CHAPTER 9

Cloud and SOA

CONTENTS

9.1 INTRODUCTION

Any enterprise-wide transformation has significant challenges for people, processes, and technology. Therefore, identifying challenges ahead of the time and defining a mitigated approach can help in making such a transformation successful. Some of the challenges include resistance to change by people and organizations. The transformation factors include roles and responsibilities, management, skills development and discipline, and cultural shifts. This also includes creating awareness in the organization for the need to drive such a transformation in the best interests of the business. It is very difficult to deal with infrastructure complexity, including hardware, software, and applications, across disparate environments (line of business' stakeholders, partners, and customers). Well-planned assessments are needed to understand where to start from and how to progress in a staged way.

Service management is one of the similarities between the cloud infrastructure and Service-Oriented Architecture (SOA) approaches. Developing an integrated service management approach for both application and infrastructure services will drive efficiency in IT operations by improving resource utilization and service levels. Such an integrated service management can move IT toward an end-to-end service-oriented environment. Such an environment enables business agility by aligning IT with the business in a better way.

9.1.1 *Enterprise Infrastructure and SOA*

Design and provisioning of an enterprise infrastructure must be focused on the needs of enterprise organizations. Through comprehensive capabilities of products, services, and integrated solutions, IT organizations are delivering increasingly complex solutions driven by SOA and related methodologies. To achieve their ambitious growth goals, leading IT organizations require continuing investments in IT and taking the advantage of emerging capabilities. A future technology platform will need to support agile business organizations through the simplification of information systems and reduce the complexity of the IT ecosystem through consolidation and rationalization.

We need a governance model for the heterogeneous environment owned by many parties and providing an end-to-end IT infrastructure. This governance model will define the IT infrastructure service requirements in support of an integrated service offering for business systems.

As the SOA projects are deployed, development of an effective design of the supporting infrastructure becomes critical. SOA introduces the requirements for availability, service continuity, monitoring, scalability, and geographic dispersion that are different than those of the past architectures.

SOA converts IT applications into composite applications. Instead of traditional monolithic applications, composite applications are created, which are composed of many services often developed and deployed independently by separate development teams on different schedules. By adhering to common standards and interfaces, the development of new composite applications and the extension of existing applications are made easier through the reuse of existing services and the rapid integration of new services.

Similar concepts to SOA drive the cloud infrastructure, an approach that makes an IT infrastructure a collection of service components with common standards and interfaces. The cloud infrastructure makes the deployment of new infrastructure using repeatable and existing processes.

Cloud infrastructure service components include the physical infrastructure (such as processors, memory, storage, and I/O networks), system software (firmware and OS), and management software (monitoring, provisioning, and workload management). While the cloud infrastructure is particularly suited to support SOA applications, Service-Oriented Infrastructure (SOI) is well-suited to legacy application support. The service components of cloud infrastructure are independent of application architecture and capable to provide flexible support to any application.

Cloud infrastructure strongly leverages virtualization technologies, which enable rapid deployment and redeployment of service components.

9.2 SOA JOURNEY TO INFRASTRUCTURE

The path to transformation consists of a long journey with a staged approach, leading to the ultimate goal of a service-oriented enterprise. Multiple islands of disparate infrastructures in today's environment need to be consolidated to gain control, reduce costs, and become operationally efficient. The next step is to introduce virtualized infrastructure to improve the utilization levels, allow dynamic flexibility to move resources, and the capacity to meet fluctuating workload demands. It is important to note how service orientation (SO) can be achieved by building capabilities on the top of virtualized and automated infrastructu ?. Service orientation is a state where the infrastructure is provided and utilized as a service rather than in piecemeal. Latest innovations, such as cloud computing, will help to further expand the service-oriented paradigm to meet the scaling demands of future state of businesses.

9.3 SOA AND THE CLOUD

SOA binds both the deliver and leverage cloud-based services. Cloud computing relies on service orientation (virtualization at the application layer) to loosely couple the applications with the underlying infrastructure model for using Web services – such as service requestors, service registry, and service providers. Web services are required to compose the complex distributed legacy applications that can be customized later. This helps to integrate bespoke applications and provide the service for extensibility, encapsulations, and interoperability. SOA works as a foundation for the stepping stones to the cloud journey. It embraces all the characteristics of the cloud – such as self-service, shared and dynamic infrastructure, and virtualization (Fig. 9.1).

Cloud computing is an infrastructure management and services deployment method with virtualized resources and it is managed as a single large resource. Clouds share and leverage the characteristics of SO with flexibility and agility. Applications and services are reused in new and dynamic ways and so rapid deployment happens in SOA-based cloud implementation.

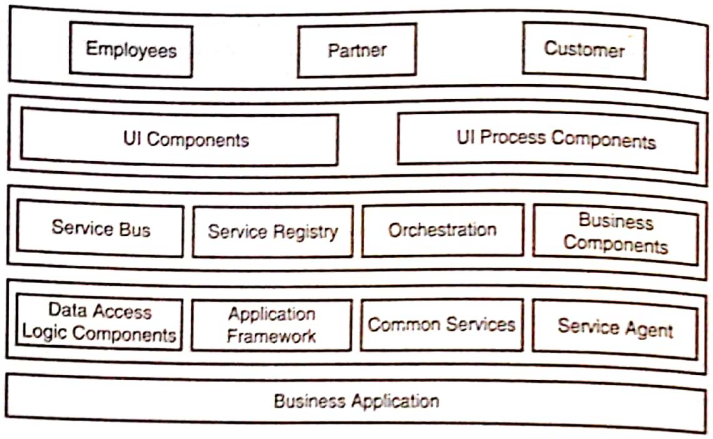

FIGURE 9.1 SOA model.

The SOA infrastructure is required in order to effectively apply SO to an enterprise or large-scale software component development. Basically, the SOA infrastructure consists of middleware, physical infrastructure, and management – collectively covering the non-functional requirements.

SOA is an application framework for building better applications. Web services and SOA do not replace applications as they are today; they complement their functionality and allow for better reuse and business flexibility. SOA helps to break an application framework into discrete service components (i.e., mini-applications) so they can be reused as common services between different applications. An IT infrastructure must continue to evolve and mature to support the new demands in the industry on a distributed and virtualized application framework. SOA applications require the same end-to-end performance, security, and management.

The key to an effective SOA infrastructure is to understand the changes that will take place and which tools are available for infrastructure architecture and design. The best SOA infrastructures have been designed keeping both application and infrastructure perspective in mind.

A well-designed SOA infrastructure is a mix of current processes and SOA infrastructure technologies. SOA and traditional applications do not exist outside of one another. All applications are a part of a shared services environment and use common infrastructure components. Traditional system designs need to be updated to support the new application requirements.

Clouds enable the deployment of cloud services, and SOA is the most sophisticated architectural approach for building and delivery of services. SOA is a design pattern that is composed of loosely coupled, discoverable, reusable, and interoperable platform-neutral services. Each of these services follows a well-defined standard and can be bound or unbound at any time as needed. The value of SOA comes from having an architecture that readily accommodates change.

Clouds are about infrastructure and deployment technology. The concept of service delivery is independent of deployment scale and may operate via a connection between two or three computers only; whereas, cloud computing represents a much larger-scale implementation. SOA is the way an enterprise builds, maintains, governs, and orchestrates the services to be delivered; cloud computing is an instance of SOA. Software-as-a-Service (SaaS) is a term used by a group of companies or individuals when they host a set of software services over the Web. SaaS focuses on the software hosted as a service, and may be considered as a consumption model in which a user is involved. Whereas, SOA focuses on the software designed as a service, and is a design model in which there is no restriction on the consumer.

Private clouds can simply be seen as SOAs with the added characteristics of virtualization and self-provisioning. Those who use private clouds, or virtualization, typically break down new and old applications as services, processes, and data, and address each as an architectural component that may be freely distributed in private clouds.

SOA is important to cloud computing and the use of SOA will promote the adoption of cloud computing. Enabling SOA is important for enabling an infrastructure for cloud service delivery. Most experts agree that without SOA, a move to clouds will be tough to justify financially, because re-engineering legacy systems that are not built to be exposed outside of the usual user community may end up being too costly. Enterprise cloud initiatives require decoupled data systems working together, without the need for personnel and other resources to set up and maintain them. To achieve this objective, integration is the key. The loosely coupled aspect of SOA is very important.

Loosely coupled application, processes, and data give the best option to move to the cloud for migration services. If SOA is not practiced in an organization, the migration to the cloud will be a difficult option as all applications, processes, and data will be tightly coupled. Subsequently, it will really be difficult to change, transform, rewrite, or migrate them.

If a service is initiated by a cloud user, there is a requirement to provide the application programing interface (APIs) for the legacy functionalities in the service model. In order to accomplish this, it calls the integration services to form firewalls for the technology and architecture barriers. This gap is bridged by the service bus and it is the pivotal point for the IT infrastructure and cloud services. SOA helps to identify service portfolios or part of the services, and decides to transfer certain services which are cloud ready to the cloud. If one thinks for the cloud services without SOA proposition, it becomes the cause of failure of the infrastructure.

A 'service communications backbone' is needed to run between the different clouds being used, allowing users to utilize remote services from any cloud without dealing with connectivity and interoperability issues. It is a simple concept, but without it, cloud-to-cloud interoperability issues may limit the growth of cloud computing. This is really going to require state-of-the-science SOA, with the ability to access thousands of services that could be hosted anywhere and to abstract them from interoperability issues. As is always the case with such industry efforts, the standards process takes time. The trick is to develop service architectures that will not require an overhaul in the future based on specifications that are yet to be defined.

9.3.1 *Infrastructure Technologies*

Cloud infrastructure is based on virtualization that is composed of dynamic systems that enable the definition and delivery of resources on demand. Current server technology can deliver hundreds of virtual servers on small clusters of physical servers, enabling flexibility and high availability.

In a virtual environment, workloads can be moved dynamically between components, allowing minimal unplanned downtime and no planned downtime. Each server contains a pool of processors, memory, and I/O resources that can be dynamically assigned and reassigned to meet needs. Surplus capacity cannot be pre-provisioned until activated.

9.4 SOA DEFINED

SOA is defined as a methodology for architecting the solution that looks for flexibility based on loose coupling and encapsulation. The functionalities of SOA are exposed as services, and there exists an instance for each service and its implementation. The deployment of these services requires less deployment time as they are built once and can be redeployed at multiple instances, and can be invoked at remote or onsite locations.

SOA is about an evolving living organism and not about building a house. This is an ongoing journey and not a project that finishes with a concrete result. Agility of the business is an important factor for business continuity as it helps to find faster solutions to changing business priorities and leverages the competitive effectiveness of business change requirements. SOA is defined by what a service is. Services are defined by the following characteristics:

1. Explicit, implementation-independent interfaces.
2. Loosely bound.
3. Invoked through communication protocols.
4. Stress location transparency and interoperability.
5. Summarized reusable business task and assets.

Conceptually, SOA can be visualized by the roles of individuals in any organization. The architect sees SOA from the perspective of the entire business and uses SOA implementation to bridge the gaps of the business.

SOA is very flexible; therefore, it facilitates the different elements of business. The most important characteristic of SOA is the flexibility to treat the following elements of business:

1. Business processes.
2. Underlying the IT infrastructure.
3. Secure standardized components (services).
4. Changing business priorities.

So when we look at the SOA vision, we need to look at the following three aspects:

1. The business view of a service – what is needed to support the business process?
2. The architecture view of a service – how do we define and design these services?
3. The implementation view of a service – how do we implement the service through the component deployed on the technical infrastructure?

In order to run the business in a smooth manner, we should bundle the business requirements in a simplistic and standard way. This creates the service offerings and helps to get the right information from the right source, particularly the information about when the service is needed. This enables us to reuse and combine different other services being offered to answer the requirements of winning against competition.

9.4.1 *SOA Lifecycle*

The SOA lifecycle is not so different from the traditional application lifecycles. It provides a new methodology. The lifecycle starts with the gathering of an organization's requirements, and information for designing the processes and architecture in the SOA model. After the optimization phase is over, SOA is implemented with the help of integrating new and existing services. Once SOA is deployed, the end customers administrate and monitor the services from the business and IT perspective. The information gathered in the steady-state operations is fed back into the lifecycle to improvise the services. Assembling all the stages of the lifecycle gives governance and becomes the torch bearer for the guidance and supervision of the SOA project.

9.4.2 *Service-Oriented Computing*

Service orientation is a design paradigm comprising a specific set of design principles. Its most important feature is its reliance on the 'separation of concerns' design philosophy. Separation of concern is based on the simple fact that a problem becomes easier to approach if it is divided and each part is handled separately.

The first question that should come into the mind is what a service is. Service is not only limited to software or IT; it is actually a culture of the organization and how it performs its operations on a day-to-day basis. We can divide all these tasks into small processes and investigate the processes that can be repeated and can be used as a business continuity process. This also implements the agility factor for the business. Now, if we talk about SO, it is based on the integration of all the business processes as related processes to get the achievable outcomes intended from the business.

The next question that arises is regarding the technology associated with SOA. This visualizes the architectural aspects of service orientation to make the process simple and gives the option of composite application. The composite application ties the running process and business requirements in such a way that it helps to achieve the business goals.

9.5 SOA AND INFRASTRUCTURE AS A SERVICE

Major industry analysts view cloud infrastructure as a key IT ingredient for business agility. With a predicted 60% of the IT spending being applied to infrastructure, analysts recommend an IT infrastructure that is:

1. Shared across customers, departments, and applications.
2. Dynamically driven by business policies and service-level requirements.

The analysts view IT virtualization and IT automation as two major elements in realizing the infrastructure as a service. One of the technological aspects of the cloud infrastructure is IT virtualization that helps to create pools of compute in shared models. This masks the physical topologies, boundaries, and restrictions from the users, and powers the user with unconditional infrastructure. In other words, virtualized resources are viewed as fluid-utility services for the consumers to consume as needed (Fig. 9.2).

At the same time, automation provides the option to govern and optimize the utility and self-service based model to enable infrastructure, policies, dynamic management, and service orientation. This works as a strategic and optimized return on investment (ROI) option rather than looking for a short-term, unpredictable, and quick fix alternatives.

9.5.1 *Architecture*

Cloud infrastructure has many service components. However, they all need not be implemented concurrently. Services can be divided into four domains: application services, information services, common IT services, and infrastructure services. Within each domain, SOA can be measured and charted across a continuum of increasing dynamism and partner involvement.

Application services provide application frameworks to enhance the execution of business services through software engineering. Adopting new technologies and techniques can accelerate the delivery of new services through the use of consistent, repeatable SOAs.

Information services provide a common, repeatable method for cataloging, accessing, and managing information. Innovative technologies can streamline information access and data management, making it easier to integrate packages and new acquisitions. Common IT services create enterprise pools of commonly used IT services. Simplifying the environment can enhance the management and cost, and increase responsiveness.

Infrastructure services provide pools of processing and networking resources for applications and business functions. Today, these resources may be isolated into business silos, but with virtualization, they can evolve into virtual pools that are dynamically allocated based on business needs.

Continuous improvement is mapped within the maturity levels of the company itself and can be measured in each domain of the service. SOA plays a fruitful exercise to decide on how to implement the infrastructure design based on SOA principles to attain the targeted goals. It offers a number of business values.

Business Agility

This helps in defining the right time to launch or rapidly scale the deployment efforts needed to implement the new solutions.

Lower Cost of Operations

1. This helps in utilizing the virtual pools efficiently which decreases the chance of procuring new systems.
2. It helps in increasing the overall effectiveness while working in an automated environment.

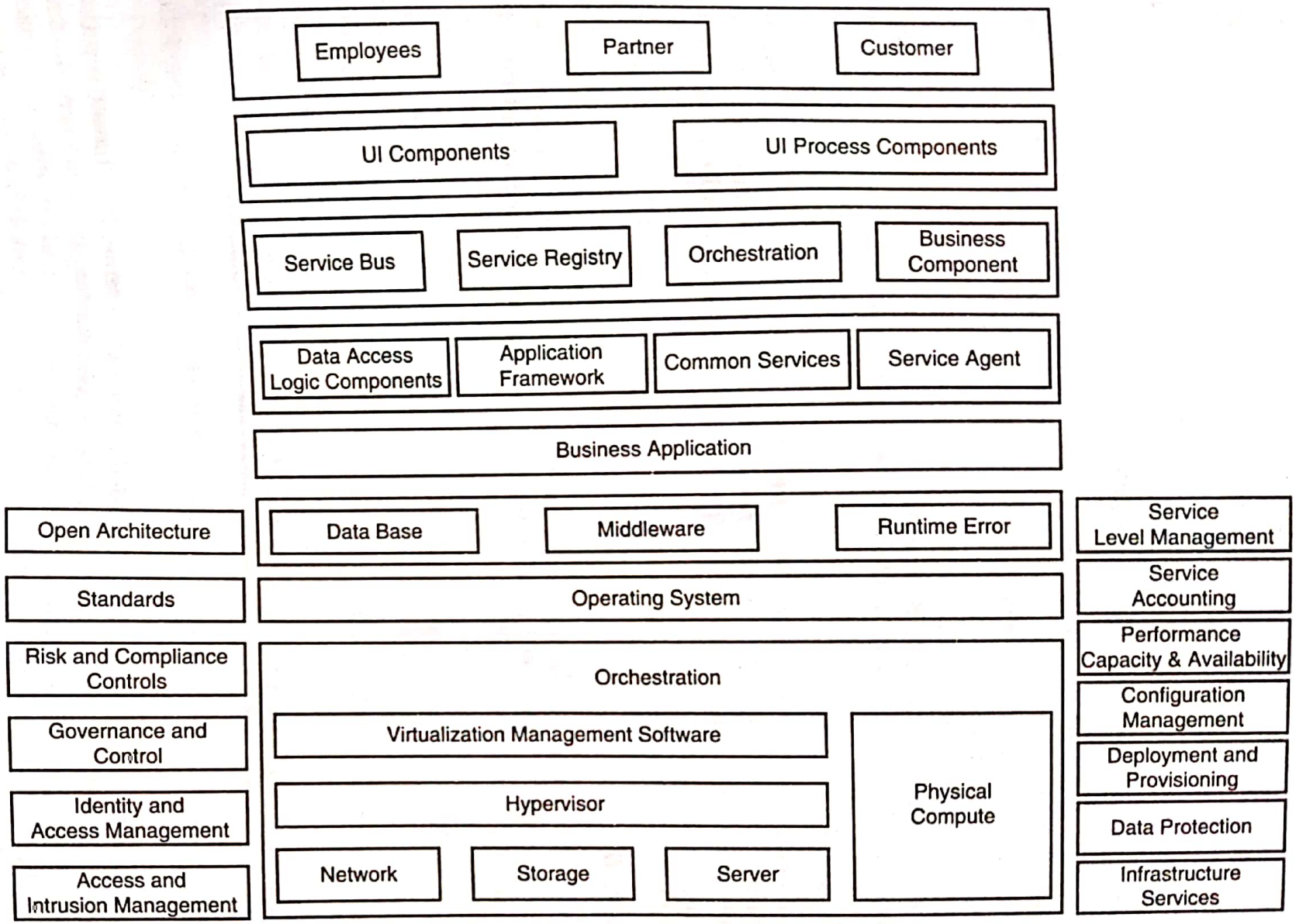

FIGURE 9.2 Cloud IT service management.

Improved Service Levels

1. SOA-based infrastructure helps to adhere to the SLAs efficiently and helps in orchestrating the resources as per the rules and policies.
2. Business analytics helps to decide different predictive, proactive, alternative approaches when we follow SOA.

Efficient Information Management

Efficient centralized virtual environment enables:

1. Information dissemination.
2. Data replication.
3. Business continuity protection.
4. Regulatory compliance.
5. Maximizing resource utilization.
6. Rapid deployment of new applications.
7. More timely response to changing business conditions.

Regulatory Compliance

1. We need monitoring to track the performance of the services to confirm that they comply with the regulatory compliances. Workflow automation helps for the same.
2. It is important to adhere with energy emission regulations and efficiencies to adopt greener solutions.
3. If there is centralized data storage it facilitates and accelerates the audit process.

Energy Efficiency

Energy requirements for the datacenter are rising day by day. To match the requirements, we have the answer: The SOA-based cloud infrastructure. The SOA-based cloud infrastructure substantially improves

1. Computing.
2. Storage.
3. Network utilization.
4. Datacenter energy efficiency.

9.6 SOA-BASED CLOUD INFRASTRUCTURE STEPS

Organizations intent upon leveraging cloud infrastructure should consider the following steps:

1. **Analysis and strategy:** It is recommended to have an incremental, phased approach for adopting SOA and the cloud infrastructure. A good starting point is to conduct a business innovation assessment to identify the business needs and key areas of impact, and use them to develop business value cases for SOA adoption. It is also recommended to conduct an enterprise IT architecture assessment to determine the IT readiness, including the applications and integration capabilities to support the business needs and the current gaps. Furthermore, an IT infrastructure assessment should be conducted to determine the capabilities and entry points for (SOI).
2. **Planning:** Once the business needs and IT gaps are identified, a strategic and tactical planning effort can be launched to develop an IT value case and roadmap for incremental IT transformation to enable business innovation/agility leveraging SOA. Through a careful portfolio analysis and change management process, the current IT efforts can be aligned and adjusted, and tactical/focused projects can be selected based on the strategy and roadmap. At the same time, organizational impacts should be analyzed and a SOA governance model, standards, and guiding principles should be developed to accelerate and manage the pace of the transformation.
3. **Implementation:** It is an excellent idea to establish an enterprise-level SOA, as well as the development and run-time environment standards before the first set of SOA projects are launched. It is best to couple the virtualization projects with SOA so that the benefits of the virtualization can be realized at the lower level (service-level versus server-level). SOA and cloud infrastructure governance should be incorporated into the existing IT governance bodies, and performance and SLAs' impacts should be monitored and managed under the same set of business rules.
4. **Value-driven:** It is important to note that the purposes of SOA and cloud infrastructure are to improve the business performance, flexibility, and agility so that IT can support business at the right speed. All SOA projects should be driven on the basis of business values rather than technical merits alone. Technical merits can only be realized when they match the business needs.

9.6.1 SOA and Cloud Infrastructure

SOA is an approach to decompose business processes and applications into loosely coupled components of service providers and consumers, and then connecting them through an enterprise service bus. It enables enterprises to reuse existing business and IT components quickly to develop new capabilities and software solutions. SOA provides an enabling foundation for enhancing business and IT flexibility and agility. The approach is gaining significant momentum not only in designing new solutions, but also in transforming monolithically defined legacy applications.

We can bank on the traditional IT model that has been successful for the deployments. These models can be based on processing of historical data and transactions. As these are established models, they gel well with the high-structured environment. But they break down when you try to extend them into applications or processes that are not so highly structured. They either feel too complex and static [e.g., long-term enterprise resource planning (ERP) projects] or are plain impossible.

With the advent of Web and Internet, we got the new wave itself for the new paradigm of models. It is based on open standards and linked easily for different requirements and components, which can be used for relatively simple activities like communication, browsing, searching, and sending e-mail. It works incredibly well. But it soon became clear that the Internet standards and mechanisms were needed to be extended to handle more sophisticated applications.

The SOA-based cloud computing model builds on IT and Internet models. It is based on what we call a service-oriented architecture, which essentially provides us with a set of modular components to be defined and manipulated (Web services), and a set of XML-based standards for doing so. Since the characteristics of the components can now be expressed in XML, we can define applications that work and manipulate these modular components. It enables a much more flexible and real-time way of implementing business policies than was possible with more structured computing models.

SOA-based cloud computing is not about technology for the sake of technology – it is about enabling new ways of doing business. It is about helping a company to reach new levels of maturity while continuing to deliver the best in class services with productivity; these are necessary to improve the bottom line.

As the processes are integrated in the SOA environment, it gives an option to the enterprise to deal with any type of situation and answer any type of customer demand with the help of partners, suppliers, and customers.

One should follow the most important approach for SOA to consider not only the technical foundation of the business but also the business' underlying principle as it will help to determine the cost of investment. Now we are in the era where models are evolving and changes are very dynamic; therefore, all the technological steps should have business backup to support it. We can consider different types of capabilities related to the organization, strategic values, and market factors that are driving the business.

SOA means that a company finds a pragmatic balance between technical rigor and time-to-market. IT organizations realize that perfect technology stacks cost far more than they deliver, but they also recognize that delivering key functions and reliability of service make it easy for the employees and customers to use the software, and drive the desired business results.

When we adopt SOA, ROI is also required to understand the value of the investment model. A simplistic approach will be to calculate the cost of change and see that it should not be more than the actual cost of implementation, and highlight the important point to adopt the changed approach.

Diversity in the portfolio gives an alternative to face the risks, it is better to work for a different set of applications and not on a single application. Therefore, it is recommended to give space to a larger range of future prospects and create value to improve the business.

It is important to integrate and control the business functions across all the units of business with the help of partners and customers. So, automation in the process delivery should be valued as it increases transparency in the system and also reduces the manual intervention. Sometimes it alerts the organization about the available opportunities and gets unexpected results out of them.

Competitive advantage and system-based performance are the levers to progress for the business. But the balance between the two is very important. SOA helps to balance both. It helps to consider the risks and overcome with spikes in standards and product deployments. This ensures security, and companies can leverage the power of software application in a more efficient way.

9.7 SOA BUSINESS AND IT SERVICES

We need different management tools for the comprehensive integration of SOA. These tools help to leverage the benefits of infrastructure services. They also help to measure the performance of business functions, and manage the run-time application and system across the portfolio of business functions. These development tools aid to get the specific outcome based on the skills and roles possessed by the people in any organization.

There is a requirement of various modeling tools to depict and simulate the processes by the analysts when they are working on business process management (BPM) projects. These tools help to model information and data flows, and system communications and functionalities. Integration architects are required to model the interconnections and integration existing in the solution. When we follow the SOA implementation, it is evident that the people in the organization will use the systems based on their roles in the organization. The tool environment and deployment framework allows working in an integrated way and using the development tools in a joint manner that ensures collaboration and asset management. So the activities, for example using tools such as version control and project management tools, are provided in the SOA framework under the banner of unified development platform. Generating metrics to measure the performance is pivotal in the progress of the business. SOA services incorporate functions to generate the metrics to control the system and processes. This requires special competencies to deal with the IT operations with the skill sets of experienced business analysts and professionals of an enterprise These capabilities are delivered through a set of comprehensive services that collect and display IT and process-level data, so that business dashboards, administrative dashboards, and other IT level displays can manage system resources and business processes. These tools make it possible for the line of business (LOB) and IT personnel to determine business process paths that may be existing as the inefficient tasks in specific processes or in the relationship of system performance to business process performance, so that IT personnel and assets are tied more directly to the business success of the enterprise.

9.8 SUMMARY

This chapter shows the integration of SOA with the cloud technology. SOA is essentially the idea that displays that companies can treat their applications and processes as defined components that can be mixed and matched at will. SOA is much more than the architecture flavor of the day – in fact, it is a new way of thinking about enabling cloud technology.

APPENDIX

A

Cloud Performance Monitoring Commands

CONTENTS

This appendix discusses several cloud infrastructure performance data collection and performance monitoring commands.

A.1 vmstat COMMAND

vmstat command: Account virtual memory information

Syntax

```
vmstat [-n] [delay [count]]
vmstat [-V]
```

Description

The important factors for virtual memory are paging, swap, memory, traps, blocks, IO and CPU. vmstat statistics gives the vital information regarding paging, swap, memory, traps, blocks, IO and CPU. It gives information about the system from the time of the reboot as an average. We can add the sampling delay for the vmstat as an option.

We will discuss the different parts of the vmstat command.

Process Part

- r field: Total number of runnable process.
- b field: Total number of blocked process.

Memory Part

- Swpd field: Used swap space.
- Free field: Available free RAM.
- Buff field: RAM used for buffers.
- Cache field: RAM used for file system cache.

Swap Part

- Si field: Amount of memory swapped from disk per second.
- So field: Amount of memory swapped to disk per second.

IO Part

- Bi field: Blocks received from disk.
- Bo field: Blocks sent to disk.

System Part

- In field: Number of interrupts per second.
- Cs field: Number of context switches per second.

CPU Part

- Us field: Time spend running user code (non-kernel code).
- Sy field: Time spent running kernel code.
- Id field: Idle time.
- Wa field: Time spent waiting for the IO.

A.2 iostat COMMAND

iostat command: Monitors server input/output (I/O) statistics

Syntax

```
iostat [ ] [ ]
```

Description

It is important to balance the workload of the system for different physical disk devices to understand the load of the IO devices. The iostat command allows administrators to monitor and tune the performance of the system. It gives the flexibility of comparing the average transfer rate of the device.

iostat generates report from the time of the first reboot of the system. It gives the information for the CPU usage for all the processors. It gives the CPU information for each device associated with the system.

CPU report requires percentage of the user, priority, IO wait and CPU idle time. It can be defined as

:* %sys: System kernel-level processor utilization percentage

Device Information

* device: This gives the device name from the /dev directory. The given name possess the mapping form the mount points of the file /etc/fstab.
* tps: This gives the transfers (I/O requests) per second issued to device.
* blk_read/s: This gives the number of blocks read per second from the device.
* blk_wrtn/s: This gives the number of blocks written per second to the device.
* blk_read: This gives the total number of blocks read.
* blk_wrtn: This gives the total number of blocks written.

The information that we get from iostat as stated above can be used to balance the workload of the devices. This option helps us distribute the workload proportionately to the idle devices.

A.3 mpstat COMMAND

mpstat – Provides account processors information.

Syntax

```
mpstat [ ] [  ] [ ]
```

Description

The mpstat command writes to standard output activities for each available processor, processor 0 being the first one. This can also give the global average statistics of the entire system. This command is also suitable for the symmetric multiple processing-based processors.

The interval option is to specify the time gap between the reports. If we give it as 0, it means we are referring to system boot-up time. We can specify the count parameter also with the mpstat command. The command provides the information in the form of multiple reports at different stipulated period of intervals.

Now we will discuss the format of mpstat command in brief.

The report generated by the mpstat command has the following format:

%user	This is the percentage of CPU utilization at the user level.
%nice	This is the percentage of CPU utilization at the user level with nice priority.
%system	This is the percentage of CPU utilization that occurred at kernel level.
%iowait	This is the percentage of time that the CPU or CPUs were idle When the processor got the IO request.
%irq	This the percentage of time spent by the CPU or CPUs to service interrupts.
%soft	This is the the percentage of time spent by the CPU or CPUs to service softirqs.
%idle	This is the percentage of time that the CPU or CPUs were idle when processor did not got the IO request to service.

A.4 netstat COMMAND

netstat – Reports network configuration and activity

Syntax

```
netstat [ ] [ ]
```


Description

The netstat command helps to understand the network configuration and activity. It includes the different functions that we will be discussing in the following sections.

Information about Routing Table

When we use netstat command with -r option, it gives the information about the kernel routing table.

When we use netstat command with -n option, it gives the addresses as dotted quad IP numbers. It helps in avoiding address lookups over the network.

If we are looking for the network gateway information for routing entry points, we can look up the second column of netstat command output. If we are looking for the network mask for this route, it is given by the third column. The fourth column of the netstat command displays the following flags that describe the route:

G	The route uses a gateway.
U	The interface to be used is up.
H	Only a single host can be reached through the route.
D	This route is dynamically created.
M	This route is set if the table entry was modified by an ICMP redirect message.

Network Interface Information

Many times we need the network interface information available in the kernel. For this, we can use the netstat command with the -i and -a option. The -i option gives the information about the network interfaces and -a gives all the network connections and listening ports.

Network Connections Information

To see the status of the network connections, we can use different commands given below. netstat supports a set of options to display active or passive sockets.

-t	This option shows active TCP connections.
-u	This option shows UDP connections.
-w	This option shows RAW connections.
-x	This option shows UNIX socket connections.

A.5 ipcs COMMAND

ipcs – Provides information about active interprocess communication facilities

Syntax

```
ipcs [ ] [ ]
```

Description

There are various interprocess communication (IPC) tools like share memory, message queues and semaphore. We can use the `ipcs` command if we are looking for the information regarding IPC. It gives information regarding the calling process that have the read permission. We can use the `-I` option to get the id to specify the id.

We can use the following ids with `ipcs` command:

`-m` shared memory segments

`-q` message queues

`-s` semaphore arrays

`-a` all (this is the default)

The output format may be specified as follows:

`-t` time

`-p` pid

`-c` creator

`-l` limits

`-u` summary

A.6 ps COMMAND

`ps` – Reports process status

Syntax

```
ps [options]
```

Description

A process is represented by a data structure known as a process control block (PCB) or process in execution. The process image as viewed by the kernel runs in its own user address space that is a protected space, and it cannot be shared by other users. In order to see the details about the status of the process, the `ps` command can be used. To get details about a particular process running on a UNIX system, you can use the `ps` (process status) command.

The following table gives the outcome of a `ps` command. Primarily, it shows four columns that actually depict the status of the process. The column headings have been explained after the table.

PID	TTY	TIME	COMMAND
2330	01	12.09	sh
2340	01	12.09	ps

`PID`: Process Identification Number

`TTY`: Terminal type or input device that the user is using

`TIME`: The time at which the process is being executed

`COMMAND`: Name of the command

The `ps` command has the following options:

- `-f`: In addition to normal ps command, the `-f` option gives the login name, parent PID, amount of CPU time consumed by the process and the command with arguments.

  The `ps` command with the `-f` option is given as

  ```
  $ps -f
  ```

  Given below is the extended information on our process that we get when we use the `ps` command with the `-f` option. This information is given as an eight-column output. Each field is explained after the output.

UID	PID	PPID	C	STIME	TTY	TIME	COMMAND
Kumar	30	1	0	12:09:11	02	2:34	sh
Kumar	89	30	22	12:22:12	02	0.19	Ps -f

  `UID`: User_id

  `PID`: Process_id

  `PPID`: Parent PID

  `C`: Amount of CPU time consumed by the process.

  `STIME`: Time that has been elapsed ever since the birth of the process

  `TTY`: Terminal type or input device that the user is using

  `TIME`: The time at which the process is being executed

  `COMMAND`: Full command with options. So, if you are running the program and forget the exact options, you could use the `-f` option to see the command and argument typed.

- `-u` – Lets you know the activities of the user. This option has to be followed with a user-id.

  Here is another `ps` command.

  ```
  $ps -u kumar
  ```

This command will display what kind of processes the user (kumar) is using.

ps Command Options

The following options available with the ps command perform the respective actions:

-A	Selects all processes
-N	Negates selection
-a	Selects all with a tty except session leaders
-d	Selects all, but omit session leaders
-e	Selects all processes
T	Selects all processes on this terminal
a	Selects all processes on a terminal, including those of other users
g	Selects all, even group leaders (does nothing without sunos settings)
r	Restricts output to running processes
x	Selects processes without controlling ttys

Process Selection with List

-C	Selects by command name
-G	Selects by RGID (supports names)
-U	Selects by RUID (supports names)
-g	Selects by session leader OR by group name
-p	Selects by PID
-s	Selects processes belonging to the sessions given
-t	Select by tty
-u	Selects by effective user ID (supports names)
U	Selects processes for specified users
p	Selects by process ID
t	Selects by tty
--Group	Selects by real group name or ID
--User	Selects by real user name or ID
--group	Selects by effective group name or ID
--pid	Selects by process ID
--sid	Selects by session ID
--tty	Selects by terminal
--user	Selects by effective user name or ID
-123	Implied --sid
123	Implied --pid

ps Output Format

-O	Is preloaded -o
-c	Different scheduler info for -l option
-f	Does full listing
-j	Jobs format
-l	Long format
-o	User-defined format
-y	Do not show flags; show rss in place of addr
O	Is preloaded o (overloaded)
X	Old Linux i386 register format
j	Job control format
l	Display long format
o	Specify user-defined format
s	Display signal format
u	Display user-oriented format
v	Display virtual memory format

A.7 top COMMAND

top – Provides information regarding the running processes

Syntax

```
top [ ] [ ]
```

Description

The top command can be used to see the info for the running processes. It also gives other information such as free memory information like physical and swap.

top Command Options

-d	This specifies the seconds and tenths of seconds of delay. The default is 3 seconds for the two consecutive updates.
-n	This gives the maximum number of iterations or frames.
-p	This monitors only processes with specified process IDs.
-s	This allows you to execute top in secure mode and when it is executed from root.

Description of the Fields

PID	Process ID of the task.
PPID	Parent Process ID, process ID of the parent task.
RUSER	Real User Name, the real user name of the task's owner.
UID	User ID, the effective user ID of the task's owner.
USER	User Name, the effective user name of the task's owner.
GROUP	Group Name, the effective group name of the task's owner.
TTY	Controlling TTY, the name of the controlling terminal.
PR	Priority, the priority of the task.
NI	Nice value, the priority of the task. A negative nice value means higher priority, whereas a positive nice value means lower priority.
%CPU	CPU Usage, this is expressed as a percentage of total CPU time as the task's share of the elapsed CPU time since the last screen update.
TIME	CPU Time, this expressed as the total CPU time the task has used since it started.
TIME+	CPU Time, hundredths, The same as 'TIME', but reflecting more granularity through hundredths of a second.
%MEM	Memory Usage, a task's currently used share of available physical memory.
VIRT	Virtual Image (kb), the total amount of virtual memory used by the task.
SWAP	Swapped Size (kb), the swapped-out portion of a task's total virtual memory image.
RES	Resident Size (kb), the non-swapped physical memory a task has used.
CODE	Code Size (kb), the amount of physical memory devoted to executable code.
DATA	Data+Stack Size (kb), the amount of physical memory devoted to other than executable code.
SHR	Shared Mem Size (kb), the amount of shared memory used by a task.
nFLT	Page Fault Count, the number of major page faults that have occurred for a task.
nDRT	Dirty Pages Count, the number of pages that have been modified since they were last written to disk.
S	Process Status, the status of the task can be one of the following: 'D' = uninterruptible sleep 'R' = running 'S' = sleeping 'T' = traced or stopped 'Z' = zombie
Command	Command line or Program name.

A.8 sar COMMAND

sar- Informs the CPU activity.

Syntax

```
sar [ ] [ ] ...
```

Description

In order to tune system performance, we need information like CPU utilization, memory paging, utilization, IO transfer rates, process and other information. All these information can be captured with the use of sar command. The beauty of this command is that we can collect and save the information captured with the use of the command in the system file itself.

Options

-a	Gives information about use of file access system routines.
-A	Gives information about all data.
-b	Gives information about buffer activity.
-c	Gives information about system calls.
-d	Gives information about activity for each block device.
-g	Gives information about paging activities.
-k	Gives information about kernel memory allocation activities.
-p	Gives information about paging activities.
-q	Gives information about average queue length while occupied and percent of time occupied.
-v	Gives information about status of process, i-node, file tables.
-y	Gives information about TTY device activity.
-o filename	Saves samples in file, filename, in binary format.
-e time	Selects data up to a set time. Default is 18:00.
-f filename	Uses filename as the data source for sar.
-i sec	Selects data at intervals as close as possible to seconds.

A.9 load COMMAND

load – Loads machine code and initializes new commands.

Syntax

```
load fileName
load fileName packageName
load fileName packageName interp
```

Description

The `load` command is used to load the binary code from a file into the application's address space to the interpreter. We can pass the name of the file that contains the code. We can give the path name of the interpreter into which we are looking the load the package.

A.10 `xload` COMMAND

`xload` – Displays the system load average

Syntax

```
xload [ ] [ ] ...
```

Description

This command graphically displays the histogram of the load average of the systems. It is updated periodically.

A.11 `tload` COMMAND

`tload` – Graphical representation of system load average

Syntax

```
tload [ ] [ ] [ ] ...
```

Description

This command prints the graph of the system load average as specified

Options

`s`	The *scale* option allows a vertical scale to be specified for the display.
`d`	The *delay* option sets the delay between graph updates in seconds.

A.12 `uname` COMMAND

`uname` – Prints system information

Syntax

```
uname [ ] [ ] ...
```

Description

Prints certain system information.

`-a, --all`	Prints all information, in the following order:
`-s, --kernel-name`	Prints the kernel name
`-n, --nodename`	Prints the network node hostname
`-r, --kernel-release`	Prints the kernel release
`-v, --kernel-version`	Prints the kernel version
`-m, --machine`	Prints the machine hardware name
`-p, --processor`	Prints the processor type
`-i, --hardware-platform`	Prints the hardware platform details
`-o, --operating-system`	Prints the operating system details

A.13 `opcontrol` COMMAND

`opcontrol` – Controls the profiling session

Syntax

```
opcontrol [ ] [ ]
```

Description

The `opcontrol` command helps us to start or end the profiling session. This command is also useful when we want to dump the profile data and set up the profiling parameters.

Options

`--help`	Shows help message.
`--version`	Shows version.
`--list-events`	Loads the `OProfile` module.

`--status`	Shows the configuration information.
`--start-daemon`	Starts the `OProfile` daemon without starting profiling.
`--start`	Starts data collection with either arguments provided.
`--dump`	Forces a flush of the collected profiling data to the daemon.
`--stop`	Stops data collection.
`--shutdown`	Stops data collection and kills the daemon.
`--reset`	Clears out data from current session, but leaves saved sessions.
`--save`	Saves data from current session to `sessionname`.
`--deinit`	Shuts down daemon. Unloads the `OProfile` module and OProfilefs.
`--buffer-size=num`	Sets kernel buffer to num samples.
`--buffer=watershed=num`	Sets kernel buffer watershed to num samples.

A.14 accton COMMAND

`accton` – Enables/disables system accounting

Syntax

`accton` `[ ]`

Description

This command helps for getting the system accounting information. This is accounted for each process executed to be placed at the end of the file. No options specifies that account information is not activated.

A.15 SUMMARY

Cloud performance monitoring tools and commands help plan, consolidate, monitor, optimize, and automate server virtualization projects and the virtual infrastructure. This helps you identify potential server virtualization candidates and generate reports quickly. It also helps in growing faster, as corrective workflows automatically avoid recurring problems.