APPENDIX A

CLOUD PERFORMANCE MONITORING COMMANDS

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

A.1 `vmstat` COMMAND

`vmstat` command: Account virtual memory information

Syntax

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

`vmstat [-V]`

Description

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

We will discuss the different parts of the `vmstat` command.

Process Part

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

Memory Part

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

Swap Part

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

IO Part

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

System Part

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

CPU Part

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

A.2 `iostat` COMMAND

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

Syntax

```
iostat [ ] [ ]
```

Description

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

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

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

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

Device Information

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

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

A.3 mpstat COMMAND

mpstat – Provides account processors information.

Syntax

mpstat [ ] [ ] [ ]

Description

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

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

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

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

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

A.4 netstat COMMAND

netstat – Reports network configuration and activity

Syntax

netstat [ ] [ ]

Description

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

Information about Routing Table

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

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

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

`G` The route uses a gateway.

`U` The interface to be used is up.

`H` Only a single host can be reached through the route.

`D` This route is dynamically created.

`M` This route is set if the table entry was modified by an ICMP redirect message.

Network Interface Information

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

Network Connections Information

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

`-t` This option shows active TCP connections.

`-u` This option shows UDP connections.

`-w` This option shows RAW connections.

`-x` This option shows UNIX socket connections.

A.5 `ipcs` COMMAND

`ipcs` – Provides information about active interprocess communication facilities

Syntax

ipcs [ ] [ ]

Description

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

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

`-m` shared memory segments

`-q` message queues

`-s` semaphore arrays

`-a` all (this is the default)

The output format may be specified as follows:

`-t` time

`-p` pid

`-c` creator

`-l` limits

`-u` summary

A.6 ps COMMAND

`ps` – Reports process status

Syntax

ps [options]

Description

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

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

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

PID: Process Identification Number

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

TIME: The time at which the process is being executed

COMMAND: Name of the command

The ps command has the following options:

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

The ps command with the -f option is given as

```
$ps -f
```

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

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

UID: User_id

PID: Process_id

PPID: Parent PID

C: Amount of CPU time consumed by the process.

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

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

TIME: The time at which the process is being executed

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

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

Here is another ps command.

```
$ps -u kumar
```

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

ps Command Options

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

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

Process Selection with List

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

ps Output Format

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

A.7 top COMMAND

top – Provides information regarding the running processes

Syntax

top [ ] [ ]

Description

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

top Command Options

-d This specifies the seconds and tenths of seconds of delay. The default is 3 seconds for the two consecutive updates.

-n This gives the maximum number of iterations or frames.

-p This monitors only processes with specified process IDs.

-s This allows you to execute top in secure mode and when it is executed from root.

Description of the Fields

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

A.8 sar COMMAND

`sar`- Informs the CPU activity.

Syntax

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

Description

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

Options

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

A.9 load COMMAND

`load` – Loads machine code and initializes new commands.

Syntax

load fileName

load fileName packageName

load fileName packageName interp

Description

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

A.10 xload COMMAND

xload – Displays the system load average

Syntax

xload [ ] [ ] ...

Description

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

A.11 tload COMMAND

tload – Graphical representation of system load average

Syntax

tload [ ] [ ] [ ] ...

Description

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

Options

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

A.12 `uname` COMMAND

`uname` – Prints system information

Syntax

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

Description

Prints certain system information.

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

A.13 `opcontrol` COMMAND

`opcontrol` – Controls the profiling session

Syntax

`opcontrol` `[ ] [ ]`

Description

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

Options

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

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

A.14 accton COMMAND

`accton` – Enables/disables system accounting

Syntax

`accton` `[ ]`

Description

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

A.15 SUMMARY

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

APPENDIX **B**

UNDERSTANDING SIZING LIFECYCLE

B.1 INTRODUCTION

To be competitive in the world, customers are looking for hardware configurations that match the customer requirements. This needs a right size and consistent sizing process. In order to accomplish this needs the understanding of the sizing. So we will discuss the sizing as a process cycle. This section we will discuss with the help of scenario to understand the sizing as a whole. Here is the small case:

B.1.1 Scenario

ABC Bank Inc., big multinational, India-based bank has managed its business successfully for many years since its inception. The bank has now decided to automate its operations to scale up its business. The bank has customers holding savings bank account current account and credit cards. The bank's management board has decided to provide Internet banking feature to the existing savings bank customers. It has three kinds of loan facility also for its Savings Bank customers, which are housing loans, personal loan and vehicle loan. They need a complete banking system for highly consistent, scalable and cost efficient web based banking solution for customers of all sizes around the globe over the cloud services.

In the above situation, the right-sized hardware for banking solutions can make all the difference between successful cloud deployments. It is important to size whenever shopping for a new system or updating a data center for cloud infrastructure. Our sizing solution will require the right processing power, amount of memory, and type and amount of storage for any application-specific job or workloads otherwise it will be difficult the best system(s).

A process sizing should not estimate immediate need but also the growth based solutions to meet the capacity requirements of a particular customer. This needs the combination of hardware and software.

Sizing is a process to estimate system resources that would be required to meet the current and forecasted capacity requirements of a particular customer. Sizing solutions dictate a specific mix of software and hardware, and the right hardware component is critical to meeting customer requirements. This helps the potential customer to estimate the precise solution. It will be big asset to cloud customer if they come to know about hardware configuration and cost.

Now the question is when can we involve the sizing process? This can be initiated when the organization is looking for infrastructure upgrades, consolidation or even totally new installments. It can also be performed when we are estimating the immediate and growth based workload characterization. In order to do so, we need the sizing data. This can be collected from the performance benchmarks, directly from the production or even from the application field data. The type of data that we need as sizing data for the above banking solution can be consumption of key system resources like memory, processor, disk and/or network operations or the factors that impact the solution. It will also require data for calculating the growth based solution as per-user, per transaction like processor utilization per transaction or user, memory required per user, disk storage required per user network operations per transaction, number and size of disk reads/writes per transaction.

This can be in the form of a complex questionnaire based on the banking solution. In order to calculate the CPU and memory requirement sizing solution as well as storage needs, we may require to determine total registered users of the application. We need to compute the number of peak concurrent users necessary from CPU and memory perspective, with total registered users of the application required for determining storage needs. For our banking solution, we can even need peak business transactions per second (tps) required as well as proportional details of transaction types like queries, updates requests, whether the transactions are in online or batch mode, etc. may be required.

The solution looks as an output for supporting operating systems, number of sockets/ physical processors populated, number of cores utilized and projected CPU utilization of the requested workload. At the disk-level, it will look for internal and external storage, number of disks and GB required. Now we are in the green IT age, thus our sizing should also looks into the energy consumption of recommended system for planning cooling requirements as well.

B.2 SIZING LIFECYCLE

Now we will discuss the banking solution-based sizing lifecycle to recommend a suitable hardware configuration.

B.2.1 Setting the Expectation

As a first step to sizing, we should analyze the actual objective of the sizing the solution. The sizing solution will be more precise if we are fixing the accuracy of our solution as this will be a forecasting. Once we fix our boundaries, it will be easy to collect the significant data for analysis as it will need the complex set of data. It is important to fix all the sizing factors judiciously otherwise it will not be possible to size the actual required mission based solution. This helps in avoiding extra cost and effort for our required solution.

B.2.2 Gearing Up

It is important to be clear about the actual requirements, architecture, application behavior and resources to make our sizing lifecycle smoother. The inputs from the customer may be routed through the sales channel can play the vital role to complete the process. The basic information that is required for the above banking solution can be CPU, memory and disk requirements.

With respect to CPU, we will need the banking transaction volumes. This can be known from the basic question like total number of users. But it cannot be the only question as CPU will need information like load average because it will not be same throughout the day. Another question can be transactions are happening in the web portal based online application.

In order to know the memory requirements based on users, we can ask how much memory is required per account maintenance in the banking solution or how many transactions exist.

These next requirements, disk, looks for the input data in form of the questions like what type of documents are stored and its size. How often does the data need to be archived? Which type of RAID is required?

Here we have discussed some of the basic question that can play the vital role in our scenario. So it is important to come up with most viable and effective set of questions for sizing lifecycle.

B.2.3 Setting Up the Environment

Now we are ready with the questions the next step is to understand the measuring and gauging parameters and finally evaluating them to size the application. In this journey first we will try to know the workload characteristics. If the application is used by many users then the resources needs will be more complex to size. At the same time, when the number of users are there in small sample, it will be simple to size the application.

There can be different set of customers based on the need and application requirements. It is important to know the functionality of the application as per the database volumes as per the transaction details of the customer. We can list the transactional details based on the user groups and their interest.

Now we are ready to execute investigative tests to understand the application tendency at different workloads and it will help us for the behavioral features of application. After this we can go for the functional and load testing scenarios. We can buy some tools to perform the same and play with different simulation for different conditions.

B.2.4 Get Set Go

Now our test bed is ready and this phase will use all the finalized tests for our workload. Here it should meet the test specifications and will actually lead towards the estimated targeted system. This will give us the option that we can see our system running like in the production environment. We can gauge the initial performance of the system and review the initial system requirements and settings. If the performance is not matching, we can fix the issue and make adjustments to deal our situation and start the process once again. It is important to maintain the entire checklists of item that are under consideration to collect the application data required for sizing life cycle.

Now we are on the verge of collecting the final data so we can emphasize on the most essential parameters. We can find the relation between the transaction and CPU required. This may require some formula to estimate. The important parameter that we should emphasize is memory. It can be calculated per web based sessions or depending on the number of user. This can be even at each server level or based on transaction. The next thing that can be collected refers to storage point required by the application. It also refers to read and writes to I/Os as well as I/O request to CPU and vice versa to determine the number of disk drives.

B.2.5 Tapping the Opportunity

We are ready with the data. We have all the relationships for the application behavior for recommending the hardware configuration. Now this is time to generate the sizing results to compare within the family of servers of the same company or competition based hardware that suits the customer requirement. Now, this is the time to validate all the results before

going live. At this level, we can take the evaluation suggestion if any to change the sizing solution if required.

B.3 SOLUTION TIER

Now we can discuss the overall environments of the banking solution because it may require the OLTP, Non-OLTP and Web server requirements.

B.3.1 OLTP

In banking solution there is the need of various data archival and retrieval transaction based operation and online transaction processing (OLTP) helps in the requirement to support various SQL transactions required for the above said banking solution.

First of all, we should describe the layer that we ware wishing for the solution. We should comment on the DB layers, Application layer and Web layer. The next step is to define the application architecture like all the applications are running on separate servers or it is logically partitioned on the single server. So we can define all the functional layers of the tiers.

The other activity is to understand the different components of the OLTP environment. We can get the answers of the questions like which database and version the banking solution workload is using. Whether the banking solution is requiring High availability option to be configured and what is total storage requirement. The next important aspect is the size of the database and even can now the CPU utilization of the DB tier.

We can also know the interactive part of the OLTP environment based on the applications users groups like total number of users, percentage of active users, transaction per our users and its complexity. The application group can be portal, account holders, bank internal employees and privileged customers.

The next section for the OLTP process requires the inputs on the basis of workload characterization. The inputs can be type of queries, reporting mechanism, batch information and application management requirement. This information can be maintained in the form of complexity like what will be its value during peak and off-peak time and IO rate for all the activities.

B.3.2 Non-OLTP

It is not certain that our banking solution will require on OLTP environment sizing but it can posses the non-OLTP workloads for the application like data warehousing, decision support systems and Business Intelligence systems. Therefore, like our OLTP environment we need to characterize the server tire requirements. First of all we should describe the layer that we are wishing for the solution. We should comment on the DB layers, Application layer and Web layer for the non-OLTP environment.

Like our OLTP environment we need to know that all the application are running on the separate servers or they are working virtual environment. This will give the functionality of

each layer with its description. Now we can know the Non-OLTP environment basic activities details like database and its version or whether compression is required or not. We may ask also the application encryption details. It is also important to know about the database segment available to the different groups of the users.

In this environment, we should know the percentage of processed queries. This needs the analysis of the query type. First thing is to analyze whether the query is light, medium or heavy. Then we can know for the specific query how many active users are there or queries per day and even the query mix percentage we can calculate it in the simple, medium and complex bracket.

Our banking solutions require various data extraction process like extracting, transforming, loading and loading the data as different operations require the data from production environment to staging server where transformation happens. In order to accomplish this, we can collect the data like the batch details, time required to run that batches. We can even know that these batch processes should run concurrently with the regular ETL processes or not. It will be good if we know the input size of the data also.

B.3.3 Web Server

In case of the Web tier for the banking solution, the Web server will play the most important role as management requires the banking solution over internet. The functionality for the banking workload will be contributed by many web server configurations factors. This can be sessions are required by the web server at the peak time. At the same time how many licenses are required to match the requirements for web requests. We can take the quote on the session response time that will help us defining many SLAs. We can even collect the data for the complex workflow through the web server.

B.4 SUMMARY

This appendix has given the insights for cloud infrastructure, hardware architecture and sizing. We have seen the sizing practices in OLTP, non-OLTP and Web server environment. This appendix also covers the sizing lifecycles its components and steps required to perform it.

APPENDIX C

Understanding PC environment

VDI: Cost Factors

Case Study

Summary

C.1 UNDERSTANDING PC ENVIRONMENT

Over the past two decades, PCs have transformed from a single product was shared by a number of users across organization for data computation to an individually owned tool required to perform basic functions.

C.1.1 Lifecycle of PC

As we are trying to calculate the business benefits of VDI solutions, it is important to understand the lifecycle of PC. The PC lifecycle can be split into four stages (Figure C.1):

1. **Acquisition:** PC hardware and software must first be acquired. Depending on the size of the organization, IT may have very high or very low levels of involvement. Currently, with many options available in the market, most of the organizations have a dedicated procurement cycle.
2. **Deployment:** After acquisition of PC hardware and software, IT department is responsible for maintaining an inventory of the PC's assets, preparing the PC for delivery to the user by installing applications and data, and then physically delivering the device to the user. This phase in the lifecycle of PC is extremely critical as there is a change (management) of user experience involved in this phase.
3. **Maintenance:** After delivery of PC to a user, IT is responsible for maintaining the PC to ensure that it continues in its working state. For this, IT should provide updates, keep track of any hardware/software changes made to the PC, and service the PC in the event of failure. This phase keeps the users satisfied and helps maintain user productivity.
4. **Retirement:** Once a PC has reached the end of its useful life for a given user, IT is responsible for performing the tasks necessary to remove it from the environment or give it to another user. This process can involve various tasks, but typically includes deleting sensitive data, removing installed applications to harvest licenses, and disposing the device.

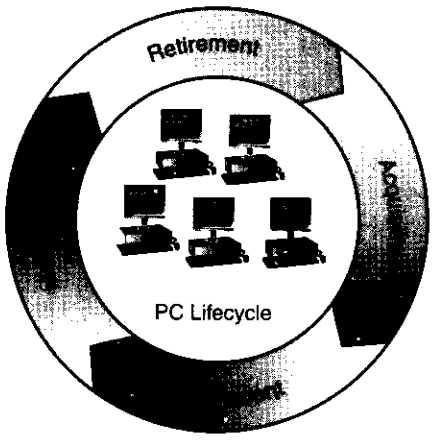

FIGURE C.1 Lifecyle of PC

To simplify the tasks that make up the PC lifecycle, technology vendors have developed a range of solutions that have been adopted in significant numbers, including configuration management databases (CMDBs), electronic software distribution (ESD) tools, asset management tools, hypervisors, among many others.

C.2 VDI: COST FACTORS

The following cost factors need to be considered while calculating the comparison between traditional desktop and Virtual Desktop Infrastructure (VDI) environment:

1. **Hardware cost:** It is important to understand that we need to go through the entire lifecycle after a fixed period, usually three years, for all the PCs as a part of technology refresh cycle. After this duration, the hardware and software available that provides computing capabilities will become redundant and newer technology will be available that will be more efficient and faster than the older technology. Although the upfront investment in building VDI solution is higher, there is a significant increase in the lifespan of hardware. Traditional PCs and their storage typically have useful lives of three years, whereas thin client hardware and data center storage used as a part of the VDI solution have a typical lifespan of six years. Currently, we are assuming the following components as part of the hardware calculations in the VDI solutions:
   - Thin clients
   - Servers
   - Storage
   - Load balancers
2. **Software cost:** To build VDI solutions, a lot of software requirements need to be fulfilled. These include procuring Hypervisor license, VDI license, application virtualization, monitoring software license, remote access (client) software license, server OS license for infrastructure, etc.
3. **Hardware and software maintenance cost and annual operational expense:** This includes the annual maintenance cost that the customer has to pay for hardware and software maintenance to the Original Equipment Manufacturer (OEM) along with the cost of power consumption, data center hosting costs, etc.
4. **Desktop operation cost:** This is the cost of services that the customer has to pay to keep the desktop environment up and running. This includes Level-1, -2 and -3 support provided to end users in the case of desktop-related problems. This also includes cost of desktop management and security.

It is important to note that based on current level of technology maturity, the business benefits of a VDI solution are also dependent on following key factors that have to be taken into consideration before building a VDI solution:

1. User profiles
2. Type of desktop images
3. Environment factors

C.2.1 User Profiles

Profiles of people who use PCs also vary on the basis of the following factors:

1. **Movement – mobility of users and how do they need to move while working:** This can be fixed point, multiple, international, or national location.
2. **Independence – the autonomy users have in defining how they use corporate resources:** Users may or may not have control over data.
3. **Application consumption – what the users need to do and how they need to do:** Users may need forms, Web pages, rich media, etc. and see if the usage is view, listen, respond, etc.
4. **Collaboration – how often the user needs to work with other groups:** This factor determines how the user needs to collaborate and the level of collaboration required. This collaboration may be one-to-one or one-in-a-synchronous or one-in-an-asynchronous mode.

Currently, VDI technology is in a stage of early adoption and hence is lacking in standardization of roles. However, it still helps in visualizing few of them and provides clarity. Following are some broader definitions of the user profiles (Figure C.2). Please remember that this is an indicative segmentation and we can organize and name them differently based on the customer requirements.

1. **Task Workers:** Factory workers, bank tellers, retail outlet or credit card call center executives, etc. This is a typical scenario for desktop used in BPOs where mobility is less, there is a very limited requirement of independence, defined application consumption is involved; and there is a need for higher collaboration.
2. **Knowledge Workers:** Finance, operations, marketing, administration, etc. In this scenario, the mobility is medium; higher levels of independence and application consumption are needed; and there is medium need for collaboration.
3. **External Contractors:** Offshore workers, outsourcing contractor, branch offices, etc. In this case, the mobility is higher as desktop environment has to be accessed from distant locations; medium level of independence is required; defined application consumption is involved; and there is higher need for collaboration.

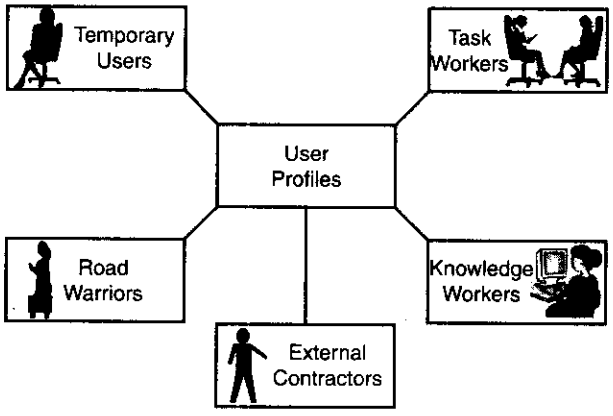

FIGURE C.2 Various user profiles

4. **Road Warriors:** Sales executives, field services, etc. In this case, mobility is maximum; there is medium level of independence and usage of application consumption, and also medium need for collaboration.
5. **Temporary Users:** University labs, training center, Internet cafe, etc. In this scenario, the mobility is medium; the level of independence is lower; the need for application consumption is medium; and the need for collaboration is limited.

C.2.2 Types of Desktop Images

The different desktop images can be classified into the following two types, shown in Figure C.3.

1. **On-Request Images:** These are images where a single copy of OS is stored in the data center and serves multiple user images. Images are composed on request when users log on by combining the OS with the appropriate set of applications. This is achieved by using different technologies from different VDI software vendors, but leads to similar results. The main advantage of this implementation scenario over the other two is that it substantially reduces storage requirements and simplifies management of the OS. Different degrees of personalization of the desktop are available for these scenarios today; however, for advanced features, third-party tools are required.
2. **Personalized Images:** These are the images where users are associated with their own unique images. User image is exactly as it was on physical PC, but it is now placed in a VM and runs on a server in the data center. Each image includes a full copy of the OS, applications, and user's personal data. Personalized images are one of the first VDIs deployed. The advantage of this option is that it can accommodate users that require full personalization of the desktop. The disadvantage is that it is typically also the most expensive scenario, as it requires a large infrastructure build out to accommodate personalization of all users.

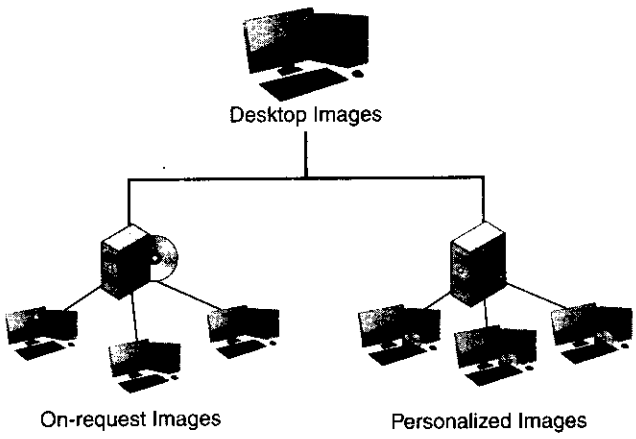

FIGURE C.3 Types of desktop images

C.2.3 *Environment Factors*

These factors are based on the environment in which the desktop will be used (Figure C.4).

1. **Concurrency:** These are the number of users expected to be concurrent at a particular period of time. For example, in an organization there may be typically x percent always logged in and using the PC environment. This factor depends on the network usage and design of the VDI solution. In the current scenario, we can imagine that if task workers in a BPO come in shifts and the organization has three shifts, then concurrency will be one-third of the total number of users.
2. **Latency:** Latency is the time difference between when the user provides the input and the time when the VDI environment sends the response. There is expectation from user regarding the latency factor in PC environment. In a traditional PC environment, the computing capacity is closest to user whereas in a VDI solution there is centralized computation from the data center. This factor can be experienced in some of the environments where applications are hosted at a different location and when we access them from the PC environment, there is some difference in the time response. To avoid unacceptable levels of latency, we have to put extra hardware as WAN accelerators in VDI solution.
3. **Software Assurance:** Most PC environments use Microsoft OS. If an organization is a part of Microsoft Software Assurance offer, then it does not have to buy Windows.
4. **OS Migration:** Windows 7 is the latest OS from Microsoft, and most of the organizations are in the planning phase for migrating to this OS. Support for Microsoft XP is going to end in mid-2014. Hence, as part of transitioning to a VDI solution, organizations can go directly to Windows 7 environment and save cost on Windows 7 migration.

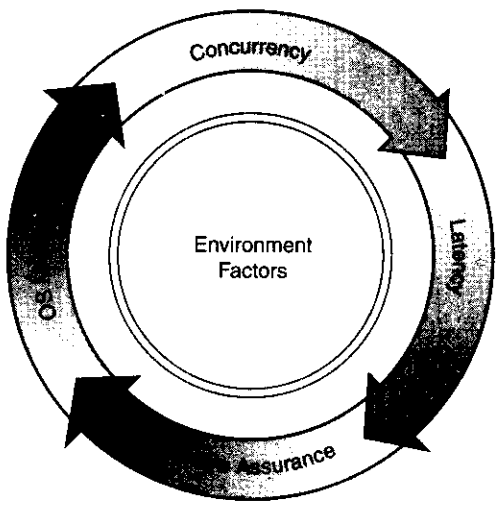

FIGURE C.4 Various environment factors

C.3 CASE STUDY

Let us assume that we want to compare the traditional desktop to a VDI solution for an organization having 1000 users. Also, let us assume that they have Software Assurance, and hence, do not need to purchase extra Windows VDA license.

C.3.1 Assumptions

The following assumptions have been made:

1. The standard cost of a desktop is USD 900.
2. The standard cost of a laptop is USD 1500. This amounts to 10–15 percent of the user base.
3. The standard cost of the thin-client hardware used in VDI solution is USD 200.
4. Citrix Xen Server is used as Hypervisor.
5. Citrix Xen Desktop is used as the Desktop Virtualization solution.
6. Windows 7 deployment cost is included as part of maintenance and annual cost as this has been the general acceptance.
7. All virtual desktops are standard desktops with 1.5–2 GB of memory.
8. Desktop operating cost includes the entire PC lifecycle management.
9. Calculations are only used for direct cost.
10. Indirect costs like the cost of user training, disruption in training, etc. are not considered as these would vary a lot from organization to organization.
11. Cost savings owing to a longer technical refresh cycle of 5–6 years for thin client vis-à-vis 3–4 years for desktop are not being considered.
12. TCO is compared over a period of three years between both the scenarios.
13. 50 percent task workers and 50 percent knowledge workers are considered for designing of solutions.
14. On-request image has been designed with this solution.
15. Design is for an on-premise cloud implementation.

Figure C.5 shows the breakup of cost for a traditional PC solution across various cost buckets.

From Figure C.5, it is evident that most part of cost is spent towards desktop operations cost. Next are the hardware cost and annual maintenance cost. We are assuming similar cost spent in traditional desktop scenario as base; hence no additional costs are assigned as spent

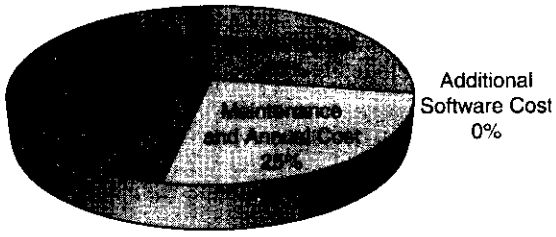

FIGURE C.5 Percentage breakup of different cost factors for traditional PC solution

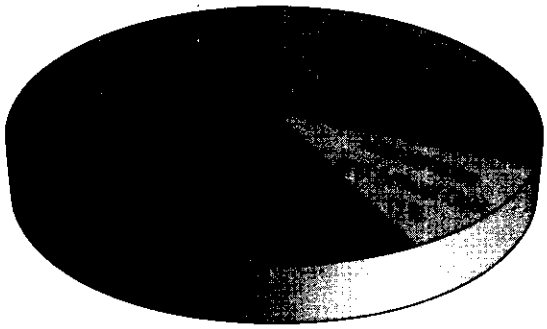

FIGURE C.6 Percentage breakup of different cost factors for VDI solution

on the software cost. Above is an optimized scenario where desktop management is an evolved function with proper desktop management function in place. Hence, the number of service requests, incidents, and changes per desktop are controlled. Cost of energy spent per desktop is also included as a part of hardware cost.

Figure C.6 depicts the percentage cost breakup for VDI solutions. This also shows that desktop operation cost remains the largest component. Complexities and number of service request, change request, and incidents reduce for desktop management service in VDI solution but the overall impact on total cost still remains the maximum. Impact of hardware cost on the VDI solution comes the next owing to extra capacity that needs to be built into the data centers for computing, storage, network, load balancing, security, etc. There is an additional component of software that is added on top of the baseline software for traditional PCs. This software includes the software that is used for server virtualization, desktop virtualization, application virtualization, session virtualization, etc. Impact of maintenance and annual cost is the least. Figure C.7 shows the comparison between VDI and traditional desktop services. This provides an idea on the impact of several solutions over the cost function. Hence, desktop operation cost remains

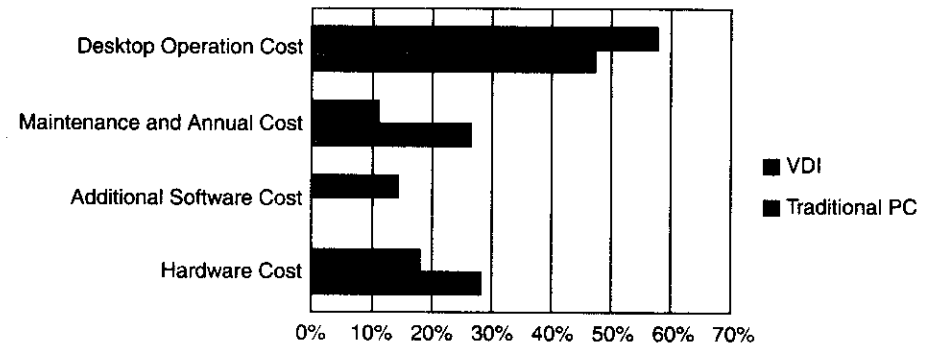

FIGURE C.7 Comparison between different cost factors of traditional PC and VDI solution as percentage of total cost

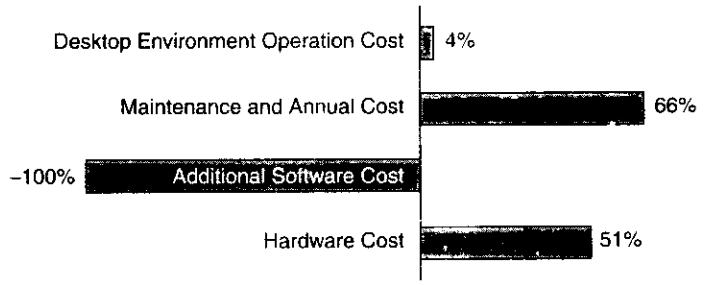

FIGURE C.8 Percent savings across various cost buckets for adopting VDI soltuions

a clear winner. Lot of organizations are struggling to keep this in control and are deploying various methodologies like offshoring, automation of low priority tickets, tool-based desktop management, etc. to keep this cost in control. There is a difference in the impact to maintenance and annual cost, need for additional software and hardware cost owing to the centralization of desktop solution and session, application, and desktop virtualization. Difference in impact would have been more significant if we would have taken this analysis over period of six years; and hence included the two technical refresh cycle of desktop but only one for thin clients. Figure C.8 shows the comparison of percentage savings in both the solutions across various cost buckets.

There are minor savings in the desktop environment operations cost, but again, we have mostly averaged the impact as there are not many data points to substantiate the reduction in incidents and service cost. This savings towards desktop service operations will become more as there are technology advancements with multiple implementations of VDI in complex and multisite locations. There are significant savings in maintenance and annual cost now as the centralized implementation leads to power savings and reduction in the number of devices and components. A traditional desktop solution does not need any additional software; hence, there is a big investment in VDI solutions towards software licenses. We also save on hardware investment on VDI implementation, which is evident owing to the difference in cost between traditional 'thickclient' desktop vis-à-vis 'thinclient' approach for VDI solutions.

C.3.2 Conclusions

On further analysis, overall for a period of three years, there has been a reduction in TCO of **15–20%** if we implement a VDI solution over a traditional desktop solution. These savings can be dependent on the factors that are mentioned under user profiles, type of desktop images and environment factors. The current calculation is based on theoretical modeling done and takes industry-wide accepted standards for cost calculations into consideration.

Organizations should adopt VDI solutions not only for TCO reduction but also for functional benefits like better security, manageability, efficient scaling, faster provisioning, and quicker respense during DR and Business Continuity Planning (BCP) scenarios, providing location independence to work force, and so on.

Business case of opting for VDI solutions becomes stronger if we go for the same in following instances:

1. **Greenfield set-up:** Currently, in emerging countries, many organizations are expanding or setting up desktop solutions. In these scenarios, if the CIOs opt for VDI solutions, then they can lead the technology adoption with major TCO savings.
2. **Technical refresh:** Most of the organizations have to go for technical refresh programs every three years. It will be beneficial if the VDI implementation becomes a part of these programs as it can immediately show major savings.
3. **Diverse and distributed work force:** Retail banks' work force is distributed across multiple locations and the need for security and centralization is higher. VDI implementations can provide an immediate relief with significant improvements in security besides centralizing the desktop functions.

C.4 SUMMARY

VDI solution is a revolution in desktop environment. Mostly, organizations have VDI on their radar and will eventually adopt the same. It is only a matter of time before we will think of our own dedicated desktop in the same way as we remember our hand-written postal mails now.

Index

P

R

S

T

U

V

W

X

Z