Performance and Capacity Themes for Cloud Computing

Selecting workloads for cloud computing

Planning for performance and capacity

Monitoring a cloud environment

Elisabeth Stahl
Andrea Corona
Frank De Gilio
Marcello Demuro
Ann Dowling
Lydia Duijvestijn
Avin Fernandes
Dave Jewell
Bharathraj Keshavamurthy
Shmuel Markovits
Chandrakandh Mouleeswaran
Shawn Raess
Kevin Yu

Redpaper

ibm.com/redbooks

International Technical Support Organization

Performance and Capacity Themes for Cloud Computing

March 2013

REDP-4876-00

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (March 2013)

This edition applies to IBM cloud offerings in general.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

developerWorks®
Global Business Services®
Global Technology Services®
IBM SmartCloud™
IBM®
Power Systems™
PowerVM®
POWER®
Rational®
Redbooks®
Redpaper™
Redbooks (logo) ®
Smarter Commerce™
SPSS®
System x®
Tivoli®
WebSphere®

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redpaper™ is the second in a series that addresses the performance and capacity considerations of the evolving cloud computing model. The first Redpaper publication (*Performance Implications of Cloud Computing*, REDP-4875) introduced cloud computing with its various deployment models, support roles, and offerings along with IT performance and capacity implications associated with these deployment models and offerings.

In this redpaper, we discuss lessons learned in the two years since the first paper was written. We offer practical guidance about how to select workloads that work best with cloud computing, and about how to address areas, such as performance testing, monitoring, service level agreements, and capacity planning considerations for both single and multi-tenancy environments.

We also provide an example of a recent project where cloud computing solved current business needs (such as cost reduction, optimization of infrastructure utilization, and more efficient systems management and reporting capabilities) and how the solution addressed performance and capacity challenges.

We conclude with a summary of the lessons learned and a perspective about how cloud computing can affect performance and capacity in the future.

The team who wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Elisabeth Stahl is Chief Technical Strategist for the IBM Systems and Technology Group and has worked in systems performance for over 25 years. She holds a Bachelor of Arts degree in Mathematics from the University of Pennsylvania and a Master of Business Administration degree from New York University (NYU). Elisabeth is an Open Group Distinguished IT Specialist, an IEEE Senior Member, and member of the IBM Academy of Technology.

Andrea Corona is an IT Architect with international customer experience in multiple industries on analysis, design, implementation, testing, and tuning of complex IT infrastructures, mainly focused on Performance Engineering and Cloud Computing. Since joining IBM in 2003, he has held a number of different positions including field technical support for mainframe sales. He is currently responsible for strategy definition, design of performance testing processes, and tools for the cloud environment of a large Italian customer. He is co-author of *IBM System z Strengths and Values*, SG24-7333 and a university lecturer in Italy. He holds a master's degree in Electronic Engineering from the University of Cagliari.

Frank De Gilio is a Distinguished Engineer from the IBM World Wide Client Technology Centers with a global focus on client enterprise infrastructures. He is the IBM System and Technology Group's Chief Architect for Cloud Computing. He has authored a book about J2EE security and a number of whitepapers about Infrastructure design and management. He is a regular presenter at user conferences and a number of IBM-sponsored venues. Mr. De Gilio holds a number of patents in the system software, Internet, security, and infrastructure management fields.

Marcello Demuro is a Certified IT Architect with extensive experience in performance engineering and system integration engagements. His expertise is in the areas of J2EE and Very Large DB, performance tuning, and capacity planning. In recent projects, he applied his skills in cloud oriented projects, designing a private cloud solution, based on the IBM cloud product, for the largest Italian utility. He also worked on a cloud performance management project for the new cloud infrastructure of a major Italian integrated energy company.

Ann Dowling is an IBM Senior Certified IT Specialist and IBM Certified Consultant with over 29 years of delivery and pre-sales technical experience with IBM in various disciplines including capacity planning, process architecture, performance engineering, and IT accounting. During the first 15 years of Ann's career she was a performance analyst and capacity planner in support of internal IBM and outsourced customers. Ann then moved into consulting with a focus on ITIL process design for Capacity Management. She was lead on offering development and innovation in the areas of IT Service Management and IT Resource Optimization. Ann then joined the Performance Engineering team with Systems Engineering, Architecture, and Testing services. She then resumed her career focus on Performance Engineering and Capacity Planning with the Performance Management & Testing Services practice. Ann is currently a Technical Solution Owner in the IBM Global Technology Services® Service Delivery Center of Excellence. She helped launch and co-lead the Performance & Capacity Community of Practice and is now an active Core Team member.

Lydia Duijvestijn is a senior IT Architect and Performance Engineer in the Global Business Services® organization of IBM, The Netherlands. She is one of the co-leaders of the IBM worldwide Community of Practice for Performance & Capacity. She led a large number of engagements with customers in the area of design for performance, performance testing, and performance troubleshooting. Two of the most recent performance-related international engagements concerned the largest bank of Russia, based in Moscow, and a large bank in the Nordics, based in Stockholm. She was a speaker at IBM internal as well as external conferences about subjects related to IT architecture and performance. She led the IBM team that published the technical paper "Performance Implications of Cloud Computing" in 2010.

Avin Fernandes is a certified IT Technology Architect with over 25 years of experience and currently leads the Performance Engineering COE (Center Of Excellence) for Canada as part of the Performance and Testing Practice. He is also a world wide core team member of the Performance and Capacity Community of Practice (CoP) providing leadership both internally within IBM and externally to academia in support of student research and thesis development. As part of this worldwide P & C CoP, Avin has led projects in SOA Performance Engineering and System Capabilities Assessments. Avin started his career with Air Canada as a Programmer and worked his way up to a Manager of Systems Development. Avin joined IBM when Air Canada outsourced their systems to IBM. Avin has functioned in many roles including chief architect for the development and implementation of Air Canada's existing reservation system RESIII and more recently as the Test Manager for Rogers Enterprise Cloud PAAS solution, which is currently being implemented.

Dave Jewell is a Performance Engineer for the Systems Engineering, Architecture, and Test organization in IBM Global Business Services. Since joining IBM in 1978, Dave has held a number of positions in application development, testing, performance analysis, and performance engineering. He has been involved in performance work for clients in multiple industries over the past 20 years.

Bharathraj Keshavamurthy is a Performance Architect for Enterprise Solutions from Bangalore, India. He works with a software group and has over five years of experience in Performance Engineering of IBM cross brand products and mainly in WebSphere® application integration areas. He holds a Bachelor of Engineering from the University of RVCE, Bangalore, India. His areas of expertise include performance benchmarking of IBM products, end-to-end performance engineering of enterprise solutions, performance architecting, designing, and capacity sizing solutions with IBM product components. He has written and posted many articles onto IBM developerWorks® pertaining to Performance Engineering and also in international science journals.

Shmuel Markovits is a Performance and Capacity SME within Strategic Outsourcing Delivery in Australia. He has over 10 years of experience within performance and capacity areas and more than 30 years overall in the IT industry. He holds a degree in Science and Electrical Engineering from the University of NSW and was a Research Fellow and Visiting lecturer in the University of Technology in Sydney. His areas of expertise include Performance and Capacity analysis in technologies, such as Windows, Intel, VMware, UNIX, and networking. He lectures on Telecommunication Management techniques, virtualization, and network technology areas. He has published in academic areas about Telecommunications topics, such as Information modelling for Telecommunication Management. Prior to that he worked in software development, testing, and deployment.

Chandrakandh Mouleeswaran is a Technical Consultant for the System Technology Group in the IBM India Software Lab. He has over seven years of experience in the Performance Testing and Engineering field. In his current job, he is responsible for enabling ISV applications on IBM platforms and consulting for performance testing, benchmarking, server consolidation, and creating sizing solutions on IBM Power Systems™, System x®, and Storage.

Shawn Raess is a Program Director for Cloud for the World Wide Client Centers and is a practicing Cloud Architect specializing in data center networking. He is an IBM professional with over 30 years of experience working within the IT and Telecommunications industries. Shawn held many positions at IBM ranging from technical, marketing, business development, sales, and project management. He has multiple external networking certifications and holds external credentials with the Project Management Institute (PMP) and the Building Industry Consulting Service International (RCDD).

Kevin Yu is the IBM Software Group Industry Solution's Smarter Commerce™ Performance Architect. He has 14 years of experience in the IT industry. Kevin worked on application servers, enterprise commerce, and database products. He led multi-disciplinary teams to solve critical customer situations and guided many to achieve new records in transaction volumes and sales on their busiest online shopping days of the year. His current role is to lead the creation and refinement of performance roadmaps and strategies for The Smarter Commerce solution.

Thanks to the following people for their contributions to this project:

Mike Ebbers
International Technical Support Organization, Poughkeepsie Center

David Challoner
IBM United Kingdom

Anindya Neogi
IBM India

Richard Raszka
IBM Australia

Special thanks to Dr. Erik Meeuwissen from TNO Delft, The Netherlands, for providing the visualisation of the cloud SLA tree in Figure 1-2.

Now you can become a published author, too!

Here is an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks® publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Introduction to cloud computing

Cloud computing represents another evolutionary step in the continued virtualization of IT systems. As computing power and network bandwidth increased with advances in circuit density, packaging, and design, IBM and other vendors continue to provide new ways of sharing computing resources between increasing numbers of users and their computing workloads. Where virtualization creates the appearance that each user or workload has dedicated computing resources while the actual physical resources are being shared, cloud computing is concerned with the dynamic allocation of these virtual computing resources and the delivery of IT services over the network in response to rapidly changing business needs.

The key terms that we use in this paper are defined in Appendix A, “Key terms” on page 57.

1.1 About cloud computing and performance

Cloud provides benefits throughout an organization, offering reduced time to market, which enables a first mover advantage and standardization that provides reduced complexity in the data center. This leads to improved operational efficiencies and offers the client reduced, predictable annual costs. Other benefits were realized in areas, such as self service, service catalog, automatic provisioning and deprovisioning, and capacity flexibility. A cloud provides clients with features, such as disaster recovery, security, and metering, which enable clients to reduce costs, increase standardization, and improve business continuity

What changed regarding performance and capacity management with the introduction of cloud computing? In one sense, what did not change is that the principles of performance engineering and capacity management still apply. We must still plan, design, implement, tune, and manage our IT solutions with performance and efficiency in mind, if we are to deliver good performing, cost effective solutions to our users and customers, particularly since the underlying physical resources are pooled and shared. Alternately, everything changed since with cloud computing it takes only minutes to allocate new virtual environments automatically, a task that previously required days or weeks to allocate manually in the past. One implication of cloud computing is that many service management processes designed to manage the delivery of information technology (IT) services using the manual paradigm must be revisited to understand how to apply IT service management principles in the cloud computing paradigm.

The experiences of the IT industry with cloud computing taught us the following lessons:

- We must manage well; otherwise, we might find ourselves in the midst of an unmanageable sprawl of virtualized environments.
- We must standardize so that we can offer a positive, repeatable, and successful experience to cloud computing users and customers.
- We must automate because standardization of our approaches to IT service management are not possible in the dynamic environment of cloud computing.

We believe these essential lessons apply to performance and capacity as much as to other aspects of IT service management. Keep these lessons in mind as we continue through this Redpaper.

1.2 Business roles and service level agreements in the cloud

In keeping with ISO/IEC[1] standard 42010, the parties that fulfill the different roles can be considered stakeholders or stakeholder groups.

1.2.1 Business roles

In the literature about service oriented and cloud computing, it is quite common to distinguish between one or more parties that provide services and one or more parties that consume these services. We adopt these roles in this Redpaper as well. Next to the business roles of cloud service consumer and cloud service provider, other roles can be distinguished, such as the cloud service integrator and the cloud service broker:

- A Cloud Service Consumer (CSC) is a party that issues a request for service from the cloud and thus triggers the composition and execution of the requested service.

[1] International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)

- A Cloud Service Integrator (CSI) is a party that provides composite services from the cloud to Cloud Service Consumers. To do so, they must identify, request, negotiate, and aggregate services from various Cloud Service Providers. In general, the CSI can assume the roles of both provider and consumer of cloud services.
- A Cloud Service Provider (CSP) is a party that provides an atomic (or composite) service from the cloud, either to the CSI or directly to the CSC.
- A Cloud Service Broker (CSB) is a trusted third-party that matches the set of requirements of the CSI (in its role as consumer) to a list or set of available services from several CSPs, based on their published functionality and quality of service (QoS) capabilities. The CSB is the mediator in the matching process between the CSI and the CSPs. Given a predefined service level for a given service, the CSB will propose those services that match that service level. Figure 1-1 summarizes the different cloud business roles.

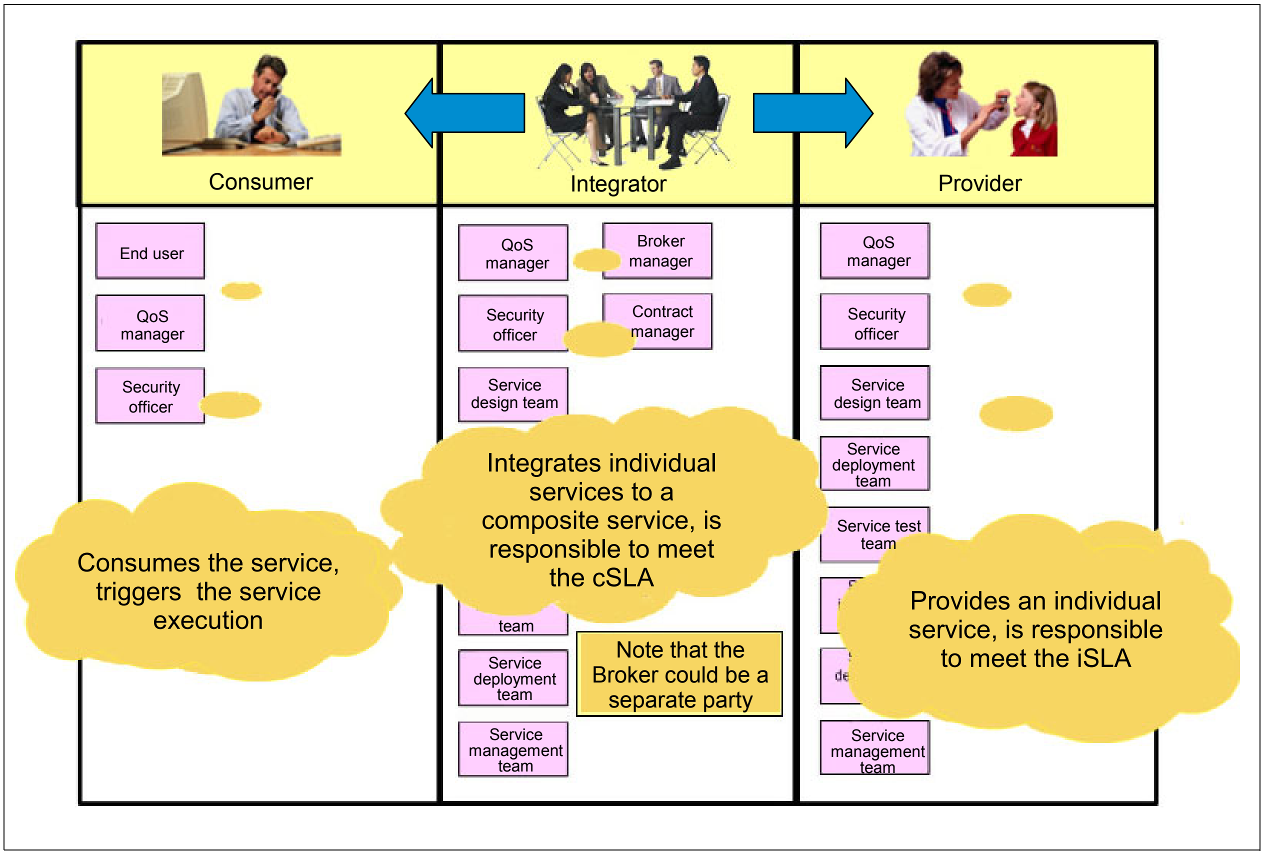


Figure 1-1 Cloud business roles

A business party can play one or more of the business roles in Figure 1-1. For example, an integrator offering a location-based information service is a CSC of the location service (offered by a CSP), and can itself be the CSP of this information service.

1.2.2 Service Level Agreements

Typically, in a business environment Service Level Agreements (SLAs) are negotiated between service providers and service consumers. A service integrator will typically offer an end-to-end SLA to its service consumers. This end-to-end SLA depends on the SLAs that the service integrator has with its service providers (typically referred to as subcontractors). The

SLA between the service integrator and the service consumer is referred to as cSLA; whereas, the SLAs between the service integrator and its service providers are referred to as iSLA1 and iSLA2 respectively.

Today SLAs are often crafted in natural language documents that are ambiguous and leave room for interpretation. However, service level requirements for performance must be defined in a precise way, taking into account the specific properties of business transactions and specifying realistic percentiles. To enable the automated testing and monitoring of the service level requirements for performance that is needed to support on demand provisioning through the cloud, the introduction of a precise and SMART SLA definition framework, such as WSLA[2], is a prerequisite.

1.3 Cloud scope and openness

The three possible cloud models, *public*, *private*, and *hybrid* cloud were introduced in our Redpaper, *Performance Implications of Cloud Computing*, REDP-4875. In this section, we briefly recap the definitions of the three models that vary in scope and openness of the cloud.

A public cloud is owned and managed by a third-party service provider, and access is by subscription. A public cloud offers a set of standardized business processes, applications, and infrastructure services on a flexible price-per-use basis.

Service Level Agreements for performance and scalability in a public cloud are likely to be standardized and predefined. The same is true for workload modeling and performance testing facilities and for capacity management.

Private clouds are owned and used by a single organization. They offer many of the same benefits as public clouds, and they give the owning organization greater flexibility and control. A private cloud provides more ability to customize, drives efficiency, and retains the ability to standardize and implement organizational best practices.

A private cloud offers more room for a customized approach towards performance aspects. Service Level Agreements are typically negotiated between provider and consumer and workload modeling, performance testing, and capacity management can be customized to the needs of the consumer. Figure 1-2 contrasts public and private clouds.

[2] References to WSLA:
Article (WSLA2002) Keller, A. & Ludwig, H. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services Journal of Network and Systems Management, Springer New York, 2003, 11, 57-81

Manual (WSLA1.0) Ludwig, H.; Keller, A.; Dan, A.; King, R. P. & Franck, R. Web Service Level Agreement (WSLA) Language Specification 2003

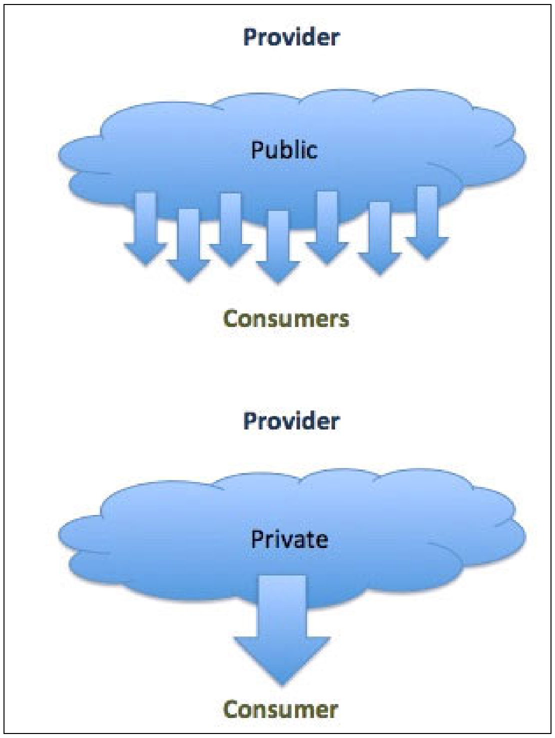


Figure 1-2 Cloud deployment models

Many organizations embrace both public and private cloud computing by integrating the two models into hybrid clouds.

While there might be a number of business and technical factors that drive the decision on which cloud deployment model to use, each model can have significant implications for performance and capacity as well.

1.4 Cloud service paradigms

The service depth (or stack) provided by the cloud can vary. Here are the different layers, starting at the bottom:

- Infrastructure as a Service (IaaS) is a provisioning model in which an organization outsources the equipment used to support operations, including storage, hardware, servers, and networking components. The service provider owns the equipment and is responsible for housing, running, and maintaining it. The client typically pays on a per-use basis.[3]
- Platform as a Service (PaaS) is the capability provided to the consumer to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment. [4]
- Software as a Service (SaaS) is the capability provided to the consumer to use the provider's applications running on a cloud infrastructure. The applications are accessible

[3] http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[4] http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

from various client devices through either a thin client interface, such as a web browser (for example, web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. [4]

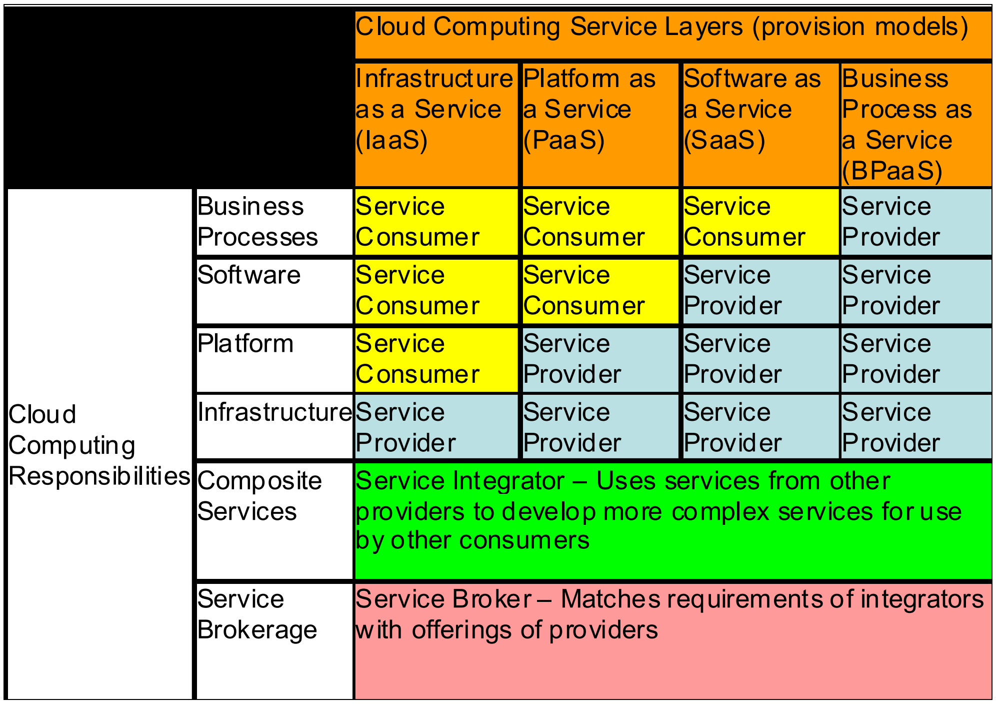
- Business Process Services (BPaaS) are any business processes delivered through the cloud service model. [4]

As more of the services stack is transitioned to cloud computing, more responsibility shifts from the Service Consumer to the Service Provider.

An abundance of cloud services at the IaaS level is available. Various providers, including Amazon, Google, IBM and others, offer different types of on demand storage and processing capacity. Cloud service offerings are known to shift from the lower, technology-oriented levels of the software stack to the higher (application- and information-oriented) levels. More and more applications that bring real value to the business will become available on an “as-a-Service” basis from the cloud. As a result, performance testing and monitoring of cloud services can no longer rely on just business use cases and workload. The challenge is to define the test cases in such a way that they can serve multiple potential consumers.

1.5 Cloud responsibilities

The purpose of cloud computing is to enable the rapid specification, allocation, and delivery of IT services over the network. Figure 1-3 on page 6 depicts the relationship between the layers of cloud computing services and the key stakeholder roles and responsibilities associated with those services.



		Cloud Computing Service Layers (provision models)			
		Infrastructure as a Service (IaaS)	Platform as a Service (PaaS)	Software as a Service (SaaS)	Business Process as a Service (BPaaS)
Cloud Computing Responsibilities	Business Processes	Service Consumer	Service Consumer	Service Consumer	Service Provider
	Software	Service Consumer	Service Consumer	Service Provider	Service Provider
	Platform	Service Consumer	Service Provider	Service Provider	Service Provider
	Infrastructure	Service Provider	Service Provider	Service Provider	Service Provider
	Composite Services	Service Integrator – Uses services from other providers to develop more complex services for use by other consumers			
	Service Brokerage	Service Broker – Matches requirements of integrators with offerings of providers			

Figure 1-3 Relationship of roles and responsibilities to services

As shown in Figure 1-3, the service consumer is responsible for the business process if the provisioning model is IaaS, PaaS, or SaaS. When the business process is delivered as a service, such as in the BPaaS provisioning model, it is the responsibility of the provider. Similarly the installation and management of software is the responsibility of the service consumer in the IaaS and PaaS provisioning models, whereas it becomes the service provider's responsibility in the SaaS and BPaaS provisioning models. The platform is the responsibility of the service provider in all provisioning models except IaaS; whereas, the infrastructure is the responsibility of the service provider in all provisioning models. Composite services are provided by a service integrator. Service brokerage is provided by a service broker.

1.6 Designing for cloud

Whether you are custom-developing or deploying a vendor's software in the cloud, consider the design of this software. Many designers are familiar with providing solutions for a specified infrastructure environment or to a limited capacity but not to a dynamically allocated infrastructure, such as a cloud offers. Some of the considerations that must be addressed to ensure the integrity of the elastic capability of the cloud are:

- Load management of users across addition instances
- Management of user sessions in event of failure
- Management of resource connections such as database and queue depths

The solution that is deployed to the cloud must consider these aspects. In addition, you might need to revise common techniques for maintaining user sessions in memory to a more persistent storage, such as a database. Consideration also needs to be provided about how to configure database connection limits. For example, do you make it unlimited or constrain it to estimate loads or do you provide a federated solution? When the capacity of the system increases, what happens to queue depths when using queuing infrastructure software and the database connections?

The designers of today need to make this adjustment to not only focus on the functional and performance side of designing their software but also understand how their software can scale and be as dynamically elastic and flexible as the cloud can be. The skill set of the designer needs to expand and be strengthened in the aspects of cloud architecture and the capabilities provided by the provider to ensure their software is not limited by the design.

1.6.1 Architecture and designing for cloud

There are many areas that need to be considered when developing an architecture for cloud. The IBM architect community relies heavily on its IBM Cloud Computing Reference Architecture (CCRA) to guide them when designing private, hybrid, and public cloud environments. It is based on open standards and delivers enterprise-class security to meet local, regional, and national compliance for privacy and governance. See Figure 1-4.

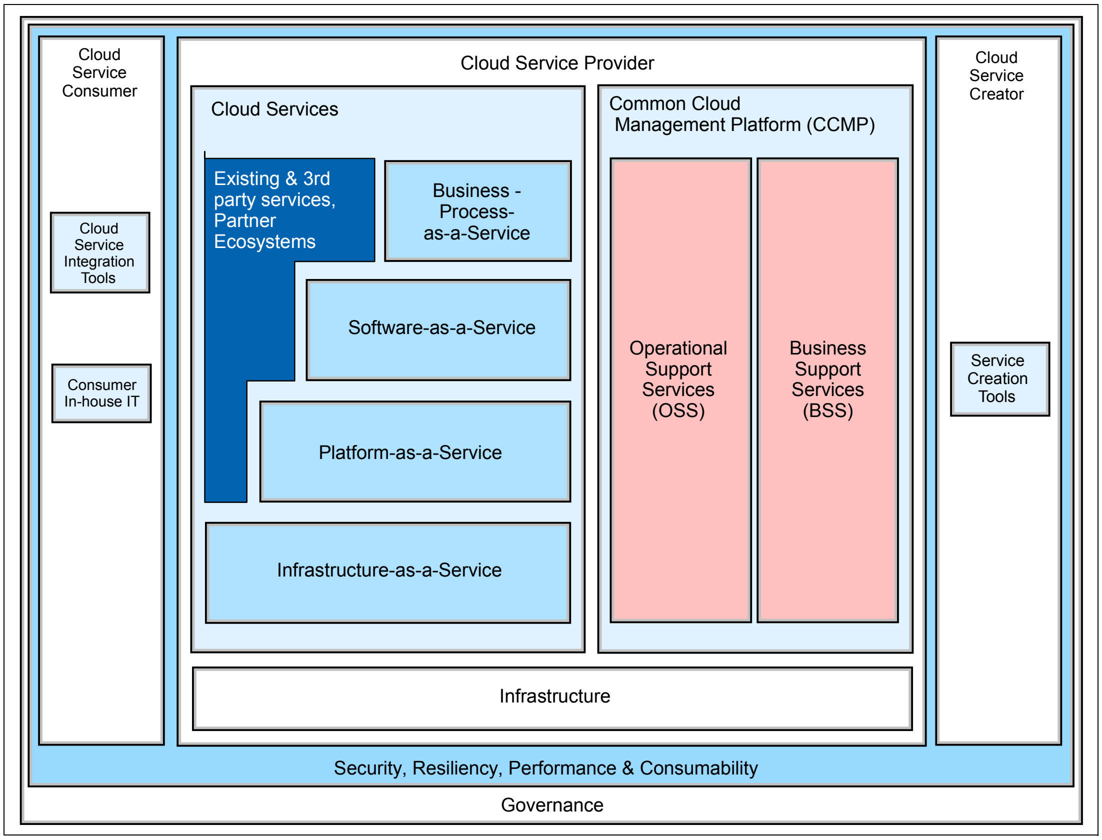

Cloud Service Consumer
Cloud Service Integration Tools
Consumer In-house IT
Cloud Service Provider
Cloud Services
Existing & 3rd party services, Partner Ecosystems
Business - Process- as-a-Service
Software-as-a-Service
Platform-as-a-Service
Infrastructure-as-a-Service
Common Cloud Management Platform (CCMP)
Operational Support Services (OSS)
Business Support Services (BSS)
Infrastructure
Cloud Service Creator
Service Creation Tools
Security, Resiliency, Performance & Consumability
Governance

Figure 1-4 Cloud architecture

It combines powerful automation and services management (low touch) with rich business management functions for fully integrated, top-to-bottom management of cloud infrastructure and cloud services. It has a full spectrum of cloud service models, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and Business Process as a Service (BPaaS). It enables the flexible scaling and resiliency required for successful cloud economics and return on investment (ROI). It facilitates seamless integration into existing customers' environments. It is based on our industry-leading expertise with SOA for building services and service-oriented architectures.

Our cloud architecture covers a broad range of important cloud-specific topics that start with architectural guiding principles and end with metering and accounting. *Performance and scalability* is a key topic when creating cloud services. Figure 1-5 lists the key factors that an architect must consider on every cloud engagement, which are: workproducts/artifacts (architectural principles, architecture overview, standards, use cases, NFRs, component model, operational model, and architectural decisions) and qualities that can be covered in an NFR document (consumability, security, resiliency, multi-tenancy).

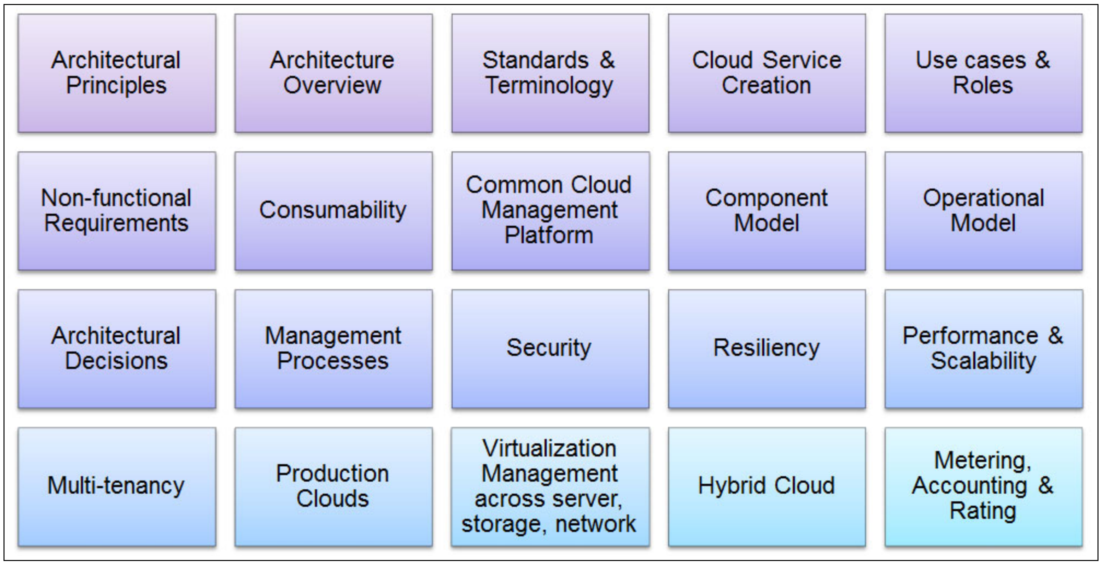


Figure 1-5 Key factors for a cloud services architecture

1.7 Cloud performance and scalability

In this section, we discuss some considerations for performance and scalability.

1.7.1 Scenarios, use cases, KPIs, and measurements

The cloud infrastructure and services must meet the performance and scalability requirements as defined by the service level agreements of the service offering. Because there are several terms used in describing performance and scalability goals, it is helpful to make some distinctions.

Table 1-1 Definitions of terms

Term	Meaning	When used	Examples
Performance and scalability requirements	Performance and scalability characteristics a solution must have to meet the needs of the business	During solution architecture, design, implementation, and testing phases to ensure goals can be met prior to deployment	The system shall handle 6,000 transactions per minute with 95th percentile response time not to exceed 5 seconds.
Service levels	Operational goals to be attained by an IT service that are meaningful to both consumers and providers of IT services	Upon deployment of IT services so that the operations team can make arrangements to objectively measure, report, and manage relevant service levels NOTE: Service levels can (and typically do) include operational attributes beyond just performance and scalability.	► Responsiveness ► Utilization ► Availability ► Problem turnaround
Service Level Objective (SLO)	Measurable goal that constitutes service level attainment		
Service Level Agreement (SLA)	Measurable level of service that a provider commits to provide to a consumer, often with financial and contractual implications		

Term	Meaning	When used	Examples
Key Performance Indicators (KPIs)	Specific measurements or metrics deemed indicative of service level attainment	When tracking and reporting production service level attainment over time	► Percent of the time system was available (or down) during committed availability window ► Percent of time exceeded 90% utilization of budgeted system resources ► 95th percentile response time by transaction

Because cloud systems are built to handle changing workloads in a flexible manner, it is important to state our requirements, service levels, and KPIs so that we can confirm the ability of workloads to grow without impacting responsiveness, throughput, or availability. For scalability, this means we must have a maximum workload level in mind, and the relationship between processing volumes and system resource utilization must be as linear as possible. For performance, our KPIs must have corresponding thresholds that can, among other things, alert us to possible issues with the suitability of the solution to the cloud environment.

For a specific Cloud Computing infrastructure implementation, the performance and scale SLAs need to be formally described to enable the correct definition of the Cloud Computing infrastructure architecture, operational modeling, and the correct selection and sizing of the software, server, and network storage infrastructure. In the context of the CCMP Reference Architecture, the approach selected defines the set of elements that need to be evaluated in the definition of the performance and scale SLA for a specific Cloud implementation based on the CCMP RA architecture, component, and operational models. The key elements that were identified are:

- The set of use cases and related actors that are critical for the scale and performance aspect of the specific cloud implementation
- Scenarios and factors that influence the performance and scale of cloud infrastructure
- The Key Performance Indicator that can be selected for characterizing the performance and scale service level objectives
- The set of critical metrics related to the infrastructure that can be used to monitor and evaluate the performance of the cloud solution

With regard to performance, the performance sensitiveness test can be used to derive what use cases / business transactions qualify. The following criteria is applied to identify a user transaction as performance sensitive:

- Does unacceptable performance of the transaction have an immediate effect on user satisfaction?
- Is the transaction complex, in the sense that:
  - It leads to the execution of a high number of system-transactions
  - It affects a large number of components and connections?
- Is the transaction executed frequently?
- Does the transaction involve large volumes (of data)?

Use cases and roles

Three roles can be identified in the Cloud Computing domain, which we discussed in 1.2.1, “Business roles” on page 2:

- Cloud Service Consumer
- Cloud Service Provider

- Cloud Service Integrator

Figure 1-6 shows a graphical overview of the perspectives and their interaction patterns.

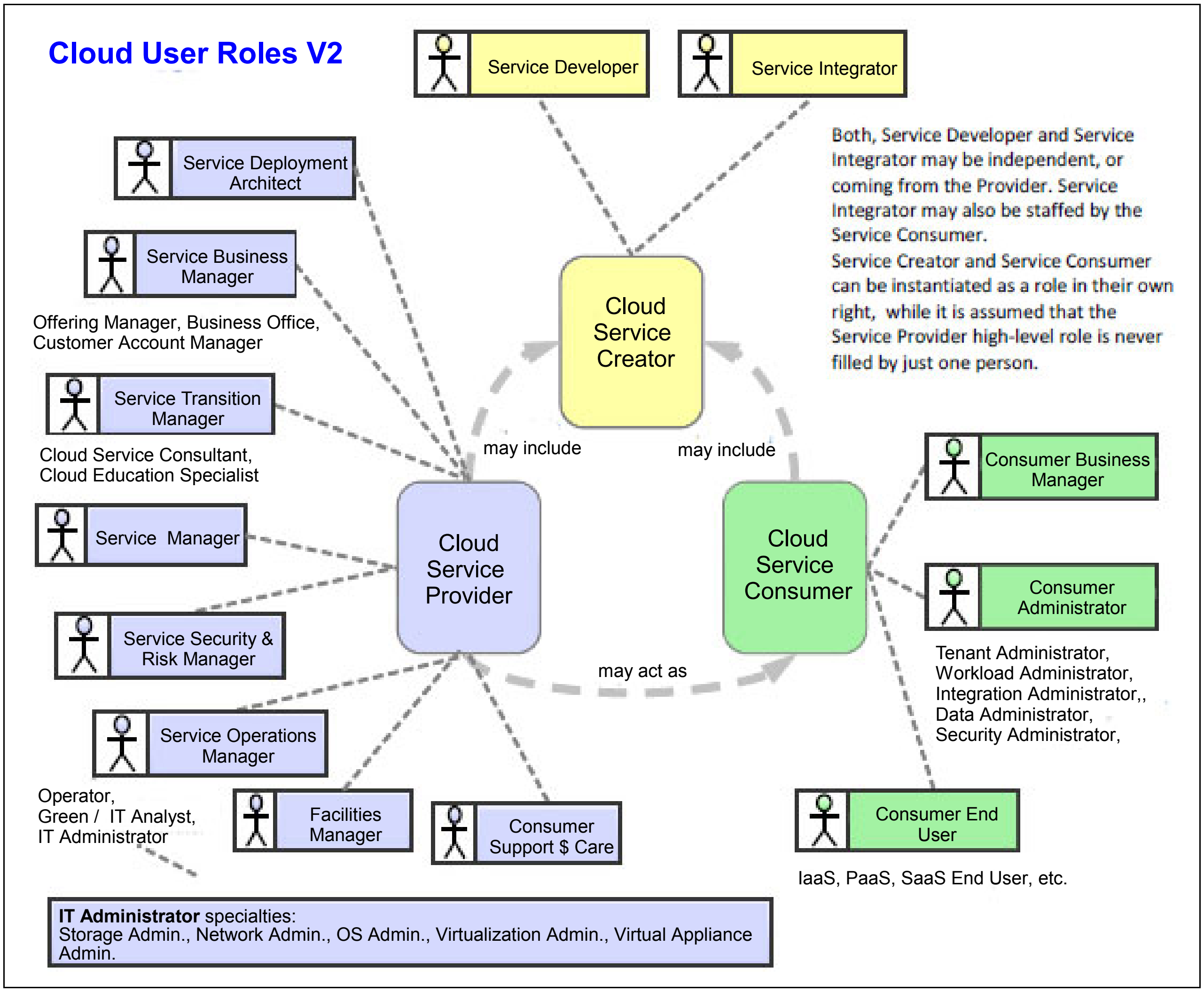

Cloud User Roles V2
Service Developer
Service Integrator
Both, Service Developer and Service Integrator may be independent, or coming from the Provider. Service Integrator may also be staffed by the Service Consumer.
Service Creator and Service Consumer can be instantiated as a role in their own right, while it is assumed that the Service Provider high-level role is never filled by just one person.
Service Deployment Architect
Service Business Manager
Offering Manager, Business Office, Customer Account Manager
Service Transition Manager
Cloud Service Consultant, Cloud Education Specialist
Service Manager
Service Security & Risk Manager
Service Operations Manager
Operator, Green / IT Analyst, IT Administrator
Facilities Manager
Consumer Support $ Care
Cloud Service Creator
may include
may include
Cloud Service Provider
Cloud Service Consumer
may act as
Consumer Business Manager
Consumer Administrator
Tenant Administrator, Workload Administrator, Integration Administrator,, Data Administrator, Security Administrator,
Consumer End User
IaaS, PaaS, SaaS End User, etc.
IT Administrator specialties: Storage Admin., Network Admin., OS Admin., Virtualization Admin., Virtual Appliance Admin.

Figure 1-6 Cloud performance perspectives—roles view

Assess performance and scalability from all three roles, as each of them expresses its own set of goals and KPIs that can be conflicting. Moreover, the architectural patterns available can be more demanding of some resources instead of other ones or change the consuming time of the same amount of resource, favoring in this way some groups of actors over other groups. Be sure to adopt an approach that takes into account all the points of view when evaluating and contrasting alternative architectural options.

2

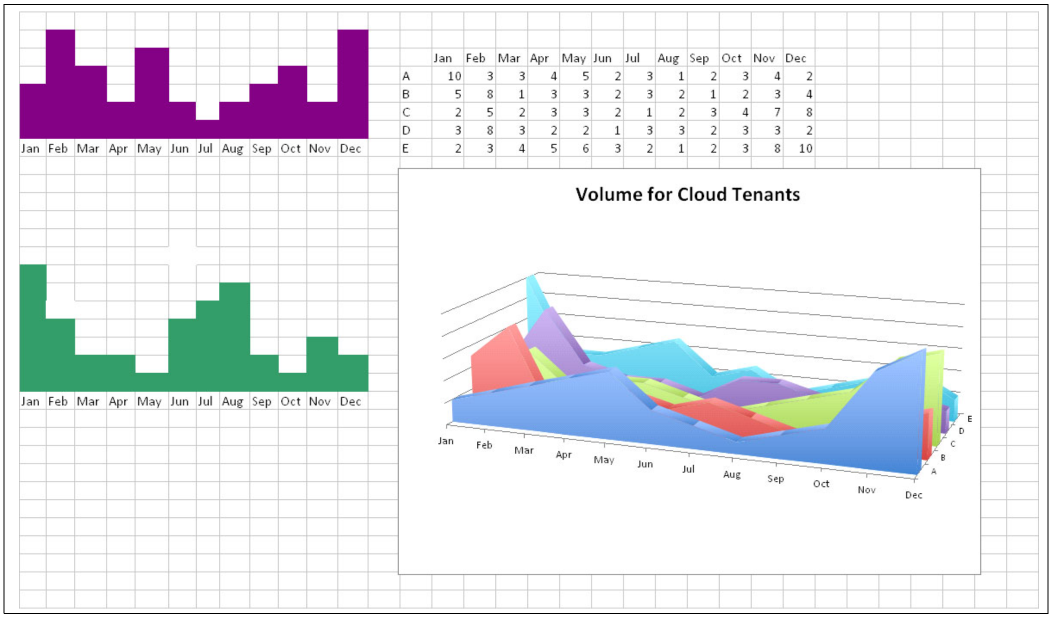
Cloud computing workloads

There is a tremendous amount of hype associated with cloud computing and what cloud can and cannot do. This speculation makes it difficult to understand where cloud can be employed to increase the efficiency and effectiveness of the data center. Understanding what can run in a cloud means the difference between saving money and wasting it because trying to put the wrong workloads into the cloud will cost more than it can save.

2.1 Introduction

One of the most important questions surrounding the study of Cloud Computing and performance is the following conundrum: which specific workloads run best in a cloud environment.

A solution that is optimized for a particular workload (or set of workloads) must correlate with the magnitude, scale, and type of business application. You can determine whether a workload can be optimized in a cloud computing environment by further study and analysis, using data similar to Figure 2-1.



	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
A	10	3	3	4	5	2	3	1	2	3	4	2
B	5	8	1	3	3	2	3	2	1	2	3	4
C	2	5	2	3	3	2	1	2	3	4	7	8
D	3	8	3	2	2	1	3	3	2	3	3	2
E	2	3	4	5	6	3	2	1	2	3	8	10

Figure 2-1 Workload and volume in a multi-tenant environment

2.2 Cloud workloads

Before we can identify which workloads fit into a cloud, we must define a set of terms.

2.2.1 What is a workload?

The problem with a term like workload is that it can mean different things to different people. For the purposes of this paper, workload is defined as a processing done on a set of data to produce a particular result. Note that this is most likely going to be a relatively small part of some larger computing activity. Most often an application is an amalgamation of multiple workloads. This is important because people often think of an application as fitting in a cloud or not. Because an application can be made up of several parts, each having a separate workload, those parts can easily fit into a cloud while other parts need to be run in a more traditional environment.

Types of workloads

Workloads refer to the type and rate of requests sent to the system plus the software and application programs to be executed. Not all workloads are the same. Some workloads run efficiently on a cloud platform and some do not.

A strict online transaction processing (OLTP) workload is characterized by fast, reliable transactions with relatively small data volumes, a large number of users and random read/write I/O with dramatic peaks. The computations are usually quite simple; however, the workload requires complex serialization and data locking of technological mechanisms that are not necessarily conducive to a cloud environment.

Alternatively, a workload that is more harmonious with a cloud environment is one with asynchronous processes. These processes can be computed and offloaded to various areas in the cloud environment. They then come back together and the application can continue. Workloads that are parallel and atomic take advantage of this environment to the fullest. An example of an asynchronous process is an application where you need to compare cities. The process might offload the computations to compare weather in Miami with weather in Cleveland. These comparisons in temperature, humidity, snowfall, and so on are then sent back to the overall application.

A web-based or high performance computing (HPC) scientific workload can also fit quite well with a cloud model, especially if there are computations or algorithms that can be off-loaded and executed concurrently with extreme parallel processing.

2.2.2 Does it run in a cloud or on a cloud?

People often use the term cloud generically, which is fine for high-level conversations. But if we are going to understand the value of the technology, we must differentiate between things that run in the environment and things that run on the environment. The difference revolves around the usage of the service. For example, if I use an IaaS solution, such as the ones offered by companies, such as IBM, Amazon, or Rackspace, I receive a virtual environment. While that virtual instance is provisioned, managed, and possibly monitored by the cloud, the user of the instance is running on the cloud. The user's code is not actively leveraging the cloud infrastructure. Indeed the service management functions are hidden from the users view entirely. It is merely using the instance provided. Now a user can gather multiple instances and create his/her own cloud environment, but that cloud environment runs *on* a cloud.

On the other hand, the services that are devoted to making an IaaS image available are running *in* the cloud. These systems are concerned with making the instance available, removing it from availability, and generally making sure that it is healthy and up to date on fixes. The key thing to remember is that any workload can run on a cloud. That is, I can run anything in an instance that is hosted by a cloud service. Alternately, not every workload fits in a cloud.

2.2.3 What workloads are fit for cloud?

The feasibility of a workload for a cloud depends greatly on the cloud infrastructure. At the time of this publication there is no prevalent cloud infrastructure supporting complex computing capabilities (such as transaction management), which limits what can be done in a cloud environment.

Today, workloads that are appropriate for cloud are ones that run within the confines of a single instance of a server or ones that lend themselves to asynchronous activity. This of course lets out many workloads that fit in a business environment.

Any workload that has synchronous activity (processing on B cannot start until processing on A is complete), must be confined to a single server instance. If that synchronous activity is intended to flow across systems, is inappropriate for cloud. If, however, the processing is not synchronous or it can be housed within a single server environment, it lends itself to cloud.

For example, businesses that are interested in hosting a collaboration event, such as a web conference, can go to the IBM SmartCloud Meetings, which start an instance of a web conference and allow others to connect to it. As long as users have the URL and password to get in, they can connect to this service in the cloud. Because the collaboration is tied to a specific instance, it lends itself to cloud. Users of SmartCloud Meetings do not care about the underlying infrastructure, nor do they have a need for sophisticated Infrastructure. Compute power is what is needed, enough to make the sharing of information expeditious. Because there is no need to synchronize the activity between this machine and other servers, the complexity is managed within a single environment.

Contrast that to systems that have transaction managers with two phase commit. These systems require the system to keep track of each activity within a unit of work to ensure that things happen in the correct order and that if something does not work, changes are backed out of the activities that led up to the error. These systems must be able to checkpoint work and perform transactional logging to ensure that the integrity of the activities is maintained.

Finally, it is important to understand if the code supports running in a cloud. Some software has dependencies that make it unsuitable for cloud. It is always good to check with the software provider to ensure that it is supported to run in a cloud. If it is written locally, it might have to be adapted to fit into the cloud. If so, it is a good idea to ensure that the cost of converting the software is added to the cost of conversion to a cloud environment.

When trying to determine if a workload is fit for a cloud deployment model, ask yourself some questions:

- Does the workload require transactional integrity? Does the infrastructure need to enforce that steps leading up to an error are reversed in case of an error?
- Does the workload require a strict cardinality of events? Does step B need to wait until step A has completed? Will those steps need to execute across machines?
- Does software have to be modified to work in a cloud environment?

Answering *yes* to any of these questions probably means that they are unsuitable for cloud.

2.2.4 Workloads are not applications

Just because a particular workload does not lend itself to a cloud deployment model does not mean that the entire application is unsuitable for cloud. If an application consists of multiple workloads, some of those components can lend themselves to be cloud enabled.

For example, suppose you have a cloud application that provides a set of banking services. While the movement of money from one account to another most definitely requires the kind of transactional integrity that does not lend itself to cloud, there are a number of components in that application that can take advantage of a cloud. If the application provides the user with mortgage payment information, that workload can be handily managed in a cloud environment. If the application wants to determine additional offers that can be made to a user, those workloads can easily take advantage of an infrastructure. Even the fraud detection services can be implemented in a cloud.

Cloud workloads can also be fairly database intensive as long as the data can be parallelized. Facebook is database intensive, yet it is one of the most popular cloud environments in operation.

2.3 Ultimately how will cloud serve IT needs

Cloud is an operational model that can help maximize efficiency. It enables a single operator to manage a larger portfolio of services at a lower price point. It allows systems to be managed more effectively and securely with more accuracy than in the traditional IT model. All of these benefits are predicated on the ability of IT to operate in a much more cohesive way. If IT cannot embrace the siloless operational model, it will never reap the true benefits of cloud.

A cloud is more than a set of products managing a workload. It requires the administrator to manage compute, storage, and network resources as a unit without interference from different teams. It requires automation to ensure that the complexities inherent in each of these infrastructure components are hidden from the administrator. Most of all it requires an adherence to a standard for each of the major components and their related processes. Without the standards, there can be no automation. Without the automation there can be no cloud. Without the appropriate operational model, all the products in the world will not make cloud a valuable IT asset.

2.4 Smarter industry example: Enterprise cloud solution

The purpose of this cloud initiative is to provide a cloud-centric solution that supports the portal. This will ultimately provide a single integrated solution to support the enterprise and business unit analytics and reporting across all lines of business. This approach is expected to reduce cost, eliminate redundancies by using its infrastructure optimally, and provide more efficient management and reporting capabilities. The major components in this environment are:

- L3 Data Center
- Network connectivity
- ISDM provisioning in a virtualized environment with SA&D automation.
- CCMP managed environment based on the enterprise cloud design
- Existing, new and changed processes

Those major components are to be deployed in the following environments:

- Production
- Performance
- Development & Test
- Disaster Recovery
- Sandbox where all the initial testing was done
- Lab for testing cloud components prior to deployment after the cloud is managed in steady state

In addition, there was a self-service catalog portal for the creation of all the services within each of the environments. It included:

- Provisioning a server image
- De-provisioning a server image
- Provisioning a set of VMs
- De-provisioning a set of VMs
- Expand/contract an existing VM

- Expand storage
- Extend or reduce the “lease” on an environment
- An equivalent of for “unmanaged” services where the end user manages the development/test environment with limits being set.

The architecture of the enterprise cloud environment included System x3850 and Power 770 hardware, all managed in a virtualized environment using EXSi and IBM PowerVM®, as shown in Figure 2-2.

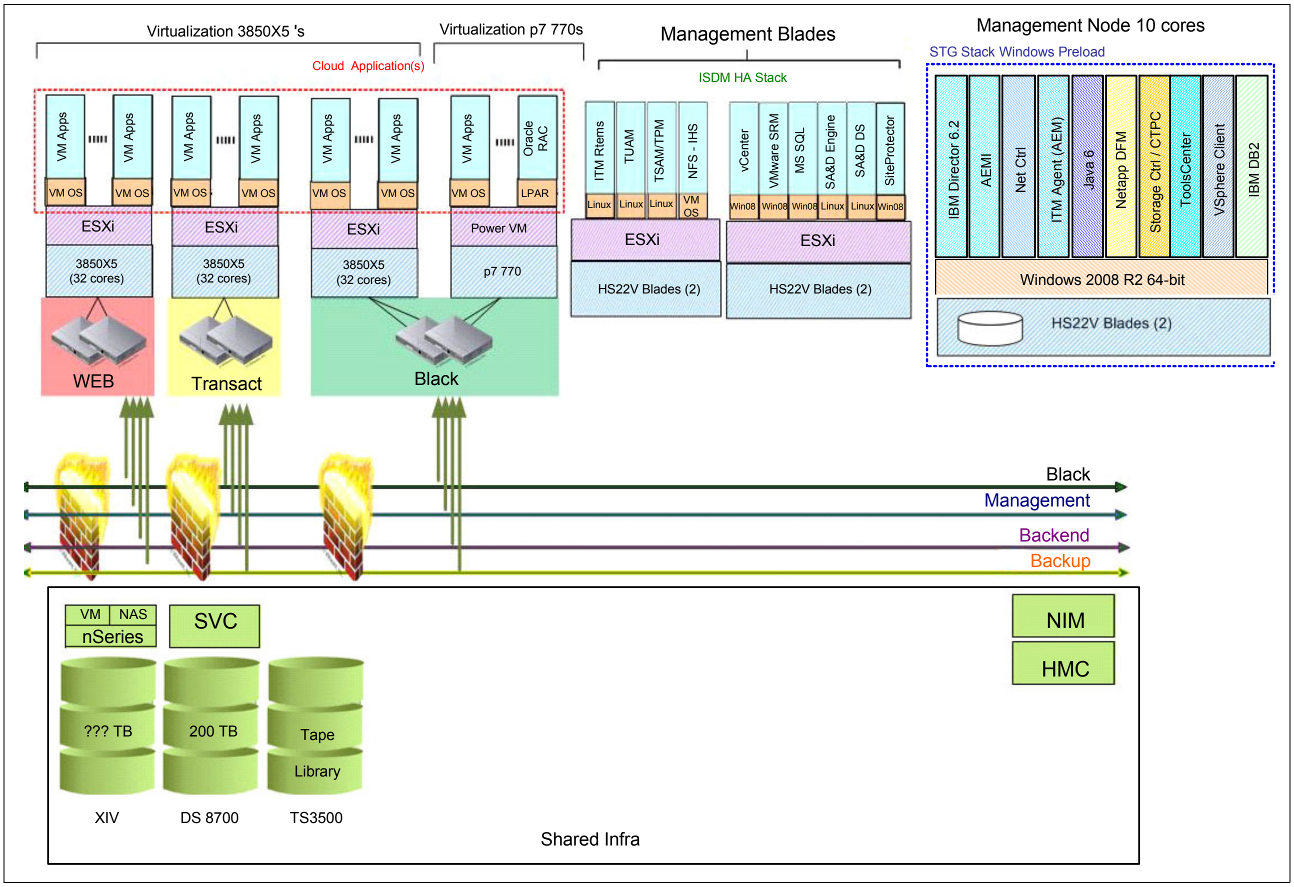


Figure 2-2 Sample enterprise environment

The IBM ISDM product was used, consisting of four virtual software images, as shown in Figure 2-3 on page 19:

- IBM Tivoli® Service Automation Manager
- IBM Tivoli Monitoring
- IBM Tivoli Usage and Accounting Manager
- Network File System (NFS): File repository, URL re-direct, and a mail server

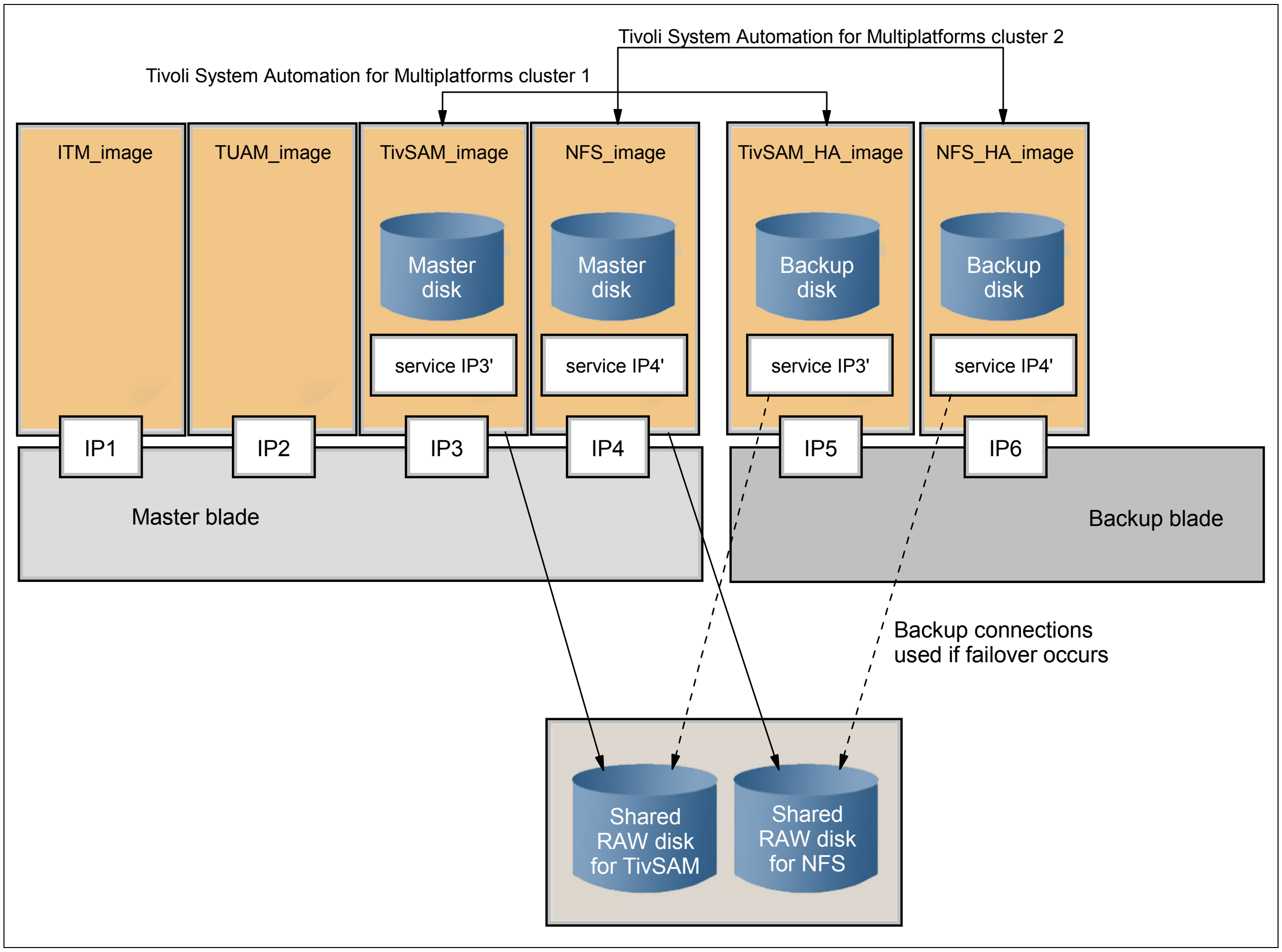


Figure 2-3 VIrtual software images

After the infrastructure was in place and the servers were autoprovisioned, each server was manually tested to ensure that is was created correctly. This was an important step because many errors were discovered and fixed before production. After this, all of the autoprovisioned servers were created the same way without errors. Performance objectives were set and met for the provisioning of a server. The other key component was the SA&D automation, where every step in the server build process is verified based on evidence gathered during each of the steps and confirmed. Any item that did not pass the SA&D automation was flagged for investigation.

The performance and capacity footprint for this cloud solution was already known because it existed in a non-cloud environment. Other cloud solutions must go through performance testing.

This example shows the need for performance planning and monitoring, which is covered in the remainder of this paper.

3

Performance testing for cloud computing

The cloud computing paradigm poses new challenges to performance management of both applications and infrastructure because of the additional complexity for dealing with quality of service (QoS) requirements in a highly virtualized dynamic infrastructure. This is particularly true for either SaaS applications in public clouds being deployed into production with stringent SLAs or critical internal applications built on PaaS/IaaS infrastructures in private clouds. Thus, performance testing activities are fundamental to reducing the risks associated with production deployment or to managing important changes in user and transaction workloads.

3.1 Introduction

Performance testing is typically a time-consuming and expensive activity, but it helps mitigate risks of going into production. Cloud environments add complexity to performance testing and therefore the costs can increase. However, because the risks of a traditional environment still exist, it is crucial that performance testing be properly designed and organized. Where a private cloud is concerned, responsibility is typically shared between internal application owners and the internal cloud infrastructure provider. In this case, explicit SLAs are not often defined. Nevertheless, when dealing with business critical applications and applications with high loads, it is strongly advisable that cloud-specific performance testing be jointly planned and executed.

Also in public clouds, when SaaS applications are offered to customers with SLAs, it is necessary to include cloud-specific performance testing along with other traditional testing activities. In this case, the cloud service provider must organize the activities because the provider has application ownership and is responsible for ensuring that SLAs can be met.

Generally speaking, cloud environments consist of a cloud management platform and a managed platform. It is advisable to consider specific performance testing scenarios for each:

- Testing of the cloud management platform
- Application performance testing on cloud infrastructure

In this section, we discuss performance testing scenarios for the cloud management platform and for the application that runs on the cloud. We discuss the responsibilities of the different stakeholders, such as service, consumer, service integrator, and service provider in each case and present some typical use cases.

3.2 Testing the cloud management platform

Testing the cloud management platform has the following goals:

- Identify cloud platform performance critical test cases (provisioning and deprovisioning)
- Simulate an asynchronous workload that stresses the cloud platform in relation to user request processing and authorization workflows, orchestration workflows (pool resources selection and reservation, topology nodes actions), low level provisioning tasks with interaction with platform managers (hypervisor, network devices manager, storage devices manager), network infrastructure services, monitoring services, and accounting services.

  That workload can be simulated using web navigation simulation tools (for example, IBM Rational® Performance Tester) because most provisioning use cases require access to a web portal.

3.2.1 Cloud management platform performance use cases

The cloud management platform includes platform management, cloud management, Monitoring, OSS, and BSS capabilities. Either a single application can provide all these features or separate applications can address specific features. See Figure 5-2 on page 39.

The management software performance is dependent on what kind of service model it is going to address (IaaS, PaaS, or SaaS).The response time, performance, and concurrency for provisioning, deprovisioning, resizing, and deleting virtual machines is dependent on what is in the virtual machine image. The disk size is the primary factor in those cases:

- IaaS The VM only has an OS
- PaaS The VM has an OS + middleware + application development platform
- SaaS The VM has an OS + middleware + application

Image management is also one of the primary factors in a cloud. Capturing an image of a running VM is a heavy task.

To provide insight into the ins and outs of performance testing the cloud management platform, we included a typical example of a provisioning use case, based on the following SLA.

An example of using an SLA for provisioning

To set up a system:

1. Go to the service catalog and make your selections from among a list similar to the following list:

   □ Linux servers

   □ Apache

   □ Web Logic

   □ Oracle RAC (two nodes)

2. From the time the catalog entry selection is made, the SLA calls for the server to be provisioned in a certain number of days. This includes the IBM Service Delivery Manager (ISDM) product auto-provisioning the servers. Strategic alignment and deployment (SA&D) automation validates that what was built passes through all the internal validation processes including evidence gathering.
3. Every step must be accounted and a report is produced that indicates whether it has passed through all the steps.

At the end of this, you have a server that is provisioned and includes the auto validation steps that can be reviewed. There are many interfaces during the provisioning process, such as asset registration, configuration management database (CMDB) updated with CI information, and many more.

3.3 Application performance testing on cloud infrastructure

In cloud environments, it is critical to avoid underestimating the impact of newly deployed applications on existing applications and that of existing applications on the new ones. In addition, difficulties with automatic or semi-automatic virtual resource upscaling and downscaling to accommodate workload changes (elasticity) can lead to performance issues if not properly tested in advance.

Application performance testing on virtualized infrastructure has the following goals:

- Validate the overall performance of the target infrastructure for hosting applications
- Evaluate the effects of applications running simultaneously on the same infrastructure
- Identify bottlenecks of critical resources of the virtualized infrastructure, such as hot spots, unbalancing, hypervisor overload, resource overcommitment

- Validate Workload Management Policies

The SLAs and use cases depend a lot on the application, so the usual processes have to be followed. For each type of testing, document (1) test goals (2) test requirements (SLAs) (3) test cases, and (4) your test approach.

The main reason to adopt a cloud is the amount of business agility achievable because of the availability of resources dynamically. If an application can run better with twice the resources with less cost in a cloud when compared with the resources in the existing on-premise infrastructure, spending time on application performance and optimal tuning to run with current resources will not be meaningful.

We need to correlate application performance with respect to IaaS, PaaS, and SaaS. The performance for an application in a cloud means mostly for the production workload. Usually an application in a cloud refers to SaaS. Of course all applications cannot be SaaS. SaaS is single-tenancy or multi-tenancy.The application has to be designed and developed specifically to address multi-tenancy. The performance testing also has to be done with respect to single-tenancy or multi-tenancy. There are different approaches to achieve multi-tenancy.

Also in a cloud, we can get dedicated CPU and memory resources. In this case, you get the same performance. But storage and network are mostly shared, and we need to test it appropriately. Usually in a private cloud, we do not see performance issues, but there are many chances in a public cloud for performance issues.

The black box, grey box, and white box approaches are more focused on the perspective of the server, cloud service provider, or SaaS provider. Ideally in a cloud, the users do not know where their virtual machines are running. They can be on the same server or distributed across servers in the cloud. If in a private cloud the user can request the same server for all their VM requirements, that is in conflict with the purpose of a cloud.

Also the cloud service provider cannot do performance testing for customer applications to provide servers. The user must benchmark their application properly to identify the resources required and then purchase VMs from the cloud service provider (either private or public).

3.4 Process for application performance testing on cloud infrastructure

Standard testing processes focus on application and infrastructure components tested in an isolated environment, generally without taking into account mutual interference between applications and workloads. Performance testing normally proceeds in a phased way with the following test types:

Single transaction testing In single transaction testing, the focus is on the performance of one specific business transaction or use case with the objective to baseline the performance of this business transaction and, if needed, to optimize it in isolation. Testing is usually done starting from a small number of virtual users and gradually increasing this number to establish an insight in the behavior of the transaction under load.

Mixed workload volume testing In mixed workload volume testing, the focus is on the performance of the system when loaded with realistic peak production volumes. Growth scenarios can be tested by gradually increasing the number of virtual users.

Mixed workload stress testing Mixed workload stress testing is similar to mixed workload volume testing. The difference is that the focus in this case is on trying to bring the system to capacity overload. This is what most performance test teams seem to like best, although it is not always the most important.

Mixed workload duration testing

In mixed workload duration testing, the focus is on the duration of the test (preferably including a date change) not the volume of virtual users. This can be a steady average number.

For the mixed workload tests, the workload must be "production equivalent". There can be agreed growth scenarios that are expected to happen in (for example) five years.

We recommend that all the performance test types for application performance testing be executed in cloud environments and in non-cloud environments. In cloud environments, however, include two new test types in the overall test process:

- Workload cross-impact test
- Application elasticity test

Workload cross-impact testing simulates the sum of two concurrent workloads, as shown in Figure 3-1:

- Expected application workload: The workload generated with standard performance testing processes and tools
- Existing cloud workload: A sort of background noise that emulates the load due to other applications sharing the same physical resources.

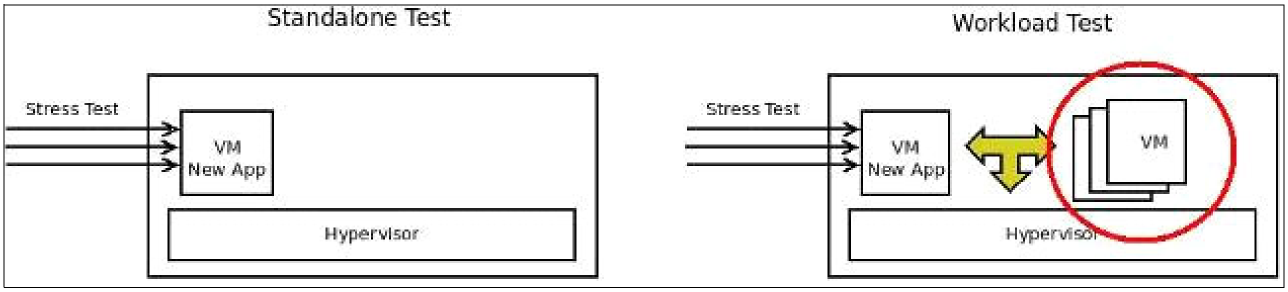


Figure 3-1 Workload cross-impact test

Application elasticity testing aims to evaluate the impact that automatic upscaling and downscaling has on the new application and on the entire cloud infrastructure. This performance test extends the workload cross-impact test with an additional step: the standard application load is varied to activate elasticity features. The objective of that test is to observe how smoothly the application reacts to the variation of underlying virtual resources.

The performance testing process in a cloud context extends the standard performance test process with cloud scenarios tests. Applications must be tested first in stand-alone mode, as usual, using standard tools. The test results can be considered as a baseline for the next analysis. Next, the applications undergo cloud-specific testing. Both workload cross-impact and application elasticity testing can be based on, and combined with, mixed workload volume testing.

Workload cross-impact tests require simulating the workload of existing applications that share the same physical resources in addition to that produced by the application on focus. The testing must also address the increase of the workload from a minimal amount to the peak loads expected to assess and identify the speed of elasticity of the cloud (how quickly the capacity comes online and what the impact is on performance of transactions based on the workload). The scope of this testing can be either a gradual increase of the load, implying the capacity comes on line gracefully or (in the case of a load from a concert sale), the load might increase from nothing to peak load in seconds and capacity can increase from one to many resources in a short period. In both scenarios, the user experience and performance are impacted and dependent on the ability of the cloud service provider to scale the capacity. The user experience can be either timeouts (cannot access) to slow response time, and the acceptable behavior depends on the system that is being implemented.

In general, there are three different approaches to simulation with increasing complexity and accuracy: black box, grey box and white box. We discuss these approaches beginning in 3.4.1. Choosing the best approach depends on the objectives of performance testing and the trade-off between testing costs and potential risk reduction. Pragmatically speaking, some hybrid approaches might be possible.

All approaches use the concept of background noise virtual machines (VMs). These different kinds of workload generators are installed and configured based on chosen approach.

Figure 3-2 illustrates the three approaches.

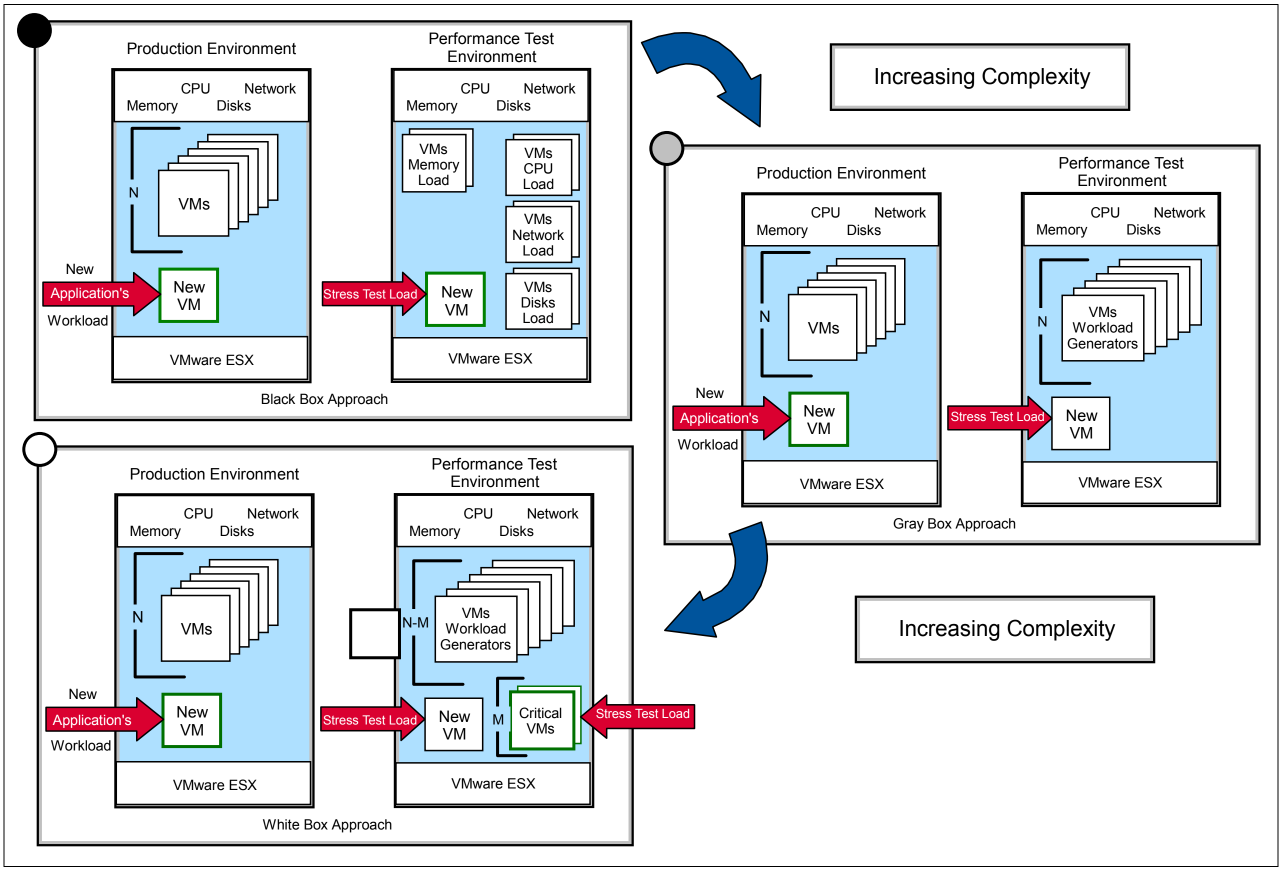


Figure 3-2   Cloud workload generation approaches

3.4.1 Black box approach

In the black box approach, background workloads are modeled only on the basis of the physical aggregated resource utilization patterns for CPU, memory, I/O channels, and network, measured at the hypervisor level. No specifics of programs running on each virtual machine are considered.

Workload is simulated through software (workload generators) in the background noise VMs that generate the expected workload in terms of primary physical resources consuming CPU, memory, disk I/O, and network. This is achieved using hardware benchmark software (SPECINT-like).

3.4.2 Grey box approach

In the grey box approach, workload generators are configured to generate resource utilization patterns of all other existing applications running on the same physical system. For modeling, the following information must be gathered for each existing application:

- Hypervisor configuration data and resources usage profile
- Resources utilization patterns: CPU patterns (multiprocessing level)
- Processor cache locality, memory pattern (scattered), I/O patterns (sequential or random read/write), network pattern (streaming, chatty)
- Performance scalability limit: CPU bound, Memory bound, Disk I/O bound, Network bound
- Workload category: Web, SAP, BI, OLTP, HPC, Filesharing, and so on

The workload generators used in the grey box approach are advanced versions of those used in the black box approach because they emulate workloads using resource patterns specific to application categories. This is achieved using specialized benchmark software for the workload category.

3.4.3 White box approach

In this case the workload generators are the real applications. In the test environment, it will be necessary to generate the stress test load for not only the application under focus but also the other applications sharing the same physical resources. This approach is highly expensive in terms of time and resources and is not considered feasible in most cases.

A hybrid approach with the simulation of only one or two critical existing applications can be evaluated. In this case, the overall existing workload is created in part using the workload generators proposed for the black box or grey box approach and in part using a standard load generation from critical applications.

Finally, to reduce the complexity in simulating an existing critical application without dependencies on real back-ends, consider an application virtualization. Several products for back-end simulation are available on the market (IBM Green Hat).

3.5 Determine the responsibilities of who does what

The provider/consumer perspective and the service depth being delivered using the cloud determine the needs for testing, monitoring, and capacity planning. The RACI matrixes in

Tables 3-1 and 3-2 show the responsibilities of the different stakeholders in the different situations. The meaning of the abbreviations in the tables are:

R	Responsible
A	Accountable
C	Consulted
I	Informed

The assumptions underlying the two tables are:

- A consumer can be a mature/corporate consumer, such as a bank or insurance company, with their own IT department. A consumer can also be a retail consumer. In case of a mature/corporate consumer, one can assume that the consumer might have facilities for testing and monitoring. This is highly unlikely with a retail consumer. In the case of a mature/corporate consumer, one can also assume that an SLA was agreed with the integrator. In the case of a retail consumer, there is probably some sort of a default (assumed) SLA, not much more than nickel or bronze.
- An integrator does not necessarily have white box access to all the layers of the service that they provide. They can work with one or more providers with whom they have to have SLAs.
- A provider has white box access to the service that they provide.
- The integrator is responsible for delivering the capabilities that they promised per SLA. They must benchmark their offering using generic workload types, depending on the service depth. The consumer is responsible to test and monitor the specific workload that they generate.
- For retail consumers there are mechanisms, such as throttling and admission control to prevent them from taking up too much capacity. Describe this information in the terms and conditions section.

Building on these assumptions, a corporate customer who uses an Interface-as-a-Service (IaaS) offering is responsible for performance testing and monitoring all layers on top of the infrastructure. If the customer uses a Platform-as-a-Service (PaaS) offering, a Software-as-a-Service (SaaS) offering, or a Business-Process-as-a-Service (BPaaS) offering, these responsibilities are more and more offloaded to the integrator and providers as the offering moves to a higher level in the solution stack. As far as the scope of testing and monitoring is concerned, only the provider of a service has white box access to the service that they are monitoring or testing, as shown in Table 3-1.

Note: The RACI tables apply to performance testing of the application stack that runs on the cloud. Performance testing of the cloud management platform is entirely the responsibility of the provider.

Table 3-1 Responsibilities for cloud performance testing and monitoring with corporate customer

Service depth	Testing and monitoring activity	Consumer (corporate)	Integrator	Provider
	Performance testing and monitoring			
IaaS	System performance testing and monitoring		RA (black box)	R (white box; generic
	Middleware performance testing and monitoring	RA (specific)	CI	CI
	Application performance testing and monitoring	RA (specific)	CI	CI
	Business process performance testing and monitoring	RA (specific)	CI	CI

Service depth	Testing and monitoring activity	Consumer (corporate)	Integrator	Provider
PaaS	System performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Middleware performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Application performance testing and monitoring	RA (specific)	CI	CI
	Business process performance testing and monitoring	RA (specific)	CI	CI
SaaS	System performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Middleware performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Application performance testing and monitoring	RA (specific)	RA (black box; generic)	R (white box; generic
	Business process performance testing and monitoring	RA (specific)	CI	CI
BPaaS	System performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Middleware performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Application performance testing and monitoring	RA (specific)	RA (black box; generic)	R (white box; generic
	Business process performance testing and monitoring	RA (specific)	RA (black box; generic)	R (white box; generic

A retail customer, who is using a public cloud, is unlikely to do any testing or monitoring, as reflected inTable 3-2.

Table 3-2 Responsibilities for cloud performance testing and monitoring with retail customer

Service depth	Testing and monitoring activity	Consumer (corporate)	Integrator	Provider
	Performance testing and monitoring			
IaaS	System performance testing and monitoring		RA (black box)	R (white box; generic
	Middleware performance testing and monitoring	N/A		
	Application performance testing and monitoring	N/A		
	Business process performance testing and monitoring	N/A		
PaaS	System performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Middleware performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Application performance testing and monitoring	N/A		
	Business process performance testing and monitoring	N/A		

Service depth	Testing and monitoring activity	Consumer (corporate)	Integrator	Provider
SaaS	System performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Middleware performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Application performance testing and monitoring	N/A	RA (black box; generic)	R (white box; generic
	Business process performance testing and monitoring	N/A		
BPaaS	System performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Middleware performance testing and monitoring		RA (black box; generic)	R (white box; generic
	Application performance testing and monitoring	N/A	RA (black box; generic)	R (white box; generic
	Business process performance testing and monitoring	N/A	RA (black box; generic)	R (white box; generic

4

Monitoring for best practices in cloud computing

Monitoring is a key component in cloud computing performance engineering and post production activities. Consistent monitoring of the solution with key monitored parameters during performance testing helps detect performance bottlenecks that can be resolved by tuning the solution. Monitoring provides deep insights about the behavior of cloud solution under real time usage by consumers and under heavy loads. This also helps in planning the capacity of the system based on the load volumes. See Appendix C, “IBM Tivoli tools” on page 61.

4.1 General monitoring concepts

The objective of monitoring is to identify and address issues before consumers of the cloud are affected. This is similar to the monitor objective of a traditional hosting model. Additional monitoring triggers are autonomic actions that can be taken when a certain threshold is reached to prevent issues, for example, start additional LPARs and JVMs as load increases.

Monitoring must have dashboards/views for the cloud service provider and its consumers. Providers will be interested in metrics that will have an impact on its ability to meet SLA, such as CPU, response time and availability. Consumers are interested in business metrics, such as the number of requests completed and total completion time. Private and Hybrid cloud consumers might also be interested in resource metrics to gauge if it needs to increase its investment in cloud capacity and its external cloud vendors service qualities[1].

- Public considerations:
  - SLA tracking—KPIs that allow monitoring of SLA compliance, such as service availability and response time.
  - KPI that measures the current utilization of the cloud resources, such as CPU/Memory and IO bandwidth.
  - Differentiate any runtime (real-time) versus batch jobs or synchronous versus asynchronous activities.
- Private considerations:
  - In addition to public monitoring, also have SLA monitoring for each external service provider.
- Hybrid:
  - Combination of public and private. Specifically for components serviced by a public cloud to ensure any negative performance and availability will not affect the application.

4.2 Effective monitoring practices

The following points will help you effectively monitor best practices:

- Know your customer's KPIs and SLAs to determine the cause of application crashes:
  - Understand the solution's performance attributes. The cloud computing solution uses multiple middleware components to carry out the core processing of computing workloads. Monitoring the middleware closely gives you a good indication of the overall performance of the solution.
  - Common KPIs are transactional throughput and response time of transactions, and so on.
- Observe the monitor and know how much the monitoring tool consumes:
  - Do not run heavy monitoring tools or agents on the OS where the overall resource OS utilization is 65 - 70%. This can cause server utilization to exceed the limit and lead to poor performance.

[1] For a fuller discussion of the attributes for cloud and the metrics for an SLA driven or differentiated cloud, see: http://www.ibm.com/developerworks/cloud/library/cl-rev2sla.html?ca=drs-

  - Minimal monitoring is recommended. Monitor only the most significant parameters during solution usage unless you encounter performance issues where deeper monitoring is required to understand the solution behavior.
- Use historical data and avoid online monitoring for analysis purpose:
  - We recommend that you use historical data, gathered over time, to analyze the solution behavior. This ensures that precise data values are obtained and that accurate inferences are made on performance characteristics of the solution.
- Use standard monitoring tools that provide automatic triggers whenever resource utilization exceeds a limit:
  - Such tools will help you accurately monitor the solution around the clock and also prevent outages by raising the required alarms on time.
- Understand negative performance and high availability during monitoring.
- Use tools that consolidate all monitoring results on one window for easier use:
  - ITM and ITCAM monitoring tools provide an integrated view of monitoring all servers in the same window, which eases monitoring multiple servers in a cloud.

We see that monitoring a variety of OS and solution component metrics helps you to understand component status in the cloud computing ecosystem, both from a current and a historical perspective. As you add back end resources to the system, you need to only instruct the monitoring tool to collect additional metrics. Using the best practices we discussed, monitoring the cloud keeps a constant check on your system and ensures that resources are effectively utilized.

Measuring SLA as experienced by the end user is essential for applications, such as discount brokerages where the speed of executing a trade is of essence. Synthetic transactions are selected based on transactions maps (unique paths through the system) for availability and response times measurements. This allows for pro-active monitoring. SLA availability can be established from an independent source that actually emulated the user experience.

4.2.1 Performance and capacity planning

Traditionally capacity can be augmented by scaling up or scaling out. Scaling up means adding resources internal to the servers while scaling out is adding more servers. Extra resources added to an existing application or service is only beneficial if they can be utilized. For example, a single threaded design (such as point-to-point video service) might not benefit while a card credit card authorization, based on a loosely coupled distributed system or SOA-based system, might benefit by the management tool auto-adding more servers.

Applications going into the cloud must have linear or near-linear scalability with resource requirements increasing in steady proportion to the workload. This allows the virtualized resources to be added either automatically, using WLM-type event monitoring, or semi-automatically, based on additional VMs or LPARs when needed to support both short-term peaks and long-term growth.

You can still end up with performance bottlenecks even though you have all the capacity you need. This is often due to tuning option settings at the component level. For these cases, you must establish baseline settings, monitor for specific thresholds, and update. (This is part of the white box analysis required as part of Performance Engineering when doing performance testing).

4.3 Differentiated services

At this point, we can consider the internet as helpful. In the early stages of the internet, transport mechanisms were characterized by best effort. As internet use matured and the need for services became specialized, another mechanism was introduced, differentiated quality of services (QoS), a transport tailored to service needs with guaranteed service levels (SLAs) with measurable metrics.

In a similar sense, we need to move away from a simple vanilla flavor cloud scenario to envision a differentiated QoS cloud. The gold, silver, and bronze service levels within the same cloud will provide differentiated services as defined by service catalogued SLA differentiation.

A service level agreement is a contract containing service level objectives (SLOs) and potential credits if the SLA is not met. The SLOs are constructed from individual performance metrics, such as latency of network. The SLA and any potential credit is defined by this SLA as an aggregated service.

In the cloud model, any request for services needs to be checked and filtered to ensure the capacity is available to actually deliver the required service. If demand is unchecked, the resource layer cannot deliver the required service.

5

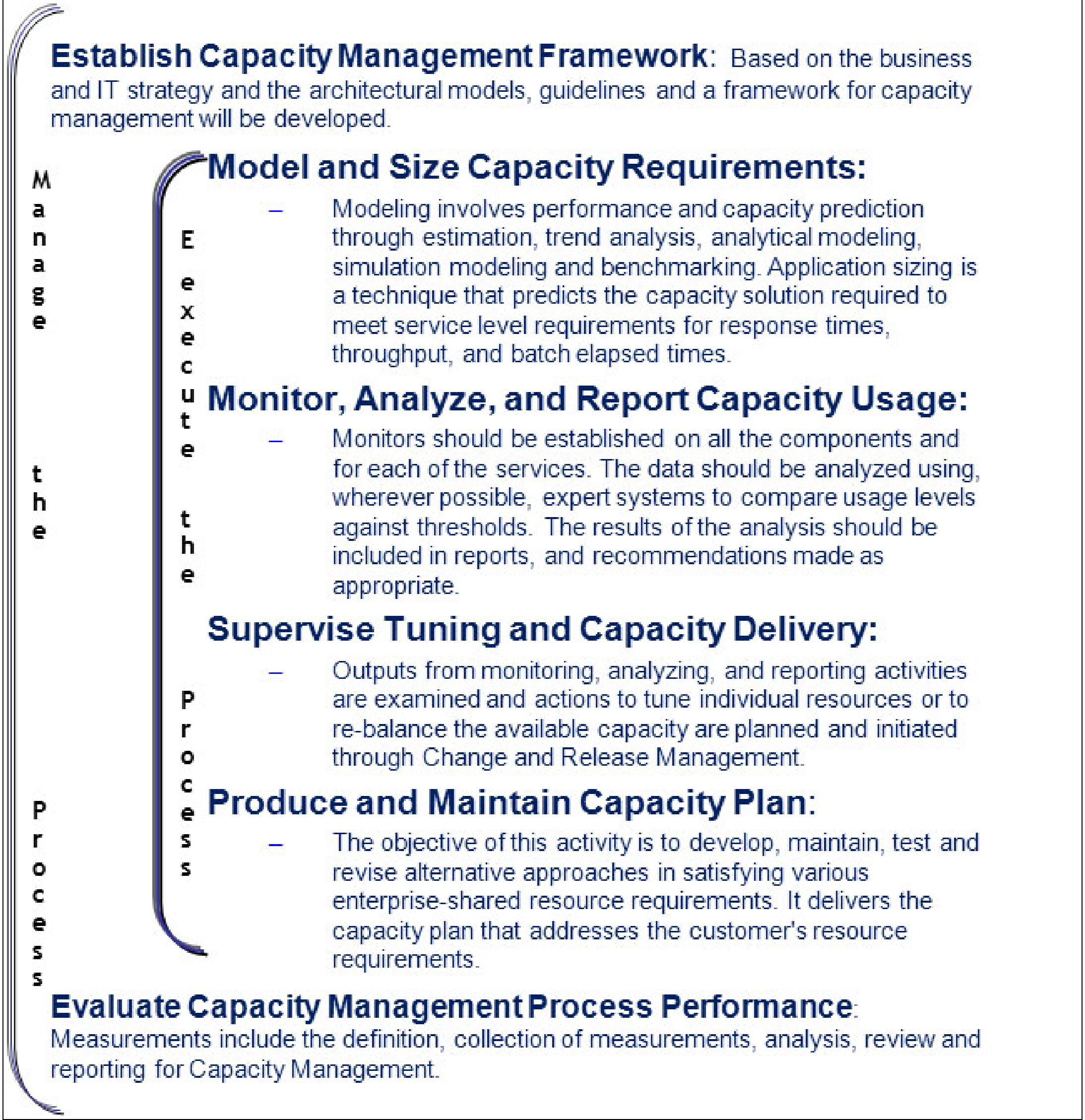
Capacity management for cloud computing

Capacity Management is one of the processes included in ITIL process framework.

The Capacity Management process is consists of six activities: two that manage the process and four that execute the process. The process is further described in Appendix B, “IBM Tivoli Unified Process (ITUP)” on page 59. A list of tools is included in Appendix C, “IBM Tivoli tools” on page 61.

5.1 Overview

The Capacity Management process is not an isolated, stand-alone process. Capacity Management has interdependencies with several other IT Service Management processes and some are even more important in the context of cloud computing. For example, Capacity Management has a strong dependence on well defined and managed Asset and Configuration management to provide and track the number, size, relationship, and location of the cloud components that are in scope for any given cloud computing environment. See Figure 5-1.



Establish Capacity Management Framework: Based on the business and IT strategy and the architectural models, guidelines and a framework for capacity management will be developed.

Model and Size Capacity Requirements:

– Modeling involves performance and capacity prediction through estimation, trend analysis, analytical modeling, simulation modeling and benchmarking. Application sizing is a technique that predicts the capacity solution required to meet service level requirements for response times, throughput, and batch elapsed times.

Monitor, Analyze, and Report Capacity Usage:

– Monitors should be established on all the components and for each of the services. The data should be analyzed using, wherever possible, expert systems to compare usage levels against thresholds. The results of the analysis should be included in reports, and recommendations made as appropriate.

Supervise Tuning and Capacity Delivery:

– Outputs from monitoring, analyzing, and reporting activities are examined and actions to tune individual resources or to re-balance the available capacity are planned and initiated through Change and Release Management.

Produce and Maintain Capacity Plan:

– The objective of this activity is to develop, maintain, test and revise alternative approaches in satisfying various enterprise-shared resource requirements. It delivers the capacity plan that addresses the customer's resource requirements.

Evaluate Capacity Management Process Performance: Measurements include the definition, collection of measurements, analysis, review and reporting for Capacity Management.

Figure 5-1 Capacity planning process

Capacity Management can drive the monitoring and measurement requirements useful to other IT Service Management processes, such as Event Management and Availability Management. IT resource utilization metrics used for capacity planning can also be used for usage-based cost recovery and billing. Metrics requirements details, such as a unified time base and relationship metadata, become critical to capacity management of cloud environment since many metrics and measurements from different tools and sources need to have a consistent time base, and a relationship method between data items to allow analytics and predictions to be performed. Automated data collection, summarization, pruning, archiving, and storage is essential. Reliability and integrity of the data elements and data management controls are essential to ensure consistent, repeatable, reliable metrics.

Elements of Demand Management are the main source of business demand that Capacity Management uses to develop capacity forecasts and solutions. Capacity Management translates business demand through to the infrastructure component level. A common, standard set of resource usage data can be shared by performance, capacity, and usage-based show-back/charge-back to forecast and manage IT demand.

These examples of interdependencies between Capacity Management and other IT Service Management processes are directly relevant to cloud services.

The process covers understanding service requirements, determining component capacities, and designing and deploying capacity to meet expectations. It collects and analyzes data relevant to infrastructure utilization and performance to determine whether there are potential problems and issues that need to be addressed.

ITIL defines the following three focus areas, or sub-processes, which are addressed by Capacity Management. Each uses the primary activities of the process decomposition in differing ways, to differing end results.

- Business Capacity Management (BCM): This focus area is responsible for ensuring that the impacts of future business requirements for IT services upon IT resources are considered, planned, and implemented in a timely fashion (requirements (NFRs), drivers, volumetrics, use cases).
- Service Capacity Management (SCM): This focus area is the management of the performance of the IT services used by the customers. It is responsible for ensuring that the service performance is monitored, measured, reported, and meets business requirements and agreements (workload by service/application/transaction and corresponding sw stack).
- Component Capacity Management (CCM): This focus area is the management of the performance, utilization, and capacity of individual technical components possessing finite resources (IT resource load/consumption on the server, such as CPU, I/O, memory, storage, network infrastructure, and other hardware elements).

When applying Capacity Management to cloud computing we explore who is responsible for the activities, inputs, outputs, and controls. This must be fully examined and understood in terms of roles and responsibilities, depending on the provider/consumer perspective and the service depth.

The two activities in Figure 5.1 are not examined in this paper because these are fairly standard process design and management activities. They are not significantly different for cloud computing, other than to underscore the need to clarify the objectives, scope, and roles and responsibilities for this process. They must be negotiated and agreed to between the various business roles and layers of service.

Capacity Planning involves tasks imbedded within each of the four activities in Figure 5-1 on page 36. The next section of this paper summarizes those four activities and the elements that are relevant to cloud computing.

5.2 Model and size cloud capacity

The modeling and sizing activity is generally iterative through the lifecycle of a development project. Estimates and models will almost certainly need revising as more detail and changed input information becomes available during the development of a system. In capacity planning of cloud, there are various stakeholders with their own responsibilities involved, based on their role and based on the cloud service model.

There are many aspects of the capacity planning approach that are common to cloud and non-cloud environments; however, there are technical factors unique to cloud that need to be considered to efficiently utilize a virtualized and distributed infrastructure with automated provisioning and the need to align with particular business models for the type of cloud services. Optimization of a virtual environment across resource pools is crucial to capacity planning for cloud computing.

5.2.1 Capacity planning business objectives

The capacity planning must align to the business objectives of the cloud consumer and the cloud provider. The consumer can move to cloud to achieve more agility and the provider must have optimal cloud infrastructure to serve individual consumer workload requirements:

- Long term versus short term business objective (total cost of ownership (TC) of a private versus a public cloud). The cost associated with Infrastructure, middleware, and software in a private cloud can be less than the cost associated with a public cloud in the long term.
- What kind of servers are required for different application workloads to run in the cloud? The cloud can be built with commodity servers and high-end servers with support for different platforms to support different application requirements from various consumers.
- How elastic or dynamic is your cloud? Can provisioning be done instantly for all requests within the organization or must it be prioritized based on the availability?
- Reduce costs by reducing the number of physical servers and Lowers Energy consumption.
- Achieve business agility for the existing business. Is my existing infrastructure enough for cloud?
- Achieve business agility for the new business. What rate will the business requirements grow?
- What options are available now? Private/public/hybrid clouds? What service model (IaaS, PaaS, SaaS, BPaaS) best suits consumer business requirements?
- Who takes ownership in an end-end capacity planning for cloud?
- Who is going to manage above hypervisor and below hypervisor monitoring and capacity planning? After the instance is provisioned to the user in the public cloud, the user has the responsibility to manage his application on the instance and effectively utilize the instance. If an organization plans to adopt a public cloud, capacity planning has to be done to efficiently manage all their virtual machines and servers in the cloud.
- Planning Disaster Recovery

Figure 5-1 on page 36 shows capacity planning for a cloud journey.

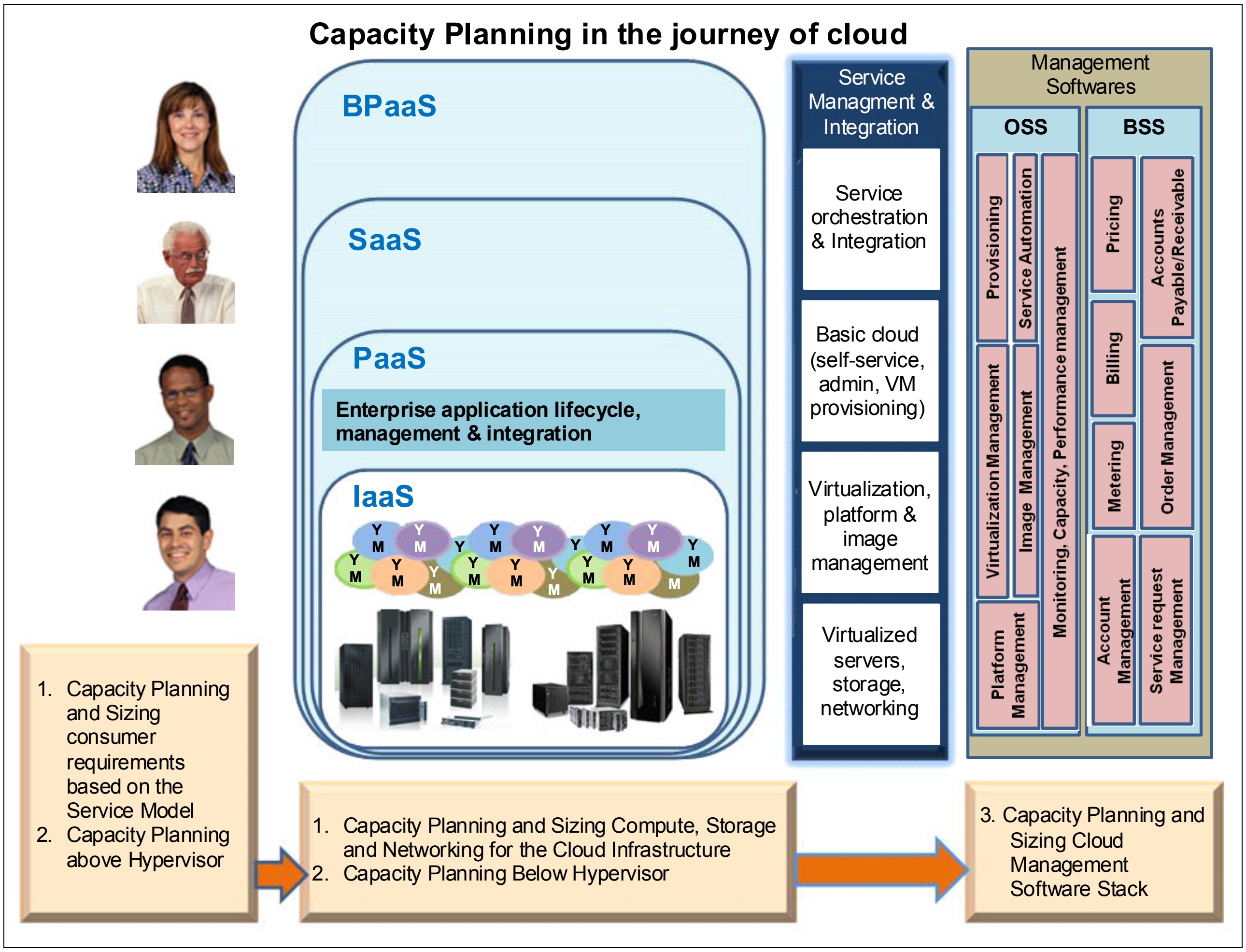

Capacity Planning in the journey of cloud
BPaaS
SaaS
PaaS
Enterprise application lifecycle, management & integration
IaaS
Service Managment & Integration
Service orchestration & Integration
Basic cloud (self-service, admin, VM provisioning)
Virtualization, platform & image management
Virtualized servers, storage, networking
Management Softwares
OSS
BSS
Provisioning
Service Automation
Virtualization Management
Image Management
Monitoring, Capacity, Performance management
Platform Management
Pricing
Accounts Payable/Receivable
Billing
Metering
Order Management
Account Management
Service request Management
1. Capacity Planning and Sizing consumer requirements based on the Service Model
2. Capacity Planning above Hypervisor
1. Capacity Planning and Sizing Compute, Storage and Networking for the Cloud Infrastructure
2. Capacity Planning Below Hypervisor
3. Capacity Planning and Sizing Cloud Management Software Stack

Figure 5-2 Capacity planning for a cloud journey

5.2.2 Capacity modeling

Capacity modeling involves categorization and characterization of entities involved in the end-to-end cloud infrastructure. There are many factors that need to be correlated to come up with the correct estimation model to provide optimal service to the consumers. Historical and live data need to be analyzed to forecast future requirements and to identify patterns to service different workloads efficiently. This is applicable for both the cloud consumer and provider:

1. Identify the tenancy model (single tenancy, multi-tenancy model).
2. Identify the performance characterization and volumetrics of the application with respect to tenancy model.
3. Identify the workload deployment type (development, test, and production).
4. Identify the correct servers and platform for your application requirements.
5. Analyze the CPU, memory, storage, and network requirements (minimum, average, and maximum).
6. Consider I/O usage versus CPU usage versus memory usage versus virtualization processing.
7. Consider the resource sharing requirements (dedicated or shared).

8. Identify the SLA requirements for production environments (response time range and throughput range).

5.2.3 Cloud consumer

The following additional factors need to be considered by the cloud consumer and cloud provider during the capacity planning cycle to create the correct estimation model:

1. It is the responsibility of the consumers to size their applications and request the correct resources in the cloud based on the service models (PaaS, SaaS and BPaaS).
2. Consumers need to plan and manage the virtual machine instances purchased (above the hypervisor).
3. Size the application and then the purchase instance or the purchase instance in the cloud based on rough estimate and then size or effectively utilize the infrastructure. It is a reiterative process.
4. How much business impact exists for oversized virtual machines in the cloud? Can a consumer do oversizing in a private cloud?

5.2.4 Cloud provider

The following additional factors need to be considered by the cloud consumer and cloud provider during the capacity planning cycle to create the correct estimation model:

1. Choose the service models (IaaS, PaaS, SaaS and BPaaS) offered through the cloud. The IaaS consumers are more concerned about capacity planning and performance of their virtual machines in the cloud, and it is better to have separate resource pools to address IaaS alone. The consumers for SaaS, PaaS, and BPaaS do not need to take care of capacity planning because it is managed by the cloud provider, which can be delivered on any platform. The cloud provider can have separate resource pools to address these services.
2. Identify the target customers, and choose the servers and platforms to support their requirements. The number of consumers is minimal during the start of their cloud business, and it will grow over time and might get steady after some time.
3. Consolidate all cloud consumer virtual server resource requirements regularly. Policy-based sizing and placement optimization of the consumer VMs on shared provider resources is key to reduce the cost of running the shared infrastructure, maintaining manageable risk levels, and planning ahead for demand growth. Policies dictate how VMs can be sized and placed for business and technical reasons. Business policies are typically hard constraints while technical policies are soft constraints or best practices.
4. Have enough infrastructure to serve all customers' dynamic requirements.
5. The objective is to achieve a higher average utilization and higher consolidation ratio. If the consumer virtual machines are sharing resources among them, the consolidation ratio is important to achieve higher average utilization. This is not applicable for the virtual machine's required dedicated resources where resources are reserved and utilization depends on consumer workload requirements. The resource sharing is common in private clouds and in development and test environments and higher average utilization will result more savings.
6. Define thresholds for the servers in the infrastructure for variable workload. The overhead caused by different server platforms and hypervisors will not be same. The thresholds and headroom must be maintained to handle overheads and yield better performance to the virtual machines running on the server. Typically, capacity planning tools provide technical policies to maintain a certain resource headroom on physical servers, given the

hypervisor, number of cores, and even the application or middleware running on the platform.

7. Plan and manage the cloud infrastructure (below hypervisor). The cloud infrastructure designed to support dynamically adding and removing compute, storage, and networking resources based on the consumer demand.
8. Monitor the customer virtual machines, application performance, and SLAs. Identify patterns and workload migrations across servers in the cloud.
9. Define workload placement models.
10. Plan for Disaster Recovery Infrastructure. The cloud environment must have a Disaster Recovery plan based on the consumer requirements. The Recovery Point Objective and Recovery Time Objective need to be defined for the DR solutions.
11. Have a charge back model to effectively apply pricing for the consumer.

5.2.5 Planning for a higher consolidation ratio

The cloud infrastructure needs to be planned effectively to reduce space, power, and cost requirements. This is important mainly for private cloud environments where cost is the primary concern. Also when the servers are running in shared mode, we need to achieve a higher consolidation ratio. The workloads for development and testing environments do not require dedicated virtual machines, and response time SLA is not the primary objective. There are many approaches available to calculate the consolidation ratio of the servers. One among them is statistical modeling.

Statistical modeling for workload variability and consolidation ratio

Statistical modeling is one of the approaches that can be used to help plan the capacity solution for a cloud infrastructure. An example of when statistical modeling is helpful to compute the workload required is for a Cloud provider, where there are multiple tenants. This cloud infrastructure solution needs to be optimized and planned effectively to minimize space, power, and cost requirements. This is also important for private cloud environments where cost is a primary concern.

When servers in the cloud environment are running in shared mode, we need to strive for a higher consolidation ratio. The consolidation ratio of a server, which is typically used in server consolidation studies, can also be applied in capacity planning for Cloud. The workloads for development and testing environments typically will not require dedicated virtual machines, and response time SLAs are not the primary objective for these workloads. The consolidation ratio for different or similar workloads is an important factor to consider when choosing the correct server for a cloud infrastructure and to effectively utilize the server for different kinds of workloads. A higher consolidation ratio will result in higher cost savings.

There are many approaches to calculate consolidation ratio of the servers, and one among them is statistical modeling. The following formula is a simple way to calculate the server capacity.

```
Mean utilization = mean demand / capacity
```

The maximum number of workloads that can be consolidated on a target server is dependent on how the workloads are grouped together. The workload variability impacts the consolidation ratio. When you combine workloads with variable demand, the overall variability of the combined workloads gets smaller. Thus, larger machines capable of accommodating more workloads can run at higher utilization levels than smaller ones.

If we were to track average utilization over time of workloads with varying demand, the results tend to exhibit a bell curve type distribution pattern, as shown in Figure 5-3. Theory tells us that 95% of all values fall within two standard deviations from the mean. Thus, if we had a Service Level Agreement (SLA) that called for us to be able to handle 95% of all workloads that occur on the system over a given time period, we must size a machine with a capacity of the mean + two times the standard deviation (sigma).

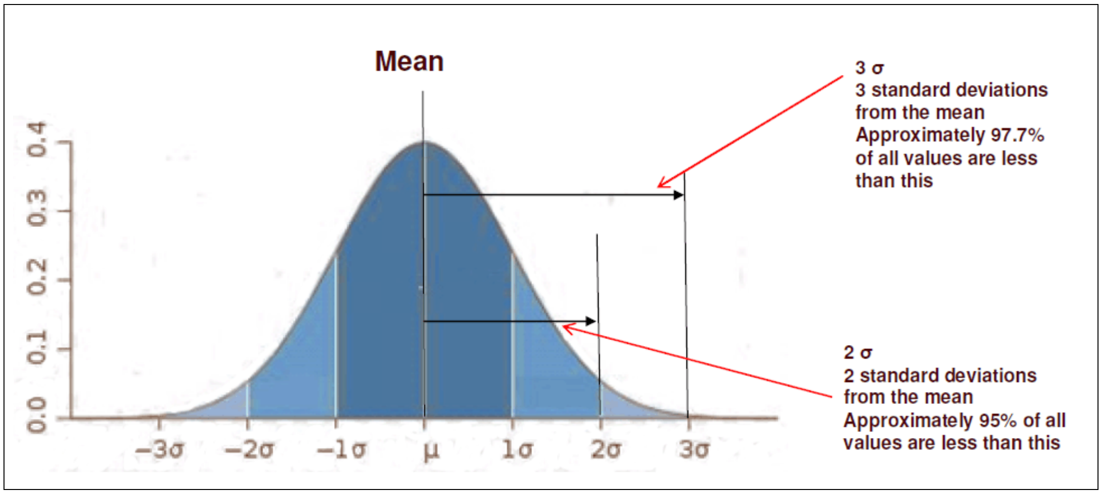


Figure 5-3 Mean

Combining similar workloads can result in a simple way to consolidate smaller servers. The peaks and valleys with the same pattern will result in high-average utilization without affecting SLAs. Combining variable workloads into a larger server results in a higher consolidation ratio. More headroom in large servers will help normalize variable workloads and achieve a higher consolidation ratio without affecting SLA. See Figure 5-4 on page 43. Combining lower priority workloads with higher priority workloads will provide a better higher average utilization. The lower priority workloads can sacrifice resources to fulfill high-priority workloads.

Be cautious when sharing the infrastructure allocated for multi-tenancy applications, and have enough headroom to accommodate variable load from different customers.

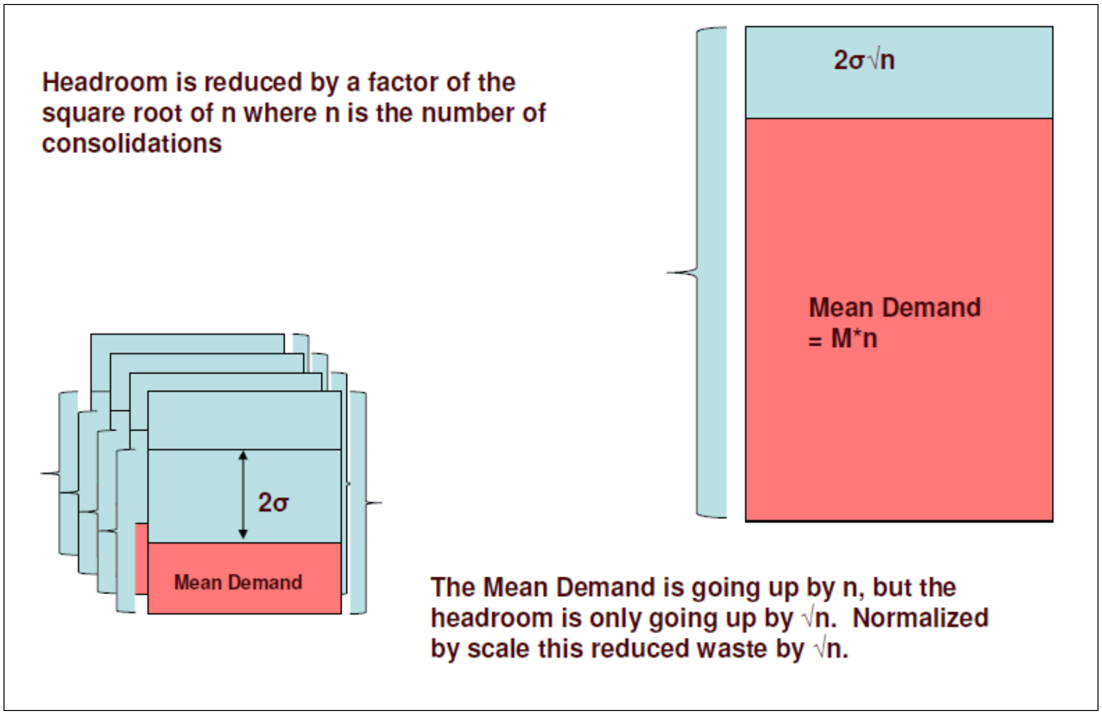

Headroom is reduced by a factor of the square root of n where n is the number of consolidations
2σ
Mean Demand
2σ√n
Mean Demand = M*n
The Mean Demand is going up by n, but the headroom is only going up by √n. Normalized by scale this reduced waste by √n.

Figure 5-4 Mean part 2

5.2.6 Single / multi-tenancy and service depth

Multi-tenancy provides the ability to the consumer to operate in isolation while sharing the cloud resources. Multi-tenancy is the primary factor in cloud to achieve more scalability and cost savings, compared to a single tenancy model. The multi-tenancy model is explained in Table 5-1 on page 44 for each service depth, and cloud solutions need to be designed as multi-tenant. It reduces duplication and results in reduction in capacity requirements for cloud infrastructure and software. The single tenancy model is mostly used in private clouds while multi-tenancy is adopted in public clouds to achieve more scalability.

The cost per consumer will decrease as sharing of infrastructure, software, and application increases. However the availability risk is high because the infrastructure is shared, if any outages in the infrastructure impact all consumers sharing the same services. The multi-tenancy cloud solution has to be designed appropriately to mitigate risks and provide high availability, scalability, optimal performance, and data protection to all consumers. See Table 5-1 on page 44.

Table 5-1 Single / multi-tenancy and service depth

Service Depth	Single Tenancy	Multi-Tenancy
IaaS	Each consumer will have dedicated virtual machines running on dedicated physical hardware (server, storage, and network). Here the virtualization is limited to a single tenant or consumer. However, multiple platforms and applications can run on the same hardware/OS.	All or more than one consumer's virtual machine(s) share the same physical server. This is done at Hypervisor/OS level. The middleware and applications are still run as dedicated fashion for each consumer or tenant.
PaaS	Each consumer will get a dedicated platform (middleware) to host individual applications. The hardware is not visible to end users in this model, but each platform can run on a dedicated infrastructure.	All or more than one consumer's application(s) share single platform. The applications are run as dedicated for each customer, but middleware and infrastructure is shared among consumers.
SaaS and BPaaS	Each customer will have a dedicated application that can be either hosted on a single-tenancy platform or a multi-tenancy platform.	All or more than one consumers share the single application. The platform and infrastructure are shared among consumers.

A multi-tenancy model is intended to provide greater elasticity of available capacity and improved efficiency in terms of the cost. From a capacity planning perspective, it is important to ensure that the workload characteristics and service level requirements from the consumer are a good fit for the multi-tenancy model. As covered in Chapter 2, "Cloud computing workloads" on page 13, some workloads are more conducive to running in a cloud environment than others. Likewise, some workloads are more conducive to running in a multi-tenancy versus a single-tenancy model. See Figure 5-5.

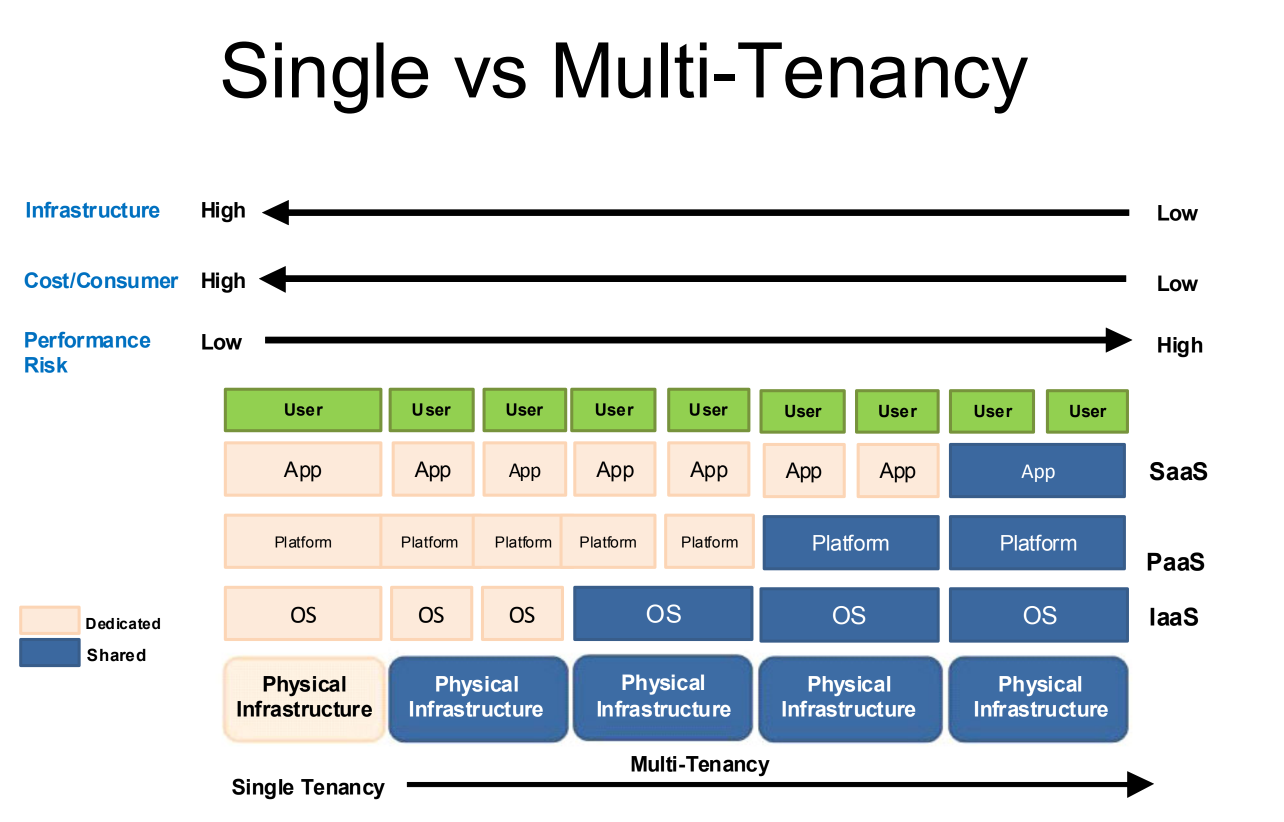


Figure 5-5 Single / multi-tenancy and service depth

5.2.7 Capacity sizing

Capacity planning has to be done by both the cloud consumer and cloud provider. The consumer has to plan their virtual machine requirements and evaluate which cloud provider can provide optimal and efficient infrastructure. The cloud provider must have the proper infrastructure to support different workload requirements of the consumers:

- Size the cloud consumer virtual machine requirements based on the business objective and service model.

  The consumer has to measure the resource requirements for unforeseen demand and respectively design their application to load balance and scale automatically in the target cloud infrastructure. The sizing of the consumer VM requirements can be done through best practice policies learned from experience and application benchmarking in test environments or can be based on historical resource usage data or a mix of both.

- Size the cloud infrastructure based on all consumer requirements.

  The capacity of private cloud infrastructure is sized optimally to suit the organization requirements and to handle unforeseen demands that the organization needs to plan effectively utilizing their infrastructure by either having enough idle resource pools or by prioritizing resource allocation among consumers. The dynamic increase and decrease of workloads in the public cloud infrastructure are high, and it has enough idle resource pools to scale and load balance automatically. A key aspect in the cross-consumer optimization is the business and technical policies that can impact how the consumer workloads can be sized and placed in the cloud on shared resources. For example, there might be business constraints that prohibit certain consumer VMs to share physical resources. There might be technical soft constraints to put certain types of VMs across consumers together on physical servers to reduce software license costs for a PaaS provider.

- Size the cloud management stack software hardware requirements.

  The infrastructure for the cloud management stack must be scalable to handle all consumer requests and manage the complete cloud infrastructure.

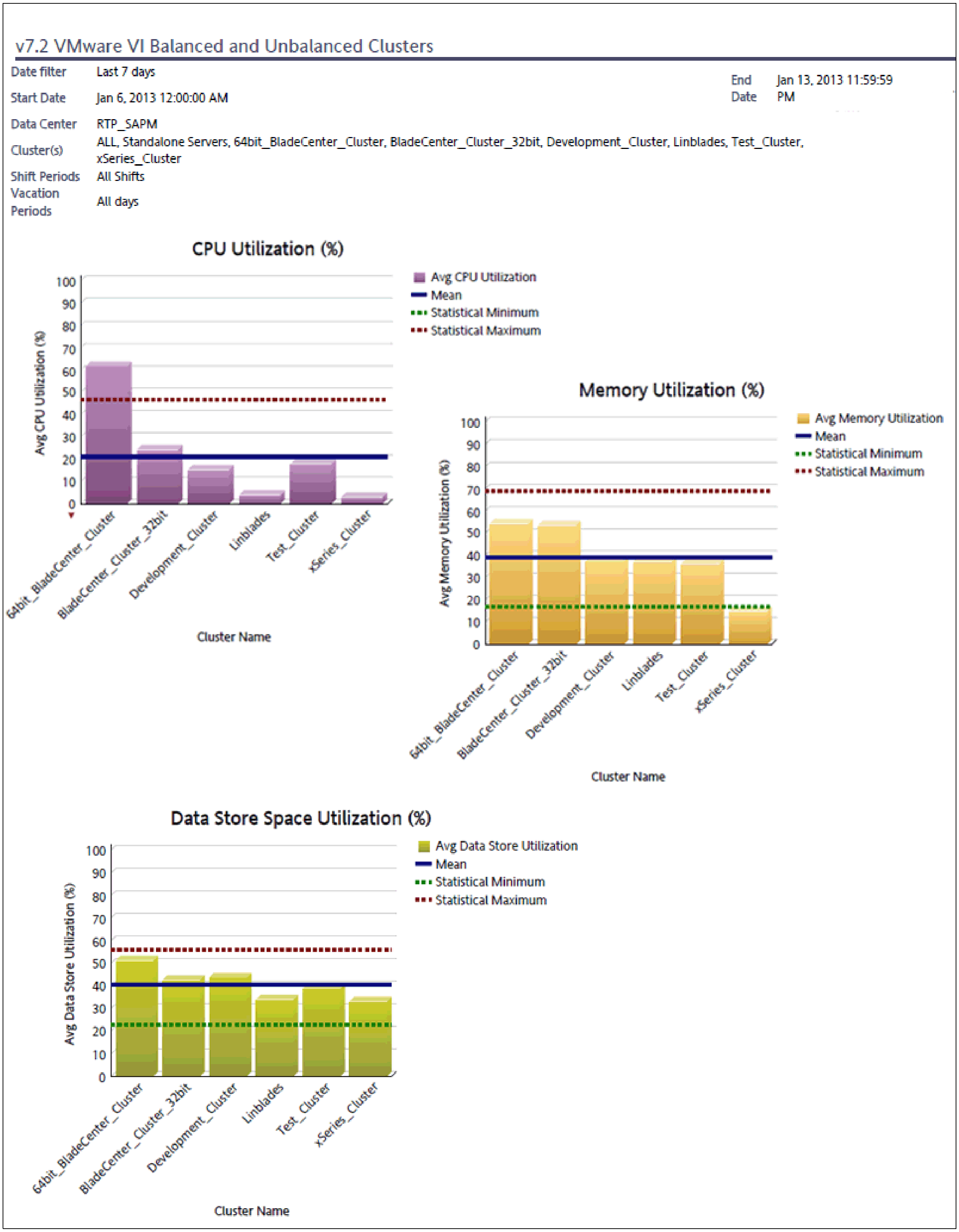
5.3 Monitor, analyze, and report capacity usage

Typically many cloud service consumers are allocated dynamic environments that are uncapped. For good cloud management, there must be upper limits and lower limits applied to maintain consistency of service within the cloud unit. The approach is that you get what you specify (within boundaries), and this way you aim for consistent delivery of service that remains constant over time and does not diminish as other loads vary. Even if not adopted, it is a design point that must be considered. An operational cloud environment can be monitored from the provider and the consumer side. A provider can ask several types of questions on the historical monitoring data to understand how the cloud is being used. Here are some sample questions:

- What is the aggregate health of the cloud and further drill down into Cluster / pool, host, and VM level?
- Which are my least or most used servers, pools, and VMs for a given resource type?
- Is there any bottleneck in my current environment and if so, where?
- Am I reaching capacity on resources and if so, which resource? When will I exhaust capacity?
- Do I have a significant difference in the usage reports from last week to this week?
- Are my resource pools balanced?

- How many more VMs can I add to a cluster host based on usage history?
- How much more resources do I need to add N additional VMs to the environment?
- How and where do I add capacity if existing systems are not enough for future growth for optimized capacity usage?
- Where do I place new workloads? Do I really need to add more resources?
- How can I optimize the VM placement to maximize usage and minimize costs?

Figure 5-6 on page 47 shows usage of VMware clusters across multiple resources to identify spares and guide VM additions.



v7.2 VMware VI Balanced and Unbalanced Clusters

Date filter: Last 7 days

Start Date: Jan 6, 2013 12:00:00 AM

End Date: Jan 13, 2013 11:59:59 PM

Data Center: RTP_SAPM

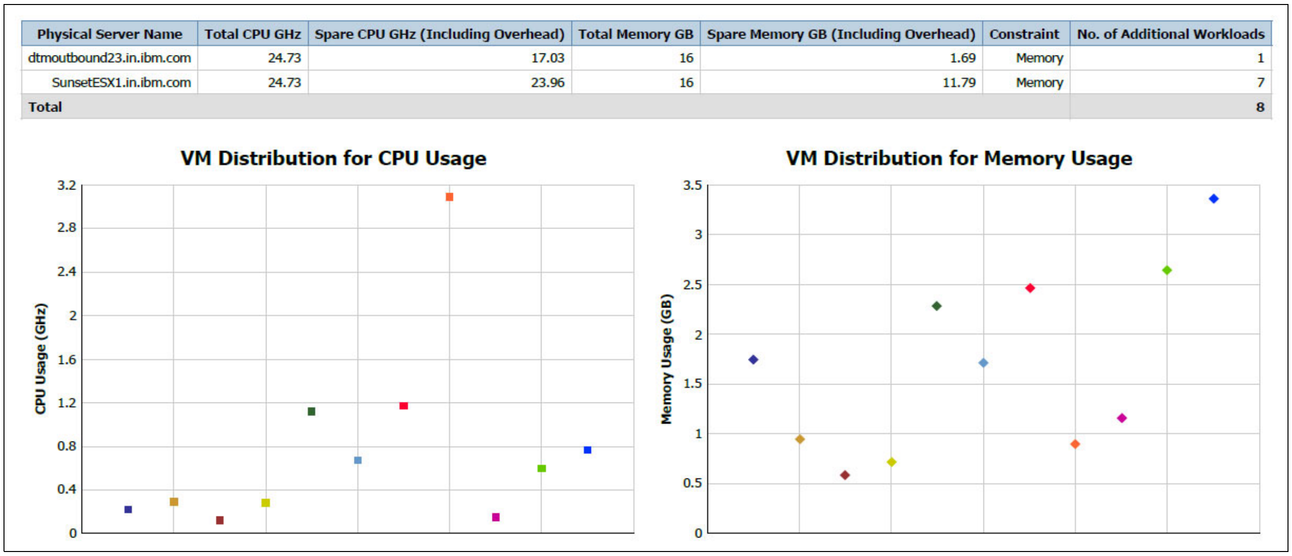
Cluster(s): ALL, Standalone Servers, 64bit_BladeCenter_Cluster, BladeCenter_Cluster_32bit, Development_Cluster, Linblades, Test_Cluster, xSeries_Cluster

Shift Periods: All Shifts

Vacation Periods: All days

Figure 5-6 Using VMware clusters to identify spares and guide VM additions

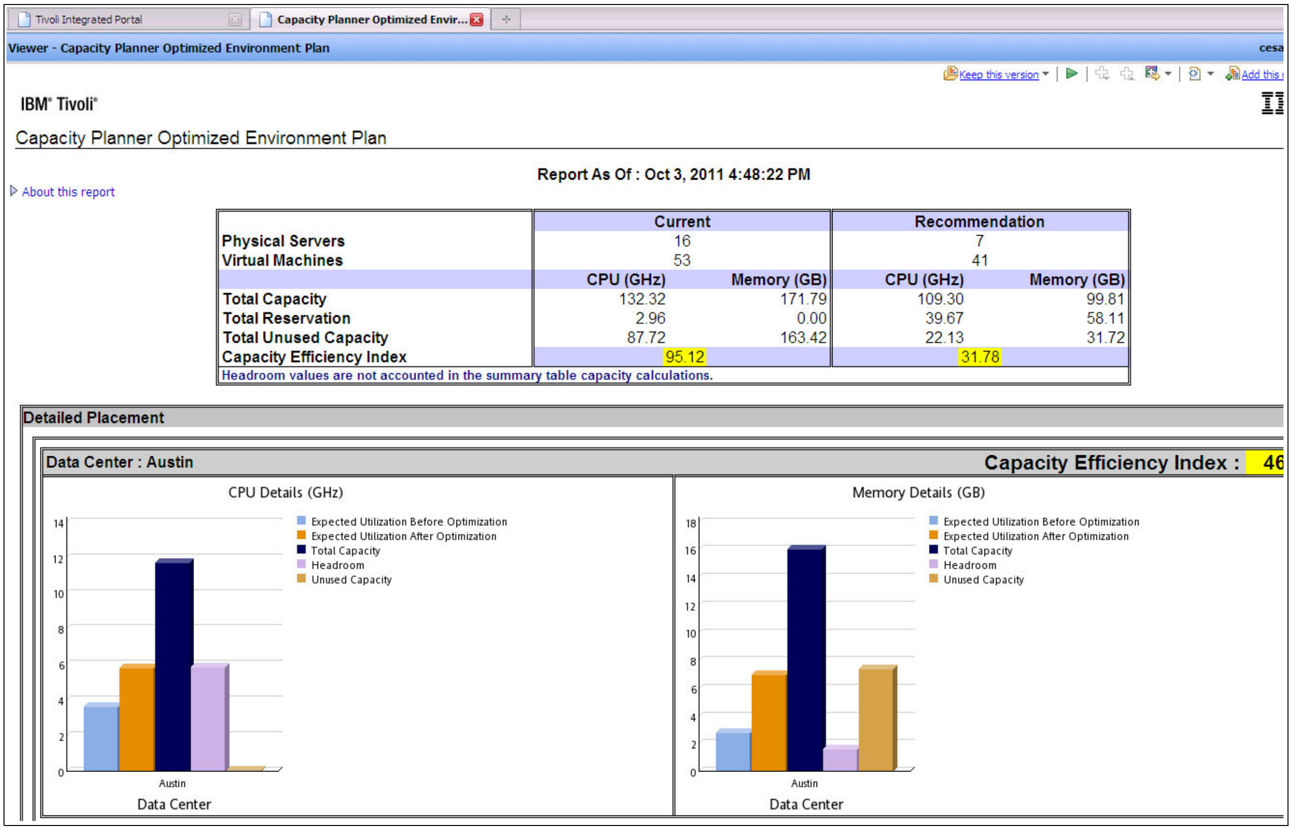
Figure 5-7 shows how many VMs can be added on specific hosts in a cluster based on historical usage.



Physical Server Name	Total CPU GHz	Spare CPU GHz (Including Overhead)	Total Memory GB	Spare Memory GB (Including Overhead)	Constraint	No. of Additional Workloads
dtmoutbound23.in.ibm.com	24.73	17.03	16	1.69	Memory	1
SunsetESX1.in.ibm.com	24.73	23.96	16	11.79	Memory	7
Total						8

Figure 5-7 Adding VMs

Figure 5-8 shows part of a placement optimization plan where a cloud with 16 hosts can be run on seven hosts based on analysis of historical usage and VM reallocations across clusters based on policies.



Tivoli Integrated Portal
Capacity Planner Optimized Envir...
Viewer - Capacity Planner Optimized Environment Plan
cesa
Keep this version
Add this
IBM® Tivoli®
About this report

Capacity Planner Optimized Environment Plan

Report As Of : Oct 3, 2011 4:48:22 PM

	Current		Recommendation	
Physical Servers	16		7	
Virtual Machines	53		41	
	CPU (GHz)	Memory (GB)	CPU (GHz)	Memory (GB)
Total Capacity	132.32	171.79	109.30	99.81
Total Reservation	2.96	0.00	39.67	58.11
Total Unused Capacity	87.72	163.42	22.13	31.72
Capacity Efficiency Index	95.12		31.78	

Headroom values are not accounted in the summary table capacity calculations.

Detailed Placement

Data Center : Austin

Capacity Efficiency Index : 46

Figure 5-8 Host consolidation

5.4 Supervising tuning and capacity delivery

Monitor and alert activities are only as beneficial as the remediation activities in place to maintain the system SLA. The main objective of remediation in a production environment is to restore service and SLA. Root cause analysis is critical because it ensures that the issues do not resurface if they were driven by user activity or an application. However, restoring the service and SLA is a top priority because it has significant business impact.

5.4.1 Root cause analysis

Increasing the resource allocation can only delay the symptom from reoccurring if it was driven by user activity or application issue. Examples include a DNS attack, user behavior much different than projected, or an application memory leak. In those situations, the cloud system can get temporary relief by increasing resource allocation. But additional user or application activities overwhelm the system. The proper step to take in this situation is to stop the activities from entering the system. For the case of user activities that deviated from projection, look to filter and funnel in the short term and discuss a proper long-term solution with the stakeholders.

5.4.2 Restoring service and SLA

There are two approaches to restore service and SLA. One is to tune and throttle based on the resources available. The other is to increase and allocate additional resources. Tuning is addressed in this section to emphasize that tuning is an iterative activity. While tuning during a capacity assessment is crucial, it is also an important technique to be used on an ongoing basis, particularly from an operations perspective, to react to changes in workload behavior patterns that impact service and to correct deviations from planned or forecasted capacity usage.

5.4.3 Tune

The objective of tuning is to ensure best system performance at a given resource level and to prevent the system from being overwhelmed by a sudden burst in traffic beyond projection. There are many knobs that can be tuned in a single system and many more that govern the interaction between systems. The golden rule is to not tune for the sake of tuning and to have a goal in mind, for example, to target tuning to address monitor metrics and resource indicators that are over the threshold.

The system is tuned as part of the initial capacity sizing matching the projected user workload and traffic volume. However, either due to changing behavior or incorrect assumptions, the actual can vary and overload the system. In a cloud environment with multi-tenants, the workload and volume will also vary depending on the business model of individual tenants. The most critical part of tuning a dynamic environment, such as a public cloud, is to ensure that the system is properly throttled to prevent the overall system from going down due to a single increased factor. The throttling methodology, shown in Figure 5-9 on page 50, is to tune the system to limit the concurrent executions beyond a level that the system can meet the SLA. This concurrency level will increase as more hardware becomes available or as application concurrency improves. Example, if an application will already consume 90% of the available heap with 3,000 concurrent users, we need to throttle the system to only let in 3,000 concurrent uses at the most. Users beyond 3,000 queue outside the application to not overwhelm the system. By following this methodology, the application allows quick execution of transactions in flight instead of coming to a halt with an overload.

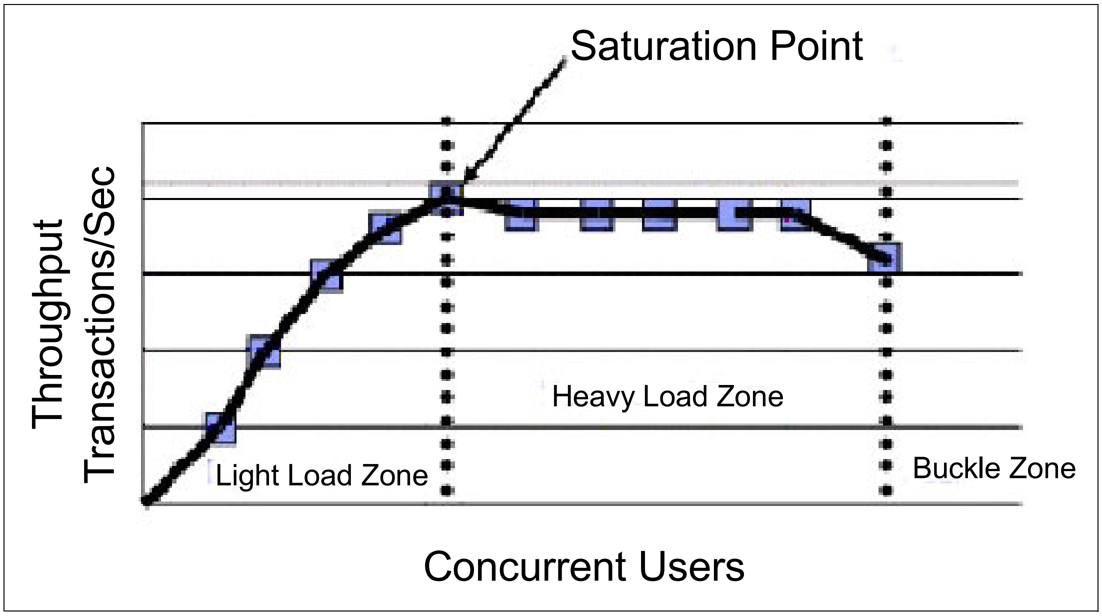


Figure 5-9 Tuning overview

This tuning approach is iterative and must be revisited as the application concurrency is improved (for example, let in 25% more concurrent users) or if the available capacity changed (fewer CPU/memory resources are available).

5.4.4 Increasing resource allocation

There are many cases where the application utilization varies. In a retail space, this can be seasonal to events, such as Valentines' Day, Back to School, or Christmas. As much as we can to evenly distribute workloads across the available hardware, there might be cases where an overlapping of events from different tenants can overwhelm the system. In this case, we can look to increase the overall capacity to satisfy the increased user volume.

This approach, however, is only viable if the bottleneck is not related to concurrency limits on a single resource. For example, the cloud application can only sustain 1,000 concurrent database operations. In this case, an increase in available database resources will not improve the maximum number of users the application can sustain. To improve the concurrency of this application, deeper profiling needs to be performed to understand the concurrency limitation, and an application redesign might be needed.

5.5 Producing and maintaining the capacity plan

The objective of this activity is to develop, maintain, test, model, and revise alternative approaches in satisfying various enterprise-shared resource requirements. This activity is the culmination and synthesis of the other activities. It must synthesize multiple data sources and consider the controls that bound options for a solution, such as the predefined configurations available in a public cloud for specific workloads:

- The inputs to this activity are forecast assumptions, forecast projections, and subject matter expert recommendations.
- The controls for this activity are financial constraints, hardware constraints, performance policies, resource standards and definitions, and strategy and direction.

- The deliverables from this activity are the agreed capacity plan that includes alternative solutions and an optimized resource solution.

The Capacity Plan will detail existing usage of critical IT resources under management. Typically, for servers this involves reporting and trend analysis for CPU, I/O, memory, storage, and the network interfaces. The Capacity Plan might also include correlation of IT resource usage to IT applications (services) and business usage or workload patterns. Similarly, planned business activity and IT application changes and deployments might be factored into forecasts for IT resource requirements.

This brief description of the Produce and Maintain Capacity Plan activity clearly highlights that its content covers all three sub-processes, BCM, SCM, and CCM. To do so, the capacity planning team must gather input from a variety of sources and analyze that data from various perspectives.

The Capacity Plan must address views of IT resource forecasts developed by aggregation of the individual resource requirements:

- IT component or resource composite views, for both physical and virtual, can include:
  - Servers (OS, CPU, memory, I/O queues), storage, data network, and voice network
- Service or application composite views can include:
  - Middleware, databases, applications, and workloads
- A composite view of IT resource requirements can be analyzed at various levels of breakout and detail combined with the above IT component and service views:
  - Locations, business functions or organization, business processes, and asset status (dedicated or shared: single tenancy or multi-tenancy)

A capacity plan for cloud must focus on the workload behavior patterns and forecasted demand to be viewed from multiple timeframes, as shown in Figure 5-10 on page 52:

- Seasonal peaks for workloads must be considered as well as event-driven increases, such as market campaigns, mergers and acquisitions, new project deployment, and so on. This is particularly important for shared or multi-tenancy cloud environments to ensure that peaks do not collide and introduce potential capacity constraint and performance degradation.

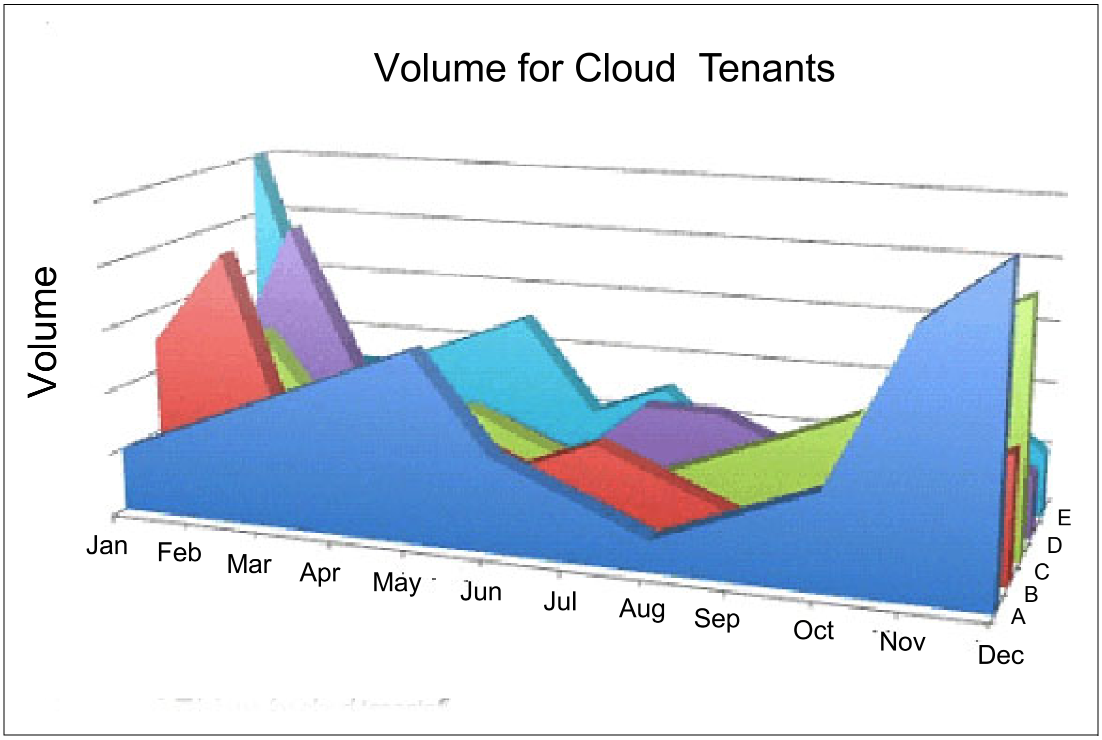


Figure 5-10 Volume for cloud tenants

A plan versus actual analysis typically applies various composite or aggregate views along with deep-dive drilldown when analyzing deviations from the plan.

5.6 Determining the need for capacity planning for cloud computing

For capacity planning, the complexity of the consumer, integrator, provider roles, and bounded responsibilities will alter the challenges of forecasting business demand and corresponding IT resource consumption. In a more traditional capacity planning team, while there is no formalized definition of or distinction between these three roles, similar challenges exist. These roles also offer a renewed opportunity to enhance capacity planning for cloud to address the three focus areas of Capacity Management defined in 5.1, “Overview” on page 36: Business Capacity Management (BCM), Service Capacity Management (SCM), and Component Capacity Management (CCM). The RACI Table 5-2 on page 53 is intended to prompt further thinking on the need for capacity planning for cloud that embraces these three focus areas within the context of the various inter-relationships of the consumer, integrator, and provider of the cloud service.

Capacity planning cannot be dismissed as a set of activities that will be automatically managed by virtualization and sophisticated algorithms to manage workload. While these technical advances will definitely provide greater capability to support increasingly large and complex and variable workloads running in a cloud infrastructure, there is still need to conduct capacity analysis and planning to understand the business, service, and component requirements and corresponding solutions as workloads change over time: changes based on the seasonal behavior of a particular workload (which is common by industry), changes based on unexpected volumes due to external drivers, such as surprise marketing campaigns or changes due to changing growth drivers, such as mergers and acquisitions, or changes in the functional and architectural characteristics of an application's workloads.

As addressed in 3.4, “Process for application performance testing on cloud infrastructure ” on page 24, the consumer/integrator/provider perspective and the service depth being delivered by the cloud influence the boundaries of responsibilities for capacity planning. For capacity planning, the complexity of the consumer, integrator, and provider roles and bounded responsibilities can increase the challenges of forecasting business demand and corresponding IT resource consumption.

The RACI matrixes in Table 5-2 show the responsibilities of the different stakeholders in the different situations. The meaning of the abbreviations in the table is:

R	Responsible
A	Accountable
C	Consulted
I	Informed

Table 5-2 Stakeholder responsibilities by situation

Service Depth	Capacity Management focus areas	Capacity Management Activities	Consumer	Integrator	Provider
IaaS	Component Capacity Management (CCM):	Model and size	I	A	R
PaaS		Monitor, analyze, and report	I	A	R
		Supervise tuning and capacity delivery	I	A	R
		Produce and maintain capacity plan	I	A	R
SaaS	Service Capacity Management (SCM):	Model and size	C	R	I
		Monitor, analyze, and report	I	R	A
		Supervise tuning and capacity delivery	I	R	A
		Produce and maintain capacity plan	I	R	C
BPaaS	Business Capacity Management (BCM):	Model and size	C	R	I
		Monitor, analyze, and report	C	R	A
		Supervise tuning and capacity delivery	C	R	A
		Produce and maintain capacity plan	C	R	A

Figure 5-11 Capacity plan

6

Conclusion

Our initial Redpaper[2], written in 2010, discussed performance implications for cloud computing in rather general terms, since the whole area was still relatively new. With the benefit of additional industry experience, it was appropriate for this Redpaper to spend more time on what we learned about deploying and managing cloud solutions that perform well while being cost-effective.

We are seeing two extremes in leading-edge computing. On one hand, we have purpose-built, highly customized systems such as Watson, in which every tier of the solution is optimized for the high-performance computing needs of a specific domain. On the other hand, we have utility cloud computing systems that must be able to rapidly allocate computing resources in response to new and diverse workloads and then characterize and manage those workloads on an ongoing basis. Can today's Watson become one of tomorrow's utilities? For cloud computing to take its proper place in the “IBM Smarter Planet” toolbox, we must build intelligent clouds that can handle increasingly diverse, variable, and challenging workloads while fully exploiting the emerging capabilities of new technology.

In this IBM Redpaper, we discussed how contemporary approaches to testing, monitoring, and capacity management of cloud solutions are starting to address these concerns. For example, yesterday's performance testing was typically concerned solely with the responsiveness and utilization of a single application at a time. In today's cloud environments, we must also ensure that the cloud management solution components perform well, and that the overall solution exhibits sufficient elasticity to manage multiapplication (and sometime multi-tenant) workloads. This places new demands not only on how we do performance testing, but also on how we monitor performance and manage capacity as well. Who takes on these responsibilities depends largely on whether a public, hybrid, or private cloud solution is chosen.

We also provided examples of how cloud computing is used to address tomorrow's problems today, but we understand that this only scratches the surface of what is being done. We want to hear from you about your journey into cloud computing, particularly the performance and capacity challenges you encountered along the way. Contact one of the authors to learn more.

[2] http://www.redbooks.ibm.com/redpieces/abstracts/redp4875.html

6.1 References

- The prerequisite Redpaper: *Performance Implications of Cloud Computing,* SG24-4875

  http://www.redbooks.ibm.com/redpieces/abstracts/redp4875.html

- IBM SmartCloud™ Capacity Sizing Guides

  http://www.ibm.com/partnerworld/wps/sizing/protect/sizingguide/RunSG?guide_id=sgq00100113170112102

- *Is Your Company Ready for Cloud: Choosing the Best Cloud Adoption Strategy for Your Business*, Pamela K. Isom and Kerrie Holley

  http://www.amazon.com/Your-Company-Ready-Cloud-Choosing/dp/0132599848

- Article (WSLA2002): Keller, A. & Ludwig, H., T*he WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services,* Journal of Network and Systems Management, Springer, New York, 2003, 11, 57-81
- Manual (WSLA1.0): Ludwig, H.; Keller, A.; Dan, A.; King, R. P. & Franck, R., *Web Service Level Agreement (WSLA) Language Specification*, 2003
- Cloud attributes

  http://www.ibm.com/developerworks/cloud/library/cl-rev2sla.html?ca=drs

A

Key terms

The following terms are used in this paper.

Quality of Service Refers to operational attributes characteristic of good performance that a system is expected to exhibit, such as the ability to complete tasks quickly (responsiveness), the ability to accommodate workload growth (scalability), the ability to accommodate sudden workload changes while avoiding unplanned outages (resiliency, stability). QoS targets can be incorporated into system requirements during development and into service level agreements (SLAs) after the system is in production. Specific metrics used to determine whether SLA goals are being met are often referred to as Key Performance Indicators (KPIs).

Virtualization Refers to the ability to share and manage computing resources more effectively. Virtualization is achieved by configuring virtual computing resources (with which users and applications interact) to share designated pools of physical computing resources. While some virtualization capabilities were available on mainframe platforms for decades, vendor improvements in sophistication, automation, and platform coverage made virtualization a key enabling technology for cloud computing as well.

Workload Refers to a grouping of similar or related computing tasks used to help test or manage system performance and capacity. Workloads can be characterized by type, mix, and arrival rate patterns of incoming requests plus the software and application programs executed by those requests. Workload Management refers to the capabilities of operating systems and virtualization platforms to respond to workload changes in a policy-based manner. Workload management is particularly important for cloud systems used to host multiple dynamic workloads.

Orchestration A sophisticated form of workload management that coordinates changes to the allocation of resources across multiple virtual environments.

Elasticity Refers to the effectiveness with which a system responds to sudden changes in workload with minimal performance impact. Elasticity is especially important in cloud environments, where services and their associated workloads are being activated and deactivated on an ongoing basis.

Cloud Management Refers to the process by which cloud services are defined, made available for selection, activated to consumers, managed during execution, deactivated, and withdrawn. The cloud management process is implemented using a Cloud Management Platform, which automatically applies required changes to the virtualized resources in the cloud environment. Since cloud management is a key part of the business value proposition for cloud computing, cloud management platform functions must be considered as candidates for performance testing and monitoring, in addition to the testing and monitoring typically done for virtual environments and applications running in the cloud.

Tenant A customer organization that is renting cloud services. Tenancy refers to the manner in which a service provider assigns tenants to cloud environments. Typical tenancy models include single tenancy (one tenant in one cloud environment, as in a Private Cloud), multi-instance (multiple tenants with each running in a separate cloud instance) and multi-tenant (one cloud instance shared by multiple tenants, as in a Public Cloud).

IT Infrastructure Library

ITIL is an IT Service Management process framework originally developed in the United Kingdom (UK), which gained wide-industry acceptance. IBM Tivoli Unified Process (ITUP) describes detailed industry best practices for IT Service Management based in part on the ITIL V3 framework. ITIL and ITUP contain guidance for Capacity Management that is relevant to cloud computing environments, among others.

B

IBM Tivoli Unified Process (ITUP)

To read about IBM Tivoli Unified Process (ITUP):

1. Go to the following web site:

   http://www-01.ibm.com/software/tivoli/governance/servicemanagement/itup/tool.html

2. Download and install the ITUP. Afterwards, there is an ITUP icon on your desktop.
3. Open the ITUP application, and at the bottom of the left navigator, select **Process Content** → **IT Processes** →**Processes by Name**. From the list that appears, choose **Capacity Management**, **Demand Management**, and so on.

IBM Tivoli Unified Process (ITUP) provides detailed documentation about IT Service Management processes based on industry best practices. ITUP is strongly aligned with industry best practices, including the recently released ITIL V3 best practices. ITUP gives you the ability to significantly improve your organization's efficiency and effectiveness. ITUP enables you to easily understand processes, the relationships between processes, and the roles and tools involved in an efficient process implementation.

ITUP contains the following resources:

- Process Content
- Tool Mentors
- Roles
- Work Products
- Scenarios

C

IBM Tivoli tools

Here are some tools for planning and monitoring.

Tools for virtual environment capacity planning

The tools for virtual environment capacity planning are:

- IBM Tivoli Monitoring for Virtual Environments v7.1 is shipped with the product. It is not available independently. Monitoring, dashboards, reporting, and planning capabilities of a virtual environment are part of this product. Version7.1 handles only VMware planning and dashboards, although it can monitor and report on other hypervisors. Future releases can support IBM POWER® systems.
- The SmartCloud Monitoring bundle that packages Tivoli Monitoring for Virtual Environments V7.1 along with a few other products, including the capacity planner. Clients purchase either the Tivoli product or the SCM bundle.

Tools for distributed infrastructure monitoring

The tools for distributed infrastructure monitoring are:

- In virtual environments, use IBM Tivoli Monitoring for Virtual Environments v7.1. IBM Tivoli Monitoring is the base product for the monitoring framework that is used by the monitoring agents.
- IBM Tivoli Monitoring v6.2.3 contains predictive analytics using IBM SPSS® Forecast (an IBM Business Analytics division product). If SPSS Forecast is purchased separately by a client, the base product can perform non-linear projections on usage data and predict capacity shortage. It can also generate alerts based on any thresholds set on the projected values.
- For application monitoring (which is not used by the capacity planner), there is a set of products available under SmartCloud Application Performance Management (SCAPM).

Physical environment capacity planning

The physical environment capacity planning tools are:

- There are tools, such as Opera/Sonoma from HiPODS.
- IBM offers Services engagements for this process.
- IBM SPSS products can be used to build custom models.

IBM®

Performance and Capacity Themes for Cloud Computing

Redpaper™

Selecting workloads for cloud computing

Planning for performance and capacity

Monitoring a cloud environment

This IBM Redpaper is the second in a series that addresses the performance and capacity considerations of the evolving cloud computing model. The first Redpaper publication (Performance Implications of Cloud Computing, REDP-4875) introduced cloud computing with its various deployment models, support roles, and offerings along with IT performance and capacity implications associated with these deployment models and offerings.

In this redpaper, we discuss lessons learned in the two years since the first paper was written. We offer practical guidance about how to select workloads that work best with cloud computing, and about how to address areas, such as performance testing, monitoring, service level agreements, and capacity planning considerations for both single and multi-tenancy environments.

We also provide an example of a recent project where cloud computing solved current business needs (such as cost reduction, optimization of infrastructure utilization, and more efficient systems management and reporting capabilities) and how the solution addressed performance and capacity challenges.

We conclude with a summary of the lessons learned and a perspective about how cloud computing can affect performance and capacity in the future.

REDP-4876-00