical character recognition (OCR), search engines and computer vision. Machine learning is sometimes conflated with data mining, although that focuses more on exploratory data analysis. Machine learning and pattern recognition "can be viewed as two facets of the same field.

**Artificial intelligence**

Artificial intelligence (AI) is the intelligence exhibited by machines or software. It is also the name of the academic field of study which studies how to create computers and computer software that are capable of intelligent behavior. Major AI researchers and textbooks define this field as "the study and design of intelligent agents", in which an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success. John McCarthy, who coined the term in 1955, defines it as "the science and engineering of making intelligent machines"

AI research is highly technical and specialized, and is deeply divided into subfields that often fail to communicate with each other. Some of the division is due to social and cultural factors: subfields have grown up around particular institutions and the work of individual researchers. AI research is also divided by several technical issues. Some subfields focus on the solution of specific problems. Others focus on one of several possible approaches or on the use of a particular tool or towards the accomplishment of particular applications.

The central problems (or goals) of AI research include reasoning, knowledge, planning, learning, natural language processing (communication), perception and the ability to move and manipulate objects. General intelligence is still among the field's long-term goals. Currently popular approaches include statistical methods, computational intelligence and traditional symbolic AI. There are a large number of tools used in AI, including versions of search and mathematical optimization, logic, methods based on probability and economics, and many others. The AI field is interdisciplinary, in which a number of sciences and professions converge, including computer science, mathematics, psychology, linguistics, philosophy and neuroscience, as well as other specialized fields such as artificial psychology.

**TensorFlow:**

Built by the Google Brain team, TensorFlow represents computations as stateful dataflow graphs. TensorFlow is able to model computations on a wide variety of hardware, from consumer devices such as those powered by Android, to large-scale heterogeneous, multiple GPU systems. TensorFlow claims to be able to, without significant alteration of code, move execution of the computationally expensive tasks of a given graph from solely CPU to heterogeneous GPU-accelerated environments. Given these details, it goes without saying that TensorFlow aims to bring massive parallelism and high scalability to machine learning for all.   
  
At the heart of TensorFlow is the dataflow graph representing computations. Nodes represent operations (ops), and the edges represent tensors (multi-dimensional arrays, the backbone of TensorFlow). The entire dataflow graph is a complete description of computations, which occur within a session, and are executed on devices (CPUs or GPUs). Like much contemporary scientific computing and large-scale machine learning, TensorFlow favors its well-documented Python API, where tensors are represented internally as familiar **numpy** ndarray objects. TensorFlow relies on highly-optimized C++ for its computation, and also supports native APIs in C and C++.

**Prolog:**

Prolog is a general purpose [logic programming](https://en.wikipedia.org/wiki/Logic_programming) language associated with [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence) and [computational linguistics](https://en.wikipedia.org/wiki/Computational_linguistics)

Prolog has its roots in [first-order logic](https://en.wikipedia.org/wiki/First-order_logic), a [formal logic](https://en.wikipedia.org/wiki/Formal_logic), and unlike many other [programming languages](https://en.wikipedia.org/wiki/Programming_language), Prolog is [declarative](https://en.wikipedia.org/wiki/Declarative_programming): the program logic is expressed in terms of relations, represented as facts and [rules](https://en.wikipedia.org/wiki/Rules_of_inference). A computation is initiated by running a *query* over these relations

**Heuristic algorithm:**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence), and [mathematical optimization](https://en.wikipedia.org/wiki/Mathematical_optimization), a heuristic is a technique designed for [solving a problem](https://en.wikipedia.org/wiki/Problem_solving) more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution. This is achieved by trading optimality, completeness, [accuracy](https://en.wikipedia.org/wiki/Accuracy_and_precision), or [precision](https://en.wikipedia.org/wiki/Accuracy_and_precision) for speed. In a way, it can be considered a shortcut.

**Similar Apps :**

**LOGalyze :**

<http://www.logalyze.com>

LOGalyze is an open source, centralized **log management** and **network monitoring** software. If you would like to handle all of your log data in one place, LOGalyze is the right choice. It supports Linux/Unix servers, network devices, Windows hosts. It provides **real-time event detection** and **extensive search capabilities**.

**Octopussy - Perl/XML Logs Analyzer, Alerter & Reporter**

Octopussy is a Log analyzer tool. It analyzes the log, generates reports and alerts the admin. It has LDAP support to maintain users list. It exports report by Email, FTP & SCP. Scheduled reports could be generated. RRD tool to generate graphs.

It is capable to parse the logs generated from lots of services such as Bind, Cisco Router, Cisco Switch….

**MLNX-OS Sysdump Analyzer**

The goal of this tool is to analyze swtich dump files and show them in a friendly GUI which eases getting a bright picture of the setup and provide a resolution as early as possible

The main purpose of the tool is to avoid dragging “ping pong” threads

All the info is shown in windows form.

The tool serves all groups (QA, Verification, FAEs, Support, Application Engineers, R&D) to enhance troubleshooting procedure and time reduction