UNIVERSITY OF OSLO
Department of Informatics

A Technical Study Of Charge back And Monitoring Systems In Virtual Environment

Kashif Shahzad

Network and System Administration

Oslo University College

May 19, 2008

A Technical Study Of Charge back And Monitoring Systems In Virtual Environment

Kashif Shahzad

Network and System Administration
Oslo University College

May 19, 2008

Abstract

In the recent years the shared services concept has become an integral part of business. These shared services can be in the form of information technology, engineering and lot more. Service providers spent huge amounts of money to build an infrastructure that can provide efficient and valued services to the customers. In IT business these services varies from providing basic consultancy and managing the IT operations of the customers to running high priority business processes,(online banking). Customers of these services pay for these services, so a mechanism of resource usage metering is required to accurately charge the users and at the same time a monitoring mechanism is required to have a check on the services being provided to the customers for any resource contention and service degradation and future capacity planning. If a service provider is unable to develop an accurate charge back and monitoring mechanism then the equation of service provider and customer becomes a point of frustration for both sides. charge back and monitoring systems developed for physical environment are not capable to measure the resource usage in virtual environment because in virtual environment (Z/VM) resources are shared between users and it becomes difficult to measure the resource usage by a specific user. Until now a few tools have been developed that provides efficient resource metering and monitoring in virtual environment (Z/VM) but every business has its own requirements and system setup so mostly these tools need some customizations to fit into the business. This work mainly concentrated on what kind of resource utilization data is available on Z/VM and on LINUX guests running on Z/VM to effectively charge the customers running there guest Linux Operating systems in virtual environment (Z/VM based) and to monitor the cpu and memory utilization to check whether the estimate of memory allocation for linux guests running different applications made by system (PWSS) is a good estimate or require some optimizations. Because memory utilization is considered more expensive in virtual environment in the context of system performance. The study also includes a comparison between this technique of charge back and some commercial products from IBM and CA (Computer Associates) that provides charge back and monitoring facility in Z/VM based virtual environment, and provides some benefits of this work in the proposed environment.

Acknowledgments

First of all I would like to thank the Oslo University College Administration and Admission comity for allowing me to upgrade my knowledge through this masters degree.
Completing this master degree is a wonderful and exciting experience.
I am thankful to my supervisor Tore M. Jonassen for his technical help in the project.
I feel proud to be a student of Mark Burgess. His critics and suggestions were very fruitful throughout the degree.
I would like to thank Mr. Per Fremstad from IBM who provided me a great opportunity to work on IBM platform.
I am specially thankful to Kyrre Begnum for his technical support throughout the project. His help and suggestions keep me up when i was totally down during the project.
I am grateful to all the classmates for having such a nice and friendly environment throughout the degree And special thanks to Ebenezer Painsil for his help and fruitful discussions during the degree.
Finally i am thankful to my wife whose love and great moral support has kept my spirits high to complete this degree in time. Being a teacher son i have one thing in my blood and that is continues updating of my knowledge. I am thankful to my father for pushing me to go further and further for knowledge absorption.

List Of Abbreviations Used

BIRT
Business Intelligent Reporting tool.
CMS
Conversational Monitor System.
DASD
Direct Access Storage Device.
DISKACNT
A virtual machine that records accounting data.
ISFC
Inter-System Facility for Communications.
ISC
Integrated Solutions console.
ISS
Internet Information Services.
IUCV
Inter-User Communication Vehicle
JDBC
Java Database Connectivity.
LPAR
Logical Partition.
MONDCSS
Monitor Discontiguous saved segment.
PWSS
Projected Working Set Size.
RMF
Resource Measurement Facility
TUAM
Tivoli Usage And Accounting Manager.
VMDBK
Virtual Machine Definition Block.
VCNA
VTAM Communications Network Applications.
DIAGNOSE X'4C'
Allows the user to examine host storage.
XAUTOLOG
Command to log on another user virtual machine automatically.

Contents

List of Figures

Chapter 1

Introduction

Virtual technology is getting more and more popular and dominating the IT business due to reductions in the IT expenses. Instead of buying separate hardware for individual purposes a single powerful hardware unit can serve these different purposes using virtualization technology. Virtualization provides business users all the required IT resources and improves resource utilization while slowing down infrastructure growth. Instead of all these benefits most of the organizations are unable to precisely estimate the costs of different service provisions to the users because it is difficult to measure resource utilization by specific user of virtualized shared resources (Z/VM). Traditional tools used for utilization metering and billing for physical environment do not help in virtual environment [1]. Some well known IT organizations like IBM, the pioneer of the virtualization technology and Computer Associates have developed some charge back and utilization reporting tools to precisely measure the costs and provides performance monitoring to identify the causes of critical performance problems in an effort to satisfy the customers of IT services in virtual environment. But different businesses have different requirements and setups not able to implement such products for charge back. For example the cost of these products and the complexity involved in the use of these tools is a problem. Trained employees are required to use these tools that ultimately increases the operational costs of the overall business. Some tools are unable to provide historical data for detailed analysis of the resource utilization like Performance toolkit from IBM. This work is en effort toward the goal of finding a better and flexible solution for charge back in virtual environment (Z/VM based) by analyzing what kind of resource utilization data is available on the Z/VM side and what kind of data is available on the host operating system (Linux). So that a complete picture of the user resource utilization can be drawn for charging the user and what kind of extra information could be drawn for system administration, from such data.

1.1 Charge Back and Monitoring system

Charge back and monitoring system is a Method of allocating costs to end user departments based on the information services rendered and information system resource utilized and to monitor the resources being utilized to find any performance related problems. A charge back system is an accounting process to monitor resources and a pricing process to interface with users [19]. Charge back and monitoring systems bring management into control of computer resources and fulfill more than the narrow function of simply allocating computer costs [2]. Charge back and monitoring systems are the key to effectively manage a data center. These systems should be designed to accurately and precisely calculate the costs of computing facilities to the users. An efficient charge back and monitoring system enables the system administrator to take timely decisions for efficient use of IT resources.

1.2 Need of a charge back and monitoring system

The motive behind running every business is basically profit. And all the businesses require continues improvements to cope with changing global business approaches. In case of shared services centers that provide IT services to the customers, they require huge investments to setup such a shared service center and have to continually invest for better service provisions. So to meet the business (shared service center) goal of making profit and to extract the set up and running costs and to meet the SLA an efficient charge back and monitoring system is required. According to Accenture consulting company the basic goal of charge back and monitoring is workforce reduction. Without charge backs the overall cost of back office financial processing may actually increase [3].

1.3 Charge back approaches in Virtual Environment

Different approaches or models exist for charge back systems in virtual environment. The models depend upon the business policy, for example whether business is running cost-plus policy, fixed revenue and fixed revenue with predefined range. In cost-plus policy the main goal of business is to squeeze every bit of profit out of existing infrastructure and resources. In fixed revenue the customers are charged a fixed amount for services provided. In fixed revenue with predefined range the customers are given a range of resource utilization and a fixed amount is charged for this range utilization but as the utilization of resources go beyond this range, pricing is adjusted for this increased usage. These policies have severe impact on the models used for charge back in Virtual environment. Most commonly used models are [4]:

- Transaction Based

  Price is based on the volume of the transactions and it is in favor of service provider if it has any cost advantage. But the service providers revenues fluctuate heavily due to the changing demands of the customers.

- Resource Based

> Resource based charge back model seems fair to the customers because customers have the sense of satisfaction that he is being charged for what he has used as a service and what resources are available to them as dedicated resources. But the model is difficult due to the complexity involved in gathering the resource utilization and sometimes the pricing seems less transparent because price is derived from different resource factors.

The work in this project is based on resource usage. The idea behind this model is that if the total costs of the service provider (installation, management, and maintenance costs) are allocated equally among the users it may be unfair in the sense that a user who is running tasks that are cpu intensive and low I/O intensive then this user must share the major cost as compared to the other users that run mixed tasks. To increase the level of fairness in the costing scheme the resource utilization based costing is considered best in shared resource environment both by service provider and customer.

1.4 System Performance Concerns

Sometimes the operating systems fail to provide the expected level of performance. There could be many reasons for such failures like hardware and software problems, workload changes, under estimation and allocation of resources, lack of tuning after workload changes, resource over commitment, wrong configurations and many more. Detection of these problems is the first performance concern and the second performance concern is to find the cause of these problems and to fix the cause. Finally the fix must be verified.

The performance of any operating system is dependent on factors such as the hardware on which it is running, Load nature that the operating system has to handle, Peak time user count using the system and the parameter settings for the system [5].
Z/VM 5.3 was released in 2007. It included the support for latest developments in hardware and improvements for guest manageability. Performance monitoring of Z/VM can be used to improve performance but before monitoring the most important thing to consider is:
what parameters we are going to monitor.
Which parameters can give most gain in the performance. Here comes the question of what we mean by good performance. How can we analyze the performance data to find out any bottlenecks in the system. The answers to these questions help to solve performance monitoring problems in a complex environment. And to answer these questions performance data is required to effectively diagnose the performance of the system.

1.5 Charge back and monitoring System design

An efficient charge back and monitoring system is one which satisfies both service provider and customers so during the design phase of such systems both sides of the equation should be kept in mind because if one of the sides of the business becomes disturbed the whole business process is disturbed. There are certain challenges faced during the design phase. These challenges include [3]:

- Will the service provider charge for all the services or just specific ones?
- Will charges be fixed or variable in nature? what will affect this?
- How the specific services and associated charges will be defined?
- Will a start-up grace period be granted where the services will not be charged?
- How will costs be allocated:
  a. By business size.
  b. Transaction volumes.
  c. complexity of tasks.
  d. Time required to complete the tasks.
- How will charges be communicated to the business units in terms of timing, reporting mechanisms and so on.
- How can a win-win seller/buyer relationship be best established.

So while designing following points should be kept in mind.

- Mode of operation of the corporate.
- Estimate and calculate the cost table for different services. This include:
  a. Service provider personnel.
  b. Service maintenance.
  c. Development costs (establishing the business)
- Define the model.
  Consider the customers and business perspectives.
  a. Charge back model must be understandable by users. How calculations are done for charging.
  b. Charging must be clear and Justifiable.
  c. Charges must be controllable. Users can control the charges.
  d. Charges must be reproducible (Differential and steady pricing).
  e. Rates stability. Should change according to technology improvements.
- External bench marking approach.
- Align the charge back approach with SLA.

Chapter 2

Motivation

Business owners need to know they are getting their fair share of of the pie, And business Management need to know the performance of the existing business resources for future planing and decisions.

Virtualization has introduced the IT world with a new revolution and reduced the IT costs a lot by replacing large number of physical hardware units with a single powerful unit hosting large number of logical systems. Managing lower number of physical units is easier. Although Virtualization has proved its benefits but still there is a group of professionals who think little bit differently. They argue that it is difficult to measure the resources utilization in virtual environment specifically in shared resource environment (Z/VM) where resources are shared among different users. And on the basis of this accurate charge back is difficult in virtual environment for shared service centers providing services to the customers. At the same time users are not able to understand fully how the calculations for costing them have been done and this frustration in the customer community is a hinderance in the popularity of virtualization (Z/VM).

The data required for IT and business decision making exists at various places within the virtual environment, and is difficult to capture and consolidate. Traditional tools (for physical environment) are not effective because they are not designed to manage the frequently changing allocations and configurations that are hallmarks of virtualization, nor they are able to capture important data specific to virtual machines such as CPU processing time and storage consumed by specific users. Although many vendors have developed special tools to monitor and measure the resource utilization and create a charge back report for users. But these tools are complex to use and require lot of configuration both on hosting and hosted systems. Trained users are required to use these tools and to produce charge back reports. So these charge back tools do not fulfil the goal of work force reduction.

These charge back problems have motivated the author a lot to work in this area and this concept project has tried to outline the problems of:

- What data is available for charge back and monitoring.
- How utilization data can be collected.
- What extra values can be drawn from this collected data.

By getting this information can help enterprizes bring optimization to virtualization by helping guide decisions regarding outsourcing, upgrades, new purchases, leases and service models. Many organizations do not use charge back data to actually charge business units for the IT resources they consume. Instead, they find the data valuable for optimizing how resources are deployed and to guide forward-looking decisions about purchases, leases, licenses, warranties and other expenditures. This ability is just as valuable for virtual environments. Detailed utilization and performance data helps IT administrators decide the value of virtualizing various assets, and measure the value of virtualization efforts. The data may reveal excess capacity in some areas, or could lead to recommendations for some business units to have their own dedicated resources instead of using shared ones. Utilization data not only helps measure true IT costs, but also the relative value of different business units or initiatives relative to the IT costs they incur.

Chapter 3

Background and literature

3.1 Virtualization

The word Virtual means existing in essence or effect though not in actual fact and Virtualization means creating an object that does not exist but we can see it. Virtualization is the ability for a computer system to share resources so that one physical server can act as many virtual servers. According to Jonathan Eunice, Virtualization is the process of presenting computing resources in ways that users and applications can easily get value out of them, rather than presenting them in a way dictated by their implementation, geographic location, or physical packaging. In other words, it provides a logical rather than physical view of data, computing power, storage capacity, and other resources [6]. Virtualization creates an external interface that hides the underlying implementation. One important thing to note is that splitting a single physical entity into multiple virtual entities is not the only method of virtualization. But reverse is also a form of Virtualization like combining multiple physical entities to act as a single, larger entity is also a form of virtualization, and grid computing is an example of this kind of virtualization [6].

3.2 Benefits of Virtualization

IT systems are growing larger and larger and becoming more and more complex. This complexity of IT systems require more people to manage them and therefore the cost of administering IT systems are growing faster as compared to the cost of new Hardware for these systems. But the primary goal of any business management is to shrink the costs, better utilization of the existing infrastructure and increase the revenues. Virtualization technology plays an important role in meeting these requirements. Other benefits of virtualization includes [6]:

- Efficient Resource Utilization
  It provides dynamic sharing of physical resources thus resulting in higher resource utilization where the nature of the workload is variable and average needs are much less.

- Lower Administering Costs
  Reduction in the number of physical resources has increased the productivity of the office staff a great. Automated management tasks has lowered the administration costs a lot.

- Higher Availability
  Physical resources can be removed, upgraded or can be changed without affecting the users of the system.

- Consolidation
  Multiple operating systems and applications can be supported on the same physical system. Similarly computing resources can be treated as a pool and can be allocated to virtual machines in a controlled manner.

- Security and Isolation
  Separation and partitioning of physical resources in a logical way is the key factor of virtualization success. Each virtual machine is isolated from the other virtual machines and a crash of one VM does not affect the other ones. Similarly there is no data leakage from one vm to the other.

- Scalability
  Virtual resource can be defined much less or much larger then individual physical resources. Physical adjustments are not required.

- Interoperability
  Virtualization provides backward compatibility with the protocols that are not supported by the underlying hardware.

3.3 Z/VM Architecture

As this work is based on Z/VM virtualized environment a brief introduction to Z/VM architecture will help the audience to understand the methodology used in this project.

Z/VM is an operating system for the IBM System Z platform that provides a highly flexible test and production environment. The Z/VM implementation of IBM virtualization technology provides the capability to run full-function operating systems such as Linux on System Z, Z/OS, and others as guests of Z/VM.

Z/VM supports 64-bit IBM Z/Architecture guests and 31-bit IBM Enterprize Systems Architecture/390 guests [6].

Each user experiences an individual working environment known as a virtual machine. The virtual machine simulates the existence of a dedicated real machine, including processor functions, memory, networking, and input/output (I/O) resources. Operating systems and application programs can run in virtual machines as guests. One can run multiple Linux and z/OS images on the same z/VM system that is also supporting various applications and end

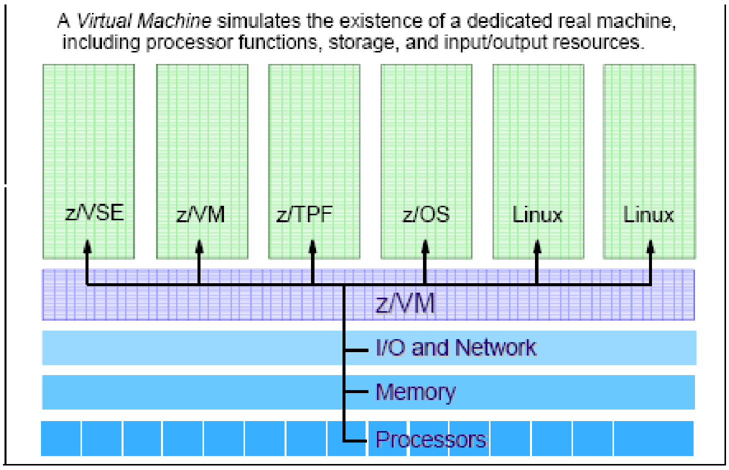

Figure 3.1: z/vm basic architecture [6]

users. In this way, development, testing, and production environments can share a single physical computer.

3.3.1 Operating Environments

Three different operating environments exist in Z Architecture: Native, LPAR, As a guest under Z/VM.
In native mode the entire system is used as a single system. In new mainframes this mode is not supported. In LPAR mode the system is logically divided into multiple partitions. This is the most common way of hardware partitioning used in mainframes. As a guest under Z/VM is a software level partitioning. Virtual machines are created on top of the Z/VM system.

3.3.2 First level versus second level guest system

First level system is the base operating system installed directly on the hardware and a second level system is the operating system installed on top of the base operating system(Z/VM). Z/VM is the first level operating system that directly sits on the real hardware and the second level systems are the guest systems and are virtualized [6].

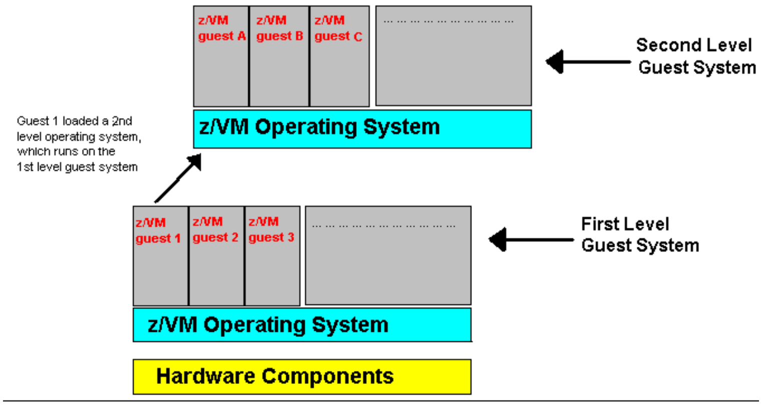

Figure 3.2: first level vs second level operating systems [6]

3.3.3 Z/VM Components

Z/VM has following components [6]:

- Control Program (CP)
- Conversational Monitor System (CMS)
- Transmission Control Protocol/Internet Protocol For Z/VM
- Advanced Program to Program Communication/Virtual Machine
- Dump Viewing Facility
- Group Control System
- Hardware Configuration Definition (HCD) and Hardware Configuration Manager (HCM) for z/VM
- Language Environment
- Open Systems Adapter Support Facility (OSA/SF)
- Restructured Extended Executor/Virtual Machine (REXX/VM)
- Transparent Services Access Facility (TSAF)
- Virtual Machine Serviceability Enhancements Staged/Extended (VMSES/E)

3.4 Advent Of Charge Back Systems

The idea of charge back in IT industry developed in the mainframe era. Mainframes were very expensive and to buy a mainframe for small to medium sized business was a problem. So the Businesses having these mainframes began providing the computing services to the small businesses to cope with the operational costs and to better utilize the resources of these expensive mainframes. At that time Information systems staff and Accounting staff developed some charge back models to assign the costs to the customers of computing services. These charge back systems evolved with the passage of time and then came the era of server based and decentralized IT architecture that deemphasized the charge back systems as most of the businesses acquired their own servers. Todays virtual environments actually combine the concepts of shared resources (hardware) and dedicated resources (the virtual machines themselves). Due to this fact most of the preestablished charge back principles and best practises used for mainframes are applicable to virtual environment today.

3.5 Importance of Monitoring System

Z/VM performance monitoring has two aspects where one can focus analysis on performance monitored data.

1. Reactive.
   In this approach the administrator react to a reported problem by the user.

2. Predictive.
   In this approach the administrator analyze the performance data for the prediction of any resource constraint or for future planning about the resource upgrading.

3.5.1 Reactive Approach

As an administrator of a system if the system developed a performance problem. What could be the best way of knowing that problem. Mostly administrators came to know about a problem when the users contact with the administrator that they are facing problems. But this approach is not efficient because that system problem degrade system performance and customers satisfaction, there is no chance for the administrator to fix the problem before it frustrates the customers. This approach (reactive) is an integral part of the monitoring systems because sometimes performance problems not detected by the predictive approach are noticed by the users of the system like a misconfigured application is not responding and the user problem reporting enables the administrator to check the cause of the problem and then fix it.

3.5.2 Predictive Approach

In this approach mostly the problems related to the system resources are diagnosed and fixed well in advance before they frustrate the user. In this approach the administrator has a complete picture of the system functionality and He is in a position to recognize the problem before it degrades performance. Further more this monitoring approach enable the administrators to take decisions well ahead about system improvements and capacity planning by examining the system utilization records produced by the monitoring system. For example if a user shows high CPU utilization over a prolonged period of time then this user can be allocated more CPU resources or in virtual shared environment its priority can be increased so that it can have more CPU share. This is not the only way we can take advantage of monitoring systems. There are many ways of monitoring and we can exploit the advantages of this technique. In this study monitoring data was collected by keeping in mind the predictive approach. The monitored data was used to know the estimate of the memory allocations to the guest operating systems by the system (PWSS) running different applications so that the paging activity of the applications can be kept very low if they are allocated low memory, because paging activity affects the performance when paging is done to slow speed DASD.

3.5.3 Performance parameters

The question arises here what parameters are considered important for performance?. And how they affect the performance of the system?. Performance parameters can include:

1. System capacity.
2. Reliability.
3. Response time.
4. Throughput.
5. Number of users supported.
6. Device utilization level.

This list presents only a few of the parameters for performance of Z/VM. These parameters are affected by the resources available on the system. So while defining the performance of a system one must keep in mind:
What kind of applications are running on the system.
Is hardware sized correctly, according to the workload requirements.
Paging space is sufficient for the guests.
Are there specific users that cause too much paging.
Is I/O configured correctly.

So keeping in mind all these points performance monitoring can be planned well in advance about what resources will be monitored and what aspects or parameters of performance will be looked for.

3.6 Vendor Supplied Tools

A lot of commercial tools are available now a days. The two most commonly used tools that provide charge back and monitoring services in Z/VM environment are:

- Tivoli Usage And Accounting Manager (TUAM)
- Unicenter VM:Account

3.7 Tivoli Usage And Accounting manager (TUAM)

Tivoli usage and Accounting manager enterprize edition V7.1 is basically a tool that provides [7]:

- collection of resource usage data.
- Assigns account codes for each resource.
- Provides charge back rates for each unit.
- Reports production through a web interface.
- Platform independent reporting facility.

Moreover it produces analysis reports on the charging environment to make sure that the charges used are correct and fair. There is also a modular function in TUAM that provides a rate analysis based on IT expenditure.

3.7.1 TUAM Components

The major components used by Tuam are shown in the diagram.

- Collection

  Resource usage data collection is mostly handled by the operating systems and other applications designed for that purpose. TUAM data collectors read this data or provides access to the databases where the metered data is stored.

- Application Server

  This component provides two functions:

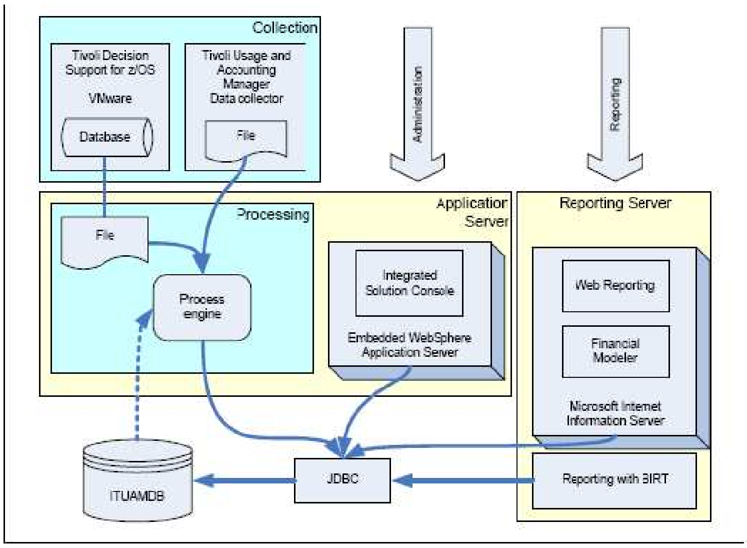

Figure 3.3: Tuam basic components overview and dependencies [7]

- Administration

  This is performed using the Integrated Solutions console (ISC). ISC is an application that runs on top of an embedded web sphere application server. This actually provides the front end for administrative work of the TUAM server.

- Processing

  Processing of the gathered data is done using the Process Engine and the TUAM integrator function. It is responsible for all data processing and data loading into the TUAM database. The JobRunner that is Java based controls the processing steps.

- Database Server

  A relational database server is required to store the administration accounting and resource usage measured data. This database is accessed using JDBC driver. This driver is required for each component that requires access to the database. TUAM database is not implemented as part of the TUAM installation. TUAM has the ability to define as many databases as required and many of them can be used as default. All these database definitions and other settings are performed from the administration server.

- Reporting Server

  All reports produced by the TUAM are generated from Database and can be stored on the filesystem for publishing and distribution. TUAM provides reports using Microsoft Report viewer under IIS or using BIRTS (business intelligent reporting tools).

3.7.2 Data Collection

Data is collected from different data sources. These data sources include [7]:

- Z/VM
- VMWare usage collector support
- Unix, Linux, Linux on system Z operating system
- Unix, Linux, Linux on system Z file system
- System i (collects all usage from System i, but the actual collector must be run from Windows)
- Tivoli Decision Support on z/OS extract (formerly the Accounting Workstation Option or IBM Tivoli Usage and Accounting Manager Enterprize Edition for z/OS)
- Generic collection (also known as Universal Collection)
- Windows disk usage
- Windows CPU processor usage

Data is collected in files and then transferred to the processing servers for data crunching, analysis and loading. Data can also be loaded remotely by using JDBC access or web services call.

3.7.3 Data Processing

TUAM processes different data sources almost similarly. The processing steps may be different depending upon the collector used. If operating systems and applications have the accounting functions built in then TUAM just uses that data for further processing. This data is converted into CSR (Common Source Resource) format before TUAM processing starts. CSR format data is summarized based on given identifiers. This summarization step is called aggregation of data. In the next step the CPU values are normalized and multiplied by the rate code. The cost table is selected and values in the cost table are used for multiplication. Data is summarized on the financial and organizational level that provides billing details, billing summary and different identifiers. The final step in the processing phase is loading the output data into TUAM database. Data duplication is prevented by a duplicate detector.

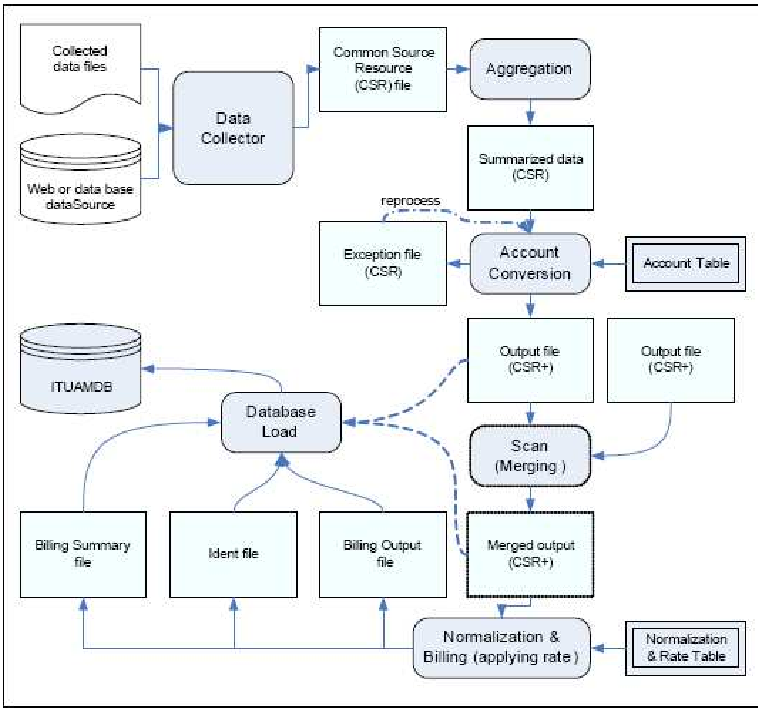

Figure 3.4: data processing overview of tuam [7]

3.7.4 Accounting Code and Rate

To identify who has used the resources and who will get the bill account code is used. This is the primary identifier for customers to be billed. Account code structure and definition is the responsibility of the organization and it should be done carefully before any data collection and processing. TUAM actually helps in the definition and designing the structure of the Accounting code. Account code is basically a string of fixed width field that define the hierarchy of the accounting break down. The fields in the accounting code are used to split the accounting string for charging different entities in the organization. TUAM defines some best practises for Account code structure definition. Rates are defined in the rates group. The definitions of the rates include format, type, conversion factor and money value for all shifts.

3.7.5 Report generation

TUAM uses two report engines for report generation:

- Microsoft web report viewer
- Business Intelligence and Reporting tools (BIRT)

Any reporting software can be used with TUAM by using SQL to generate reports directly from TUAM. Microsoft web reporting server is based on

IIS. BIRT is an open source tool. BIRTS reports need to be customized and these reports can be run from batch commands or can be published through an application server that has BIRT reporting plug-in.

Following diagram explains the billing process.

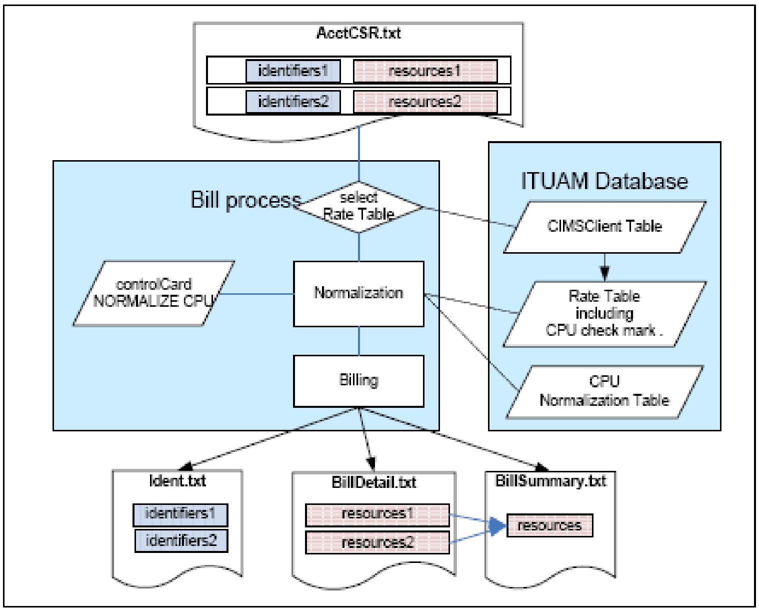

Figure 3.5: billing process overview of Taum [7]

3.7.6 Installation and Configuration

Installation and configuration of TUAM involves lot of steps before it is ready for report generation. These steps include [7]:

- Database Size Estimation
- Architectural Implications
- Administration Server Installation
- Application Server Installation verification
- Initial configuration of the Application
- Verification of the database configuration
- Report Server Installation and configuration
- Verification of the Installation of the report server

These steps require good expertise and care otherwise the required results can not be met.

3.8 Unicenter VM:Account

Like TUAM Unicenter VM:Account is a tool from Computer Associates that provides Resource accounting, reporting and capacity management for the VM environment. Unicenter VM:Account provides [8]:

- Project accounting, software-package accounting, and minidisk and Shared File System (SFS) accounting
- Online querying
- Reporting and invoicing
- Budget control facilities
- Real-time collecting, validating, processing, and reporting on VM accounting data, allowing accurate cost allocation and cost recovery
- Preservation of data integrity and protection against lost accounting data
- Workload balancing through discounts and surcharges
- An audit trail for all CMS user activity, allowing long-term trend analysis
- Full-screen maintenance of the Unicenter VM:Account project, customer, account number, and rate information tables.

Unicenter VM:Account also provides greater control over the data center by providing information like [8]:

- Monitor system resource usage and plan for system upgrades
- Track and charge for software package usage
- Charge users or organizations for resource consumption
- Allocate resource usage to projects
- Control and track project costs and budget limits

VM:Account provides up-to the minute information because it collects continuously accounting information and do real time validation and costing.

3.8.1 Information Retrieval from VM:Account

Accounting information from VM:Account can be obtained in three different ways.

- Full Screen Queries
- Reporting
- Trend Files

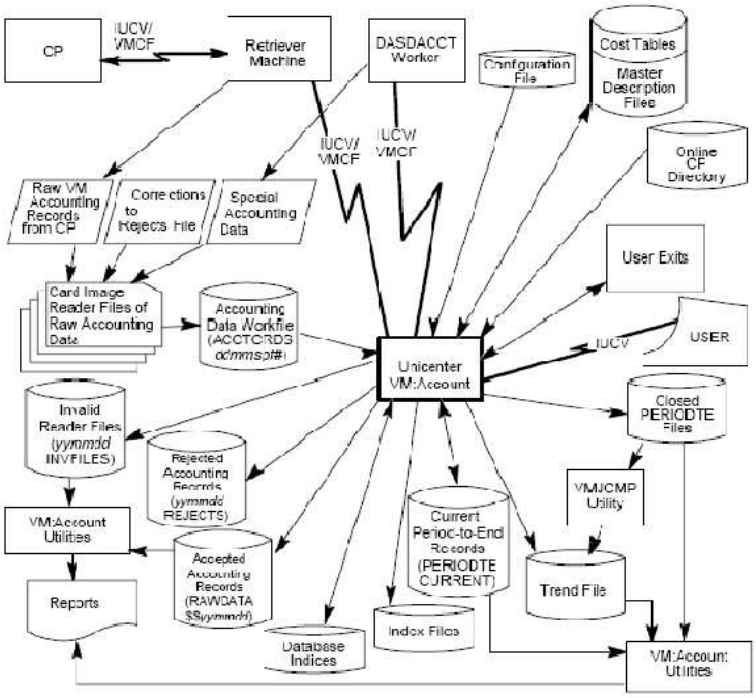

Figure 3.6: General system overview and component dependencies [8]

Query usage Selection screen contains resource usage information for specified users. Further more the type and amount of information can be specified. VM:Account supports 10 cobol reports to provide information about the resource utilization and accounting system. Reports control file provides control over the amount and type of information that should be displayed on the report. Trend files contain data collected over long period of times for analyzing long term trend in the resource utilization.

3.8.2 Data handling

VM:Account has the ability to deal with incorrect and invalid data. It uses different files to deal with invalid and incorrect data.

- Invalid Reader File
- Rejected Accounting records
- Accepted Accounting records

As the name indicates invalid reader file is used to store the accounting records coming from a users that is not authorized to send Accounting records to the VM:Account. Rejected accounting records actually comes from a valid users but are missing some information to be costed correctly. Accepted accounting records are exactly 80 bytes records and contain complete information used for costing the users.

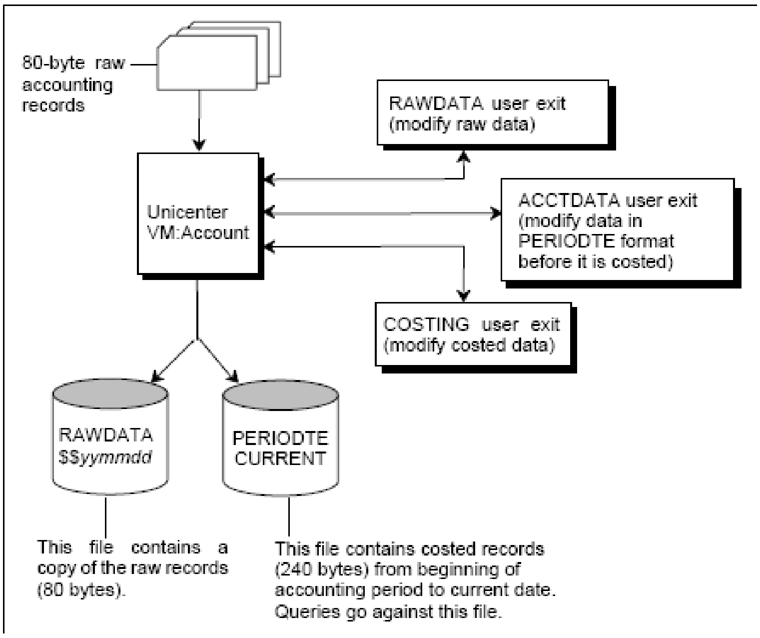

Figure 3.7: processing steps involved in VM:Account [8]

3.8.3 Accounting Structure

VM:Account provides a very elaborative and flexible to use accounting structure. It defines the accounting elements as

- Customer
- account numbers
- Projects
- Costable files

Customers are the users or organizations that are using the resources to be charged. Customers receive the invoices containing the charges for resource utilization. VM:Account uses a 12-character field Customer-ID to identify the customers. An account number of 8 characters is used in each VM user directory for each users. An account no. can have multiple userids associated with it but a user is assigned to only one account no.

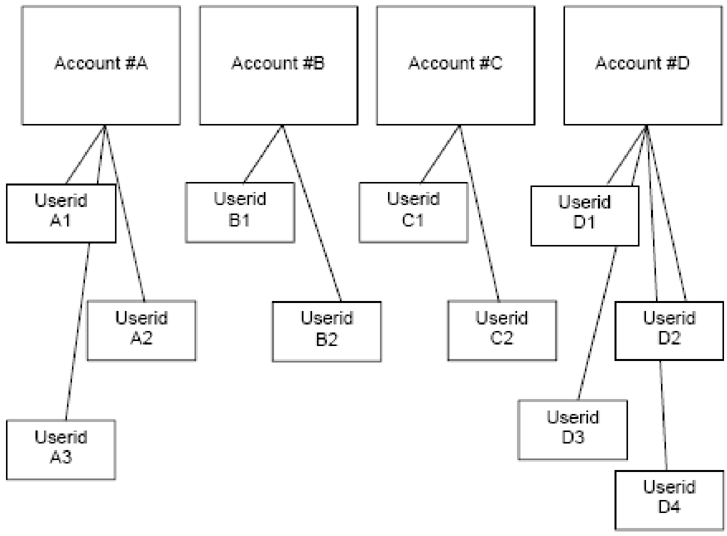

Figure 3.8: Account code design [8]

A customer id can be associated with multiple account no,s but an account no. belongs to only one customer id.

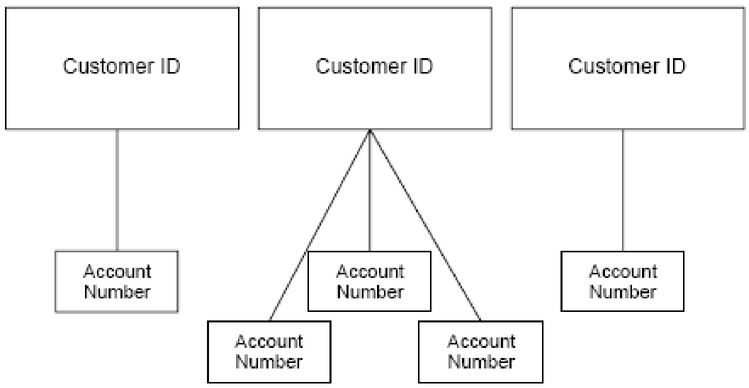

Figure 3.9: customer id definition [8]

VM:Accoun contains 1-24 characters long project information. This provides an other way to charge the users. One or more account numbers can belong to a project and an account no. can belong to more than one projects.

Users can change their projects by using PROJECT command available in the VM:Account. Costable files contain rates information for different resources that is used for calculating the charges.

Accounting structure includes:

- Charging by Account Numbers
- Charging by Project

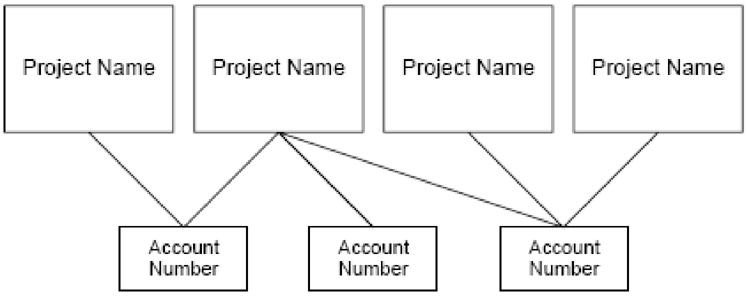

Figure 3.10: Project vise accounting definition [8]

3.8.4 Resource Charging and Differential Charging

VM:Account provides the facility to charge the users for software packages use as well as differential costing. Differential costing means charges for resource usage at peak times (during normal working hours) differ from the off time resource usage. Similarly charges for holidays can also be defined in the VM:Account for differential charging. So customers can be encouraged to use SQL/DS at evening times for extra discounts. This requires implementation of these steps into the VM:Account config file. Similarly reports for different working shifts can be created easily.

3.8.5 Security Violations Report Utility

VM:Account consists of lot of utilities that perform different functions. Among them is the VMJRSE utility that provides reports about security accounting records that are generated when illegal login or autolog attempts are made and when invalid links to the user minidisks are tried. Invalid attempts means that invalid passwords are tried. This utility can be run every day to have check on the security breaks.

3.9 Analysis

Above brief introduction to these commercial tools indicates that these tools provide lot of functionality and reports for charging and trend analysis but the complexity involved in the configuration and their use requires user training that violates the charge back systems goal of workforce reduction. Although these tools have menu based screens but to define cost tables, differential pricing and other customized reports require command based approach to configure these settings and require expertise for such tasks.

3.10 Related Work

A lot of work has been done in developing the efficient charge back and monitoring systems in the Z/VM based virtual environment. Not only IBM has provided solutions in this area but many other companies and vendors has provided such tools to deal with the difficulty involved in metering the resource utilization, that is the main hinderance in the popularity of the virtual environment. Now a days many commercial and open source tools are available that measure the resource utilization in virtual environment and produce the business intelligent reports based on the resource utilization for costing the users of the resources and finding the performance problems. But the main problem associated with these tools are the complexity involved in their use. And secondly the costs of these professional tools are too high. An Object Oriented information model for metering and accounting has been proposed by Agarwal [9]. In this model metering information is modeled into business goal of end application by defining the relationship to monitoring components from which it is derived. A paper published by a team of IBM in 2003 has done a similar work that provides a good starting point in the development of an efficient charge back and monitoring system. Actually the work done in this thesis resembles with this paper because the goal of this thesis was also to develop a concept project that can help the organization, in which this work was conducted to develop a method of collecting resource metering data for costing and producing historical data files for future trend analysis. For monitoring aspect CPU utilization and the memory utilization was chosen.

3.11 Research Question

The basic questions this work is based on are:

- What resource utilization data is available for costing the users of the system in Virtual environment (on Linux side and on Z/VM side).
- How this data can collected.
- What charging policy is best for both customer and service provider.
- What extra values can be derived from this data.

A general concept in performance monitoring in virtual environment is that both Linux side and Z/VM side resource utilization data is examined. In this project Linux side and Z/VM side CPU utilization was correlated to find dependency or relationship between them. The reason behind this approach is that in Z/VM resources are shared among the users and 100% Linux side CPU utilization does not mean that the user is using the real CPU fully, because the user gets only a percentage of the real cpu. And also sometimes the cpu utilization on the linux side is less than on the z/vm side due to the the fact that system have to do some work on behalf of the linux guest and this overhead depends upon the nature of workload the linux is performing at that time. This overhead can include page translations for the guest and some scheduling tasks.

In Z/VM the main problem for service providers is the memory allocation for new web applications. The reason behind is that some application are more memory intensive means they require all of the packages concurrently to be in the main memory before any response. If the application level data is not available to check how much memory such applications utilize then the only possibility is a guess work. Sometimes this guess work works well but mostly optimization is required to get a good estimate of the memory allocation. Over provision of memory is a waste both for customer point of view and for service provider point of view. Customers has to pay extra for non used memory and the system memory utilization is not optimal. And at the same time under estimated memory allocations leads to degraded performance in the form of delay in response time. Based on this scenario the Z/VM has a built in facility that provides an estimate for the memory required by the guests in their next run called Projected Working Set Size (WSS) based on the guests current memory usage trends during that specified interval. The question chosen in this scenario is:

Whether the system estimated Working set size is a good estimate or require further optimization.

Chapter 4

Model and Methodology

Pearson product-moment correlation was used to estimate the correlation between cpu usage on z/vm and on linux side. The choice was made because the fact is that although the linux is provided virtual resources but it is running on some physical resources in the back end. So if linux guest is running cpu intensive workload it is reflected in the physical resources as well, but in case of Z/VM environment resources are shared among users so when one linux guest is waiting for I/O in the mean time the linux is showing high cpu utilization on the linux side but the z/vm side resource becomes idle and is provided to any other user ready to run the jobs. So to check whether the Z/VM environment also has linear relationship between the cpu usage on linux side and on z/vm side the Pearson product-moment correlation was selected.

Virtual technology is getting popularity due to the benefits discussed in the previous chapter. But with the development in IT sector more and more new applications are emerging every day. Customers of the computing services want to use these new applications but for Service providers it is a problem to estimate the resources (Memory) these applications will use. Some times a guess work, works well but sometimes this guess work fails and it can lead to degraded performance. For example if a web application is more memory intensive and if the memory allocated is not sufficient then paging activity will increase considerably and the response time will degrade badly. IBM provides a way for calculating the memory allocation for different applications. This method includes allocating a minimum fixed memory defined by the vendor for that application plus an extra amount up to the level where swapping stops. In this work the Projected Working Set Size (PWSS) estimated by the system was used to know how accurate that estimate is.

4.1 Approaches for Data Collection

Two approaches exist for data collection:

1. Black box Approach
2. White box Approach

Black box approach does not provide the insight into which applications are actually using how much of the resources. It generally provides an overall external view of the system. By using this approach only the overall resource usage by a user can be measured. What is happening inside which user programmes are using more CPU and which programmes are more memory intensive is ignored in this approach. But on contrary to Black box White box approach provides a deep insight into the system into which application is using what resources and how much of these resources. Both of these approaches have their benefits and drawbacks depending upon the system one is considering for metering. If resource metering is required to charge the end customers then black box approach seems a good choice because resource usage by a specific user is required not the resource usage by different applications run by the user. But when comes the case of performance monitoring then white box seems a better approach. In Z/VM resources are shared among the users and users are assigned a weight (priority) to have a share of the resources. If a guest is running heavy CPU intensive workload like sequence alignment methods based on profile-HMM, Hidden Markov Model (Bioinformatic Algorithms) [10], then the response time of that guest can degrade if the weight assigned to that user is low. So to monitor which application is utilizing what amount of cpu cycles an insight into the system is required so White box approach seems a good choice for this scenario.

In this study only black box approach was used for data collection.

IBM has many facilities built in to the Z/VM, providing great flexibilities for recording resource utilization data. This resource utilization data is divided into two classes:

1. Accounting Data

2. Performance Data

Both of these classes of data are recorded into separate places. Accounting data can be used for charging users using the computing services of the system. This accounting data contains metered data for many different resources. Performance data provides information about the current situation of the overall system as well as the individual users of the system. Any performance problems and resource contention can be diagnosed by analyzing this data. Performance data is also actually the resource utilization data but it provides more detailed status of the system at a specified time.

4.2 Basic Charge back Model

Different charge back models exist for costing the resource utilization by individual users as well as on the project basis but to some extant all the models have the same goal of gathering data and then calculating costs to charge the users. The only differences that exist are costing tables that are purely dependant on the organization running the shared services center. Some organizations follow the fixed costs for all the resources allocated, some follow the variable prices like transaction based costing and some follow all the available costing schemes for different variables in the system. Following diagram explains the costing routines followed by most of the organizations.

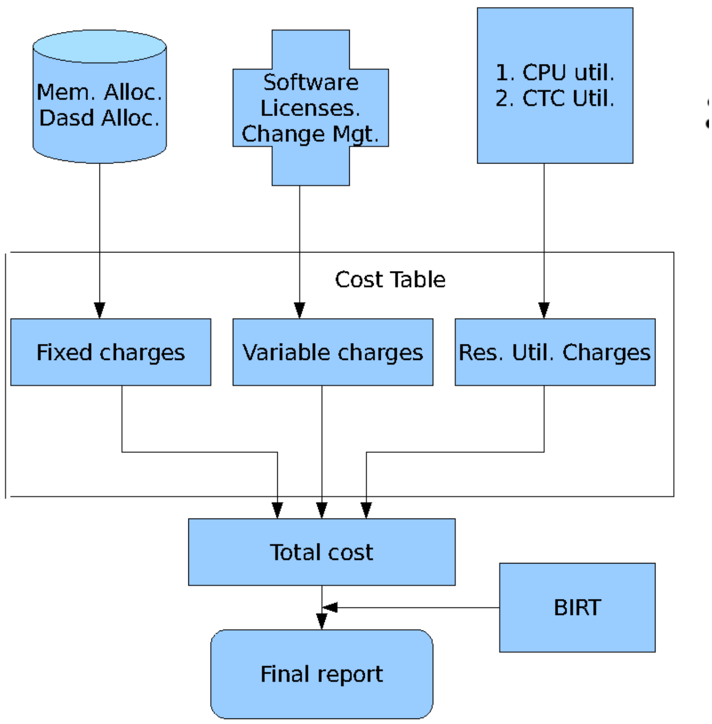

Figure 4.1: Simple costing model

According to the diagram the costs for different resources are calculated by using the costing categories, depending upon whether they are fixed price resources, variable priced or costing is done on the basis of their use. Each costing category include companies setup costs, running costs and maintenance costs. As in this study only CPU utilization was recorded, so to charge the customer of the CPU utilization the calculation can be done on the basis of how many seconds the user has used the CPU multiplied by the unit price of cpu per second usage. Where unit price comes from the costing table that depends upon the organization's costing policy providing the computing services.

The main hinderance in the popularity of Virtual technology is the true and accurate cost allocation where resources are shared among different users. The existence of many commercial charge back and monitoring tools for virtual environment has made it easy for the management to choose a tool that fits in the organization,s environment and get rid of the huge stress of cost allocations for IT resources to different customers and future decision making for capacity planing, resource upgrading and any outsourcing required by the company. But the smaller companies running Virtual environment still face the problem because they do not have the budget to buy such costly commercial tools and to hire trained personnel to run these commercial products. And secondly by hiring new trained personnel does not fulfill the goal of charge back and monitoring system which is workforce reduction.

The study was conducted in the production environment where many Linux servers of different customers were running. Although the real production environment provides a benefit of true traffic and resource utilization but there are some problems with such scenarios that it is difficult and sometimes impossible to implement the planed scenario typical for such studies. Sometimes the required variables can,t be metered and the worker has to comply with the constraints imposed by the environment.

4.3 Experimental Design And Setup

The experiment was conducted in a leading IT group in Norway, Ergo group As. The Ergo group is using IBM provided Z9 series mainframe that is running Z/VM. It is partitioned into four Logical partitions assigned one physical cpu to each LPAR and running Linux servers of many different customers. The Group is using Performance toolkit for reporting the resource usage by individual customers. Performance toolkit does not provide any facility to produce files containing historical data for trend analysis. Only reports about current data when Performance toolkit is run are produced. Performance toolkit provides the ability to analyze historical data but unable to produce historical data files. Moreover RMFPMS, is used to collect Linux side data, running on Guest Linux operating systems. This data is then used by the Performance toolkit for report generation. Performance toolkit also provides web based interface for report generation. The experiment was designed in a way not to disturb the Production environment of the Group, running many customer,s servers.

4.4 SET UP

The experimental setup for this experiment is depicted in the diagram below.

There are lot of tools available to collect resource utilization data on Linux side and on Z/VM side but in this project effort was made to use the IBM supplied services on Z/VM side and on Linux side.

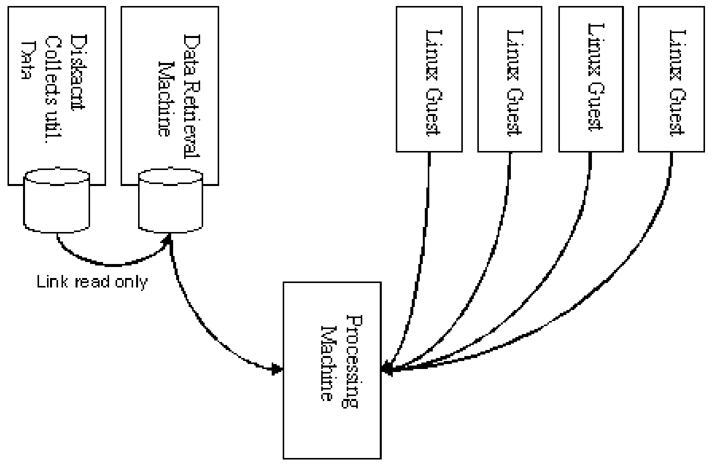

Figure 4.2: Data retrieval setup

4.5 Accounting Records

In the diagram above the DISKACNT is a virtual machine that is IBM provided user virtual machine, and comes as a default virtual machine, for collecting accounting data. More virtual machines can be defined for data collection but it requires administrator level access (class A). Z/VM has a built in service that collects Accounting data for virtual machines running on Z/VM. To collect Accounting data accounting service needs to be initialized and if it is ON CP collects resource utilization data and stores on the Disk. This data can be retrieved by the virtual machines (Service virtual machines) if the virtual machines has IUCV(*ACCOUNT) entry in the user directory. This entry enables the virtual machines to connect with the CP Accounting facility for data retrieval. The DISKACNT virtual machine was defined in a way that accounting service was ON permanently and it continuously collects accounting data for guest machines running on Z/VM. Users can be specified for which accounting records are to be collected. But in this study all users using the virtual machines were included in the Accounting list for data collection. DISKACNT,s disk is linked to the processing machine that processes the accounting records. The accounting records produced by the DISKACNT are in non printable Hexadecimal format. A Rexx script was run on the processing virtual machine that processes the accounting records and retrieves the required fields from the accounting records. The script is written in a flexible way so that with changing demands of the business usage data for other resources can be added easily with any changes in the existing script. Only the new fields of data are required to be added in the script. The following diagram explains the script flow and flexibility.

These accounting records are all 80 characters card images. Before the resource utilization data can be retrieved from the DISKACNT virtual machine a CP command ”CP ACNT ALL” is required to be run by the processing Virtual

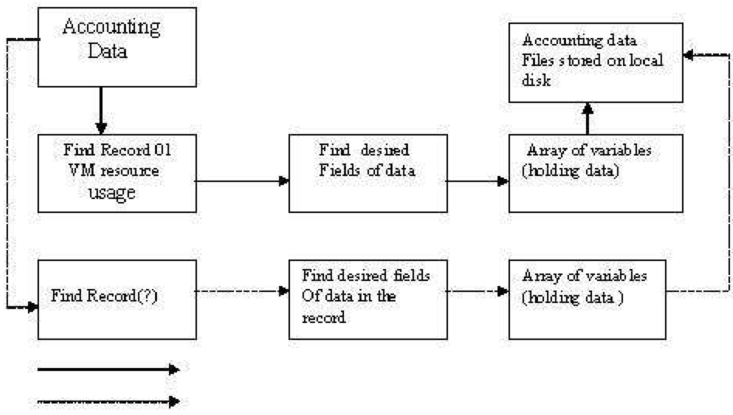

Figure 4.3: Flow chart describing the script flexibility

machine.

CP produces the following types of Accounting records [11].

- Virtual machine user records (record type 1).
- Records for devices dedicated to a virtual machine user (record type 2).
- Records for temporary disk space dedicated to virtual machine user(record type)
- Records that are written when CP detects that a user has entered enough LOG ON, AUTOLOG, XAUTOLOG, or APPCVM CONNECT invocations with an invalid password to reach or exceed an installation-defined threshold value (record type 04).
- Records that are written when CP detects that a user has successfully entered a LINK command to a protected minidisk not owned by the user (record type 05).
- Records that are written when CP detects that a user has entered enough LINK commands with an invalid password to reach or exceed an installation-defined threshold value (record type 06).
- Records generated whenever a user logs off or disconnects from a device controlled by the VCNA (record type 07).
- Records that are written when CP detects that a user has successfully entered a LINK command to a protected minidisk not owned by that users virtual machine (record type 08).
- Record type 08 is also generated when the user logs off or disconnects or when a SHUTDOWN or FORCE command causes a logged-on virtual machine to be forced off the system. Disconnected virtual machines

do not have another 08 record generated for them if they are forced off. Records generated about ISFC (record type 09).

- Records logging changes to a users privilege class (record type 0A) v Records for virtual disk in storage space (record type B)
- Records for Network Data Transmissions (record type C).
- Records generated as a result of a user-initiated DIAGNOSE X'4C' instruction (record type C0).Note: Record types 04, 05, 06, and 08 are generated only when LOGON, AUTOLOG, XAUTOLOG, LINK, and CONNECT journaling is on.

4.6 Record Selection

Records type 1 was used for charge back system because mostly CPU and Memory are considered most expensive in virtual environment. The system on which this study was conducted has customers being allocated fixed amount of memory and DASD according to the requirements of the customers and these customers are charged a fixed amount for the memory and DASD allocations. As the physical CPU was shared between users so its utilization data was collected to charge the users. So that an accurate estimate of the price for CPU utilization can be made to satisfy the customer. The description of the type 1 record is as under [11]:

Columns	Contents
1 – 8	User ID
9 – 16	Account number
17 – 28	Date and Time of accounting
29 – 32	Number of seconds connected to CP
33 – 36	Milliseconds of processor time used, including time for supervisor functions
37 – 40	Milliseconds of virtual CPU time used
41 – 44	Number of page reads
45 – 48	Number of page writes
49 – 52	Number of requested virtual I/O starts for non-spooled I/O
53 – 56	Number of virtual punch cards sent to a virtual punch
57 – 60	Number of virtual print lines sent to a virtual printer
61 – 64	Number of virtual punch cards received from a virtual reader
65 – 72	reserved
73 – 76	Number of completed virtual I/O starts for non-spooled I/O
77 – 78	CPU address (for system VMDBK, this is real processor address)
79	Card generator field
80	Accounting record identification code

4.6.1 Data Selection

The data fields selected for charging the users in this project are:

- 1-8
- 17-28
- 29-32
- 33-36
- 37-40

The first field was used to identify the user precisely, the third field indicates the total time the user remains logged on. The field 37-40 indicates user CPU utilization that is actually charged to the user.

4.7 Monitoring Data

For monitoring aspect of the study the CP MONITOR facility was used that is also Z/VM built in facility. The CP monitor facility when enabled through CP MONITOR command, collects resource usage data that can be made available for further processing and analysis. CP Monitor facility provides the flexibility to control the amount and nature of data. Monitor collects performance data and saves in a saved segment called discontigous saved segment (DCSS). The Saved segment can be defined by using DEFSEG and SAVESEG CP commands. This Saved segment is writeable by CP and readable by an application MONWRITE. Monwrite module is shipped with Z/VM. MONWRITE uses IUCV to connect with the *MONITOR system service. MONWRITE writes monitor data from segment to the disk or tape. Virtual machines are defined with IUCV option in the directory entry for user that can access this Saved segment for data retrieval from the segment. The nature and amount of data depends upon the Sample or Event data that can be defined by using CP commands. Event data is collected when ever a specified event takes place in the system. This data provides the status of the system at that specific time. Sample data is collected and reported at the end of specified time interval. There are two kinds of sample data:

- Single Sample Data
  Single sample data is collected is collected once and some of this data provides a snapshot of the system at the time the data was collected. Other data consists of counters, or elapsed time values gathered at the end of each time interval.

- High Frequency Sample Data
  The data is collected more frequently then it is reported. The data is reported along with single sample data.

The diagram shows the performance monitoring overview.

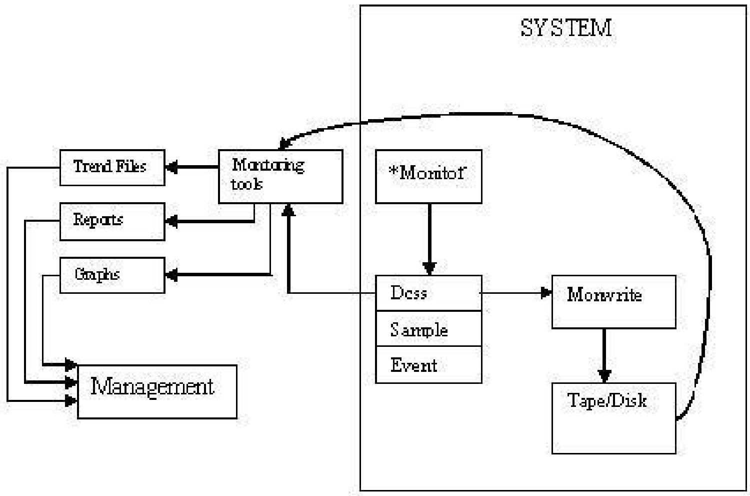

Figure 4.4: The basic overview of monitoring system

4.7.1 Data Organization

The performance data is stored into sets called Domains. Domains contain data about specific system operation areas. Following table describes the different domains and information stored in these domains [11].

Number	Name	Contents
0	system	Contains information about system-wide resource usage
1	Monitor	Contains information about installation configuration (processors, paging,storage, I/O, and so on) and on the type of monitoring enabled
2	Scheduler	Contains information about scheduler queues, the flow of work through the scheduler, and the resource allocation strategies used by the scheduler and the dispatcher
3	Storage	Contains information about use of real, virtual, expanded, and auxiliary storage
4	User	Contains information about virtual machines (such as scheduling status, virtual I/O and vector use, and logon and logoff events)
5	Processor	Contains data on work dispatched to a given processor as well as other data related to processor usage
6	I/O	Contains information about I/O requests, error recovery,interrupts, and other information for real devices
7	Seek	Contains information about seek operations for DASD devices
8	Application Data	Contains application data copied from a virtual machines storage when this storage has been declared to CP for collecting the data generated by the application program in that virtual machine

And with in these domains data is identified by the record type. Record type further divides the specific system operation areas (Domains) into specific part of that system operation areas . Each domain contains a lot of record types.

4.7.2 Data Selection

For this study domain 4, (user domain) and record type 3 was used to collect user related CPU utilization data and memory related data on z/vm. And for Linux side CPU usage, memory used and swapping activity was selected. Performance toolkit was used to retrieve selected data from MONDCSS. Performance toolkit has command line as well as web based interface. Six most busy Linux guests were selected for this part of the study. Resource utilization data of these guests was collected by running instances of Performance toolkit with a delay of 5 minutes. For Linux side resource utilization data RMFPMS was used. RMFPMS is installed on all the Linux guests and it runs continually on the Linux guests. The data is gathered by the processes running in the Linux virtual machine and fed to the Performance toolkit using the Distributed data server (DDS). DDS runs as a server process on the linux and feeds the utilization data to the performance toolkit over a TCP/IP connection. Linux guests must be registered before data can be retrieved from the linux guests.

The performance data is continuously collected on the z/VM side in a Saved segment (MONDCSS). All this collected data, Accounting as well as Performance data on Z/VM and Linux side was transferred to the reporting machine for further analysis. The data transfer from processing machine and from Linux servers was manual.

4.8 Tools Used

4.8.1 Performance Toolkit

Performance toolkit is a product from IBM for real time performance analysis as well as historical performance analysis. But for historical performance analysis the performance data should be collected first and then that historical data can be fed to the performance toolkit for analysis. Performance toolkit does not provide historical data itself. It connects with MONDCSS and displays the analysis reports on demand. Only report about the previous sample time is displayed. For example if sample delay is defined 1 minute. Then running Performance toolkit at 9:30 will display the data collected at 9:29. Following diagram explains the major functionality of the Performance toolkit.

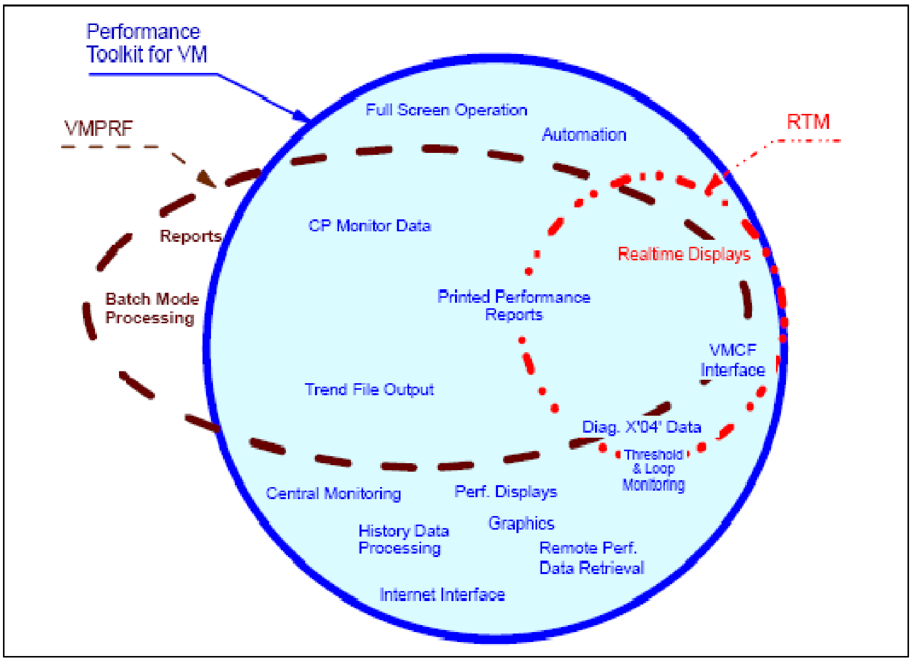

Figure 4.5: Functionality provided by the performance toolkit[12]

4.8.2 REXX

Rexx was used as a programming language to develop the scripts for data retrieval from DISKACNT and MONDCSS. Rexx stands for REstructured eXtended eXecutor and it is an interpreted programming language that was developed by IBM. REXX was used because it is easy to learn and easy to understand. The scripts written in Rexx were designed in a flexible way so that if a user wants to repeat the work with some extra parameters for charging and monitoring these parameters can be added without making any changes to the code and simply by adding the field selection criteria in the programs. The diagram illustrates the flexibility of the scripts and easiness of adding more parameters.

4.8.3 RMFPMS

"Remote measuring facility performance measurement" was used to collect linux guest data. This facility was already installed and configured on the linux servers running in the production environment. This is a modular data gatherer and the collected data can be analyzed by using RMF PM client application. The collected data can be accessed through XML over http[13]. RMFPMS can generate graphical trend reports, and data can be stored in spread sheet like formats that makes it easy to perform analytical calculations, can gather historical performance data. It also provides filtering to choose performance data for analysis.

4.9 Constraints And Limitations Of the Experimental Design

As this work was framed for a specified time frame so by keeping in mind this time constraint and the constraints imposed by the working environment like limited access privileges, impossibility to change the system configurations according to the experimental design, working domain was limited only to answer the basic questions described in the Research question section. The study does not propose any concrete and accurate charge back and monitoring system for the proposed environment. The main limitation of the experiment is the time interval used as a delay in the data collection in performance monitoring. The data was collected with a delay of 5 minutes between each collection. In production environment this time interval is considered more to find any performance problems. Because a problem can arise and go with in this time interval without being noticed. And also the sudden spikes in the cpu utilization can also happen in this time interval that can not be noticed.

Chapter 5

Results

The work was divided into two parts:

1. Charge-Back, Collecting the CPU utilization data for allocating costs to the customers, using the computing services of the system (Accounting Data).
2. Monitoring part, correlating the resource usage on the Z/VM side and resource usage on the Linux side.

Mainly the Cpu utilization by the user on the system side and user cpu utilization on the linux side was chosen for correlation. and similarly memory utilization on the Linux side was monitored along with the paging activity to see whether memory allocated to the Linux guest was sufficient for different applications running on the Linux server.

5.1 Charge Back

As the aim of this work is to provide a concept prototype showing what accounting data is available for charging customers and how this data can be metered in Z/VM environment. So focus was centered more on the way of collecting data and finding different parameters for metering resources. As described in the fig 4.1 the only resource that was metered for charging the users was CPU and all other resources were considered constant (charged on a fixed amount basis per month). The data collected for cpu utilization was in seconds, no. of seconds the physical cpu was used by the Linux guests running on Z/VM. The variable chosen contains three fields of information for resource utilization.

1. Time since Connected to cp
2. Total time for which cpu has been utilized (totcpu).
3. Time the user has used the cpu (Vcpu).

Here the first field of information provides the time value the Linux guest remains online. The second field of information gives the total time the cpu has been utilized by the user including the system overhead, (functions performed by the system on behalf of the user). And third field provides the actual time figures the user has used the cpu for its work.

5.2 limitations

The accounting data is dependent on the timings of the CP ACNT ALL command. When we issue the command the CP starts processing the accounting data recorded by the DISACNT virtual machine. The ideal time to run this command is at 00:00 so that data recorded the whole day can be processed. The data show the figures the cpu has been used from the time the first CP ACNT command was issued upto the next CP ACNT command issued. The table shows the look of the cpu utilization data retrieved from the Diskacnt virtual machine and processed by the processing machine running the script.

UserId	Time Conn. to CP(sec)	TotCpu(sec)	Vcpu(sec)
GJFTEAQ2	84363,33	250,499	243,589
GJFTWSQ4	84363,33	209,016	201,331
GJFTWSQ2	84363,33	2117,28	1903,277
GJFTLDQ2	84363,33	481,38	444,151
GJFTLXU2	84363,33	385.561	367.209
GJFTEAT2	84363,33	0.056	0.053
GJFTWST4	84363,33	87.978	83.771
GJFTWST2	84363,33	172.677	167.175
GJFTLDT2	84363,33	1.182	1.157

Table 5.1: Sample Accounting data retrieved and processed from Diskacnt virtual machine

5.3 Data Normalization

The accounting data was first normalized by applying some basic mathematical calculations for calculating system overhead and getting the values in percentage figures.

The following calculation was used to convert the cpu utilization into percentage values.

$$SystemCpubusy(percentage) = \frac{totalcputime}{Onlinetime} \times 100 \tag{5.1}$$

This calculation basically provides a way to see the total CPU utilization trend including the system overhead.

For user cpu usage percentage value the following formula was used:

$$UserCpubusy(percentage) = \frac{usercputime}{Onlinetime} \times 100 \tag{5.2}$$

This calculation shows the user daily cpu utilization trends. These trends can be used as information for future planning and decisions.

The following graph shows the virtual cpu utilization by the users in a period of 24 hours. From this graph one can estimate the nature of the work load of different users, Like the user GJFTSP4 is running more cpu intensive jobs as compared to the other users.

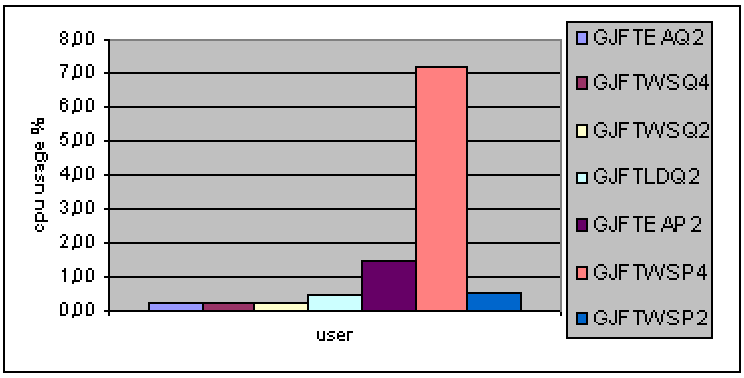

Figure 5.1: Cpu utilization by individual users in 24 hours

To have a clear picture of the user trends of cpu utilization the cpu utilization data for longer periods can provide a better estimate of the nature of the work load of users.

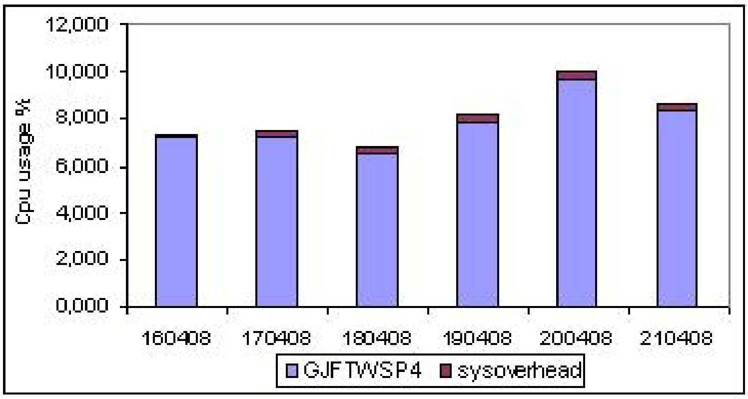

Figure 5.2: Cpu utilization by one user along with system overhead during six days period (z/vm view)

This graph shows the cpu utilization trend of the user during the period of six days and the cpu utilization shows a consistency on daily basis. These cpu

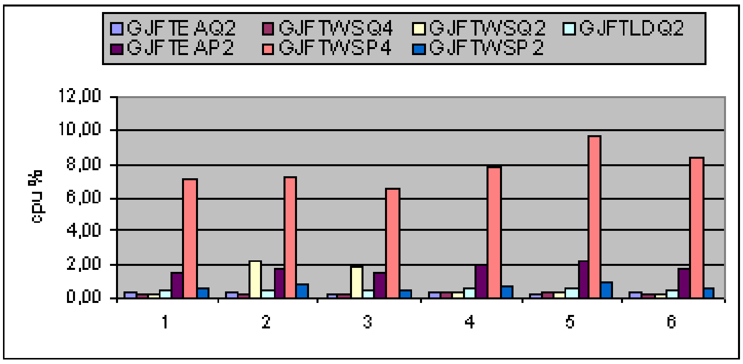

Figure 5.3: Cpu utilization by individual users during six days

usage trends can be used to know the overall system resource utilization and level of workload on the system. Like in the above graphs the the workload shown by the linux servers is not too much.

5.4 Performance Monitoring

The performance monitored data consists of CPU usage, memory usage and estimated working set size by the Z/VM for each guest. This working set size estimation done by the system is actually based on the memory usage by the user for that specified time interval.

5.4.1 CPU Usage Trend

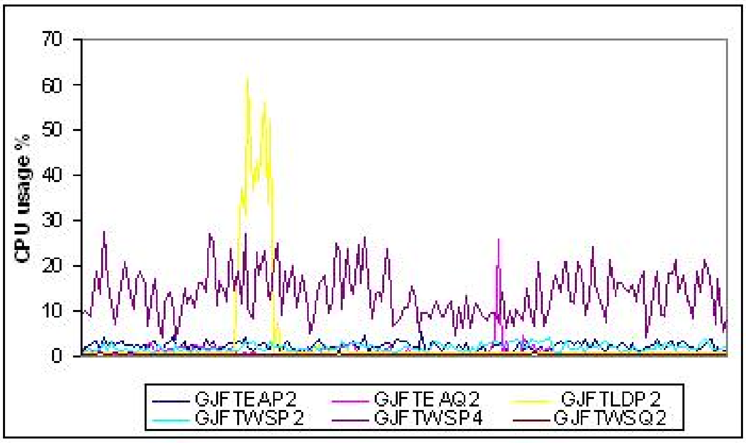

Figure 5.4: Cpu utilization by users during working hours in four days (z/vm view)

The graph shows Z/VM view of the cpu utilization during working hours for four days. The usage trend is almost consistent for all users throughout this period except one user who is showing an unusual behavior in the resource consumption. This graph is actually showing the overall picture of the system wide workload put by the selected linux guests. The following graphs drill down the usage trend on per user basis to isolate the behavior of the user.

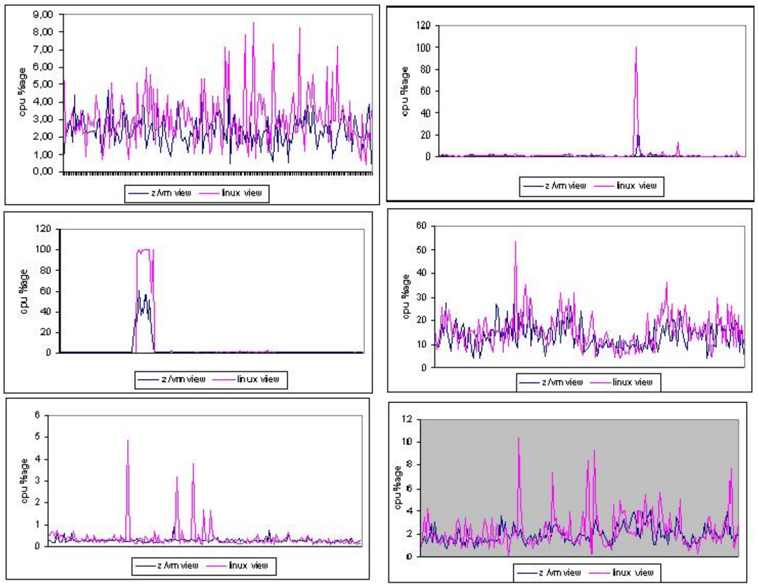

Figure 5.5: Cpu utilization per user during a period of 4 days (z/vm and linux view)

This graph shows the correlated values of cpu usage by guests on z/vm and linux side. Here the differences between the cpu usage values are visible that is a characteristic of shared resource environment. If z/vm side is showing high cpu value than Linux side it means z/vm is using some cpu cycles on behalf of that user. This system overhead can be in the form of page translations for that user or some scheduling activities. Some unusual patterns seen in the graph can be attributed to the delay used for data collection in this work. The delay interval between each collection was 5 minutes. The following graph shows the cpu usage trend for working hours. From daily trend reports the work load of the system can be examined on a daily basis. And having such daily reports helps to identify any increasing, decreasing or consistent usage trends by the users.

The graphs 5.7 and 5.8 shows the cpu utilization by one user both on Z/VM and on Linux side. The graphs shows some differences in the usage values on the z/vm and linux side. The difference is more clear in graph 5.8. This graph actually explains the fact that what we see on the linux side is not what is happening in the Z/VM side. The cpu utilization of 100% on linux side does not mean that the user is using the allocated cpu resource fully. The actual percentage of cpu usage by a linux guest is what we see on the Z/VM side. In the graph the user is showing 100% cpu utilization on the Linux side but at the same time interval it is just showing 55% cpu utilization. If we analyze the data carefully it becomes clear that the user is running data base application

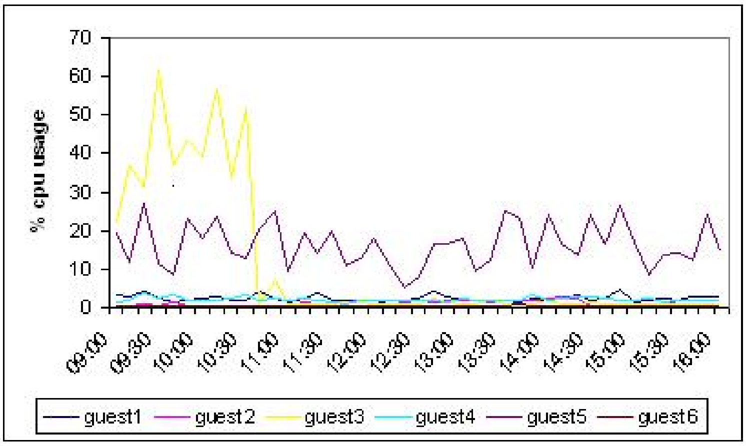

Figure 5.6: Cpu utilization by users during a period of 7 hours (z/vm view)

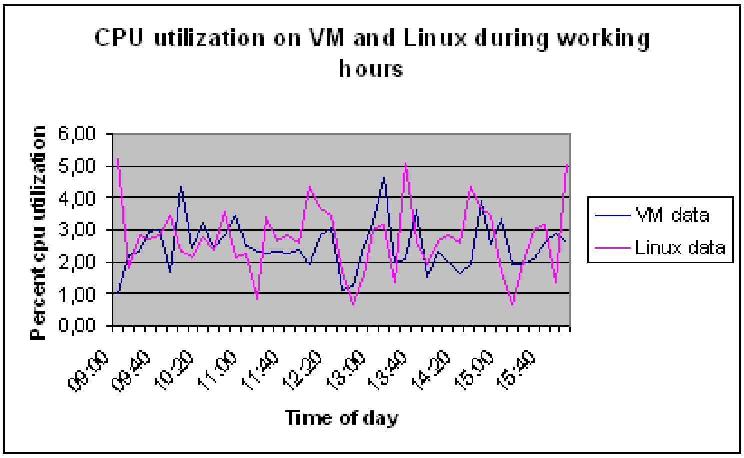

Figure 5.7: Comparison of cpu utilization on z/vm and linux side by one user during a period of 7 hours

and is busy in DASD I/O operations and the I/O wait value indicate that it is spending most of the time in wait state on z/vm side and z/vm puts the guest into dormant list until the I/O operation is complete. From dormant list the guest is put into eligible list ready to receive the resources and according to the class of the guest it is moved to the dispatch list and is dispatched according to the deadline priority calculated by the dispatcher. During this wait state the cpu usage on the z/vm side is zero but on linux side as linux fully occupies the virtual cpu so when it is doing I/O it shows high cpu usage value. Generally speaking in shared resources environment CPU is shared between users. And all the guests are just receiving only a share of the cpu cycles according to the weight values defined for those users. Similarly there is a point in the graph where Linux side data is showing 100% cpu utilization but on the Z/VM side

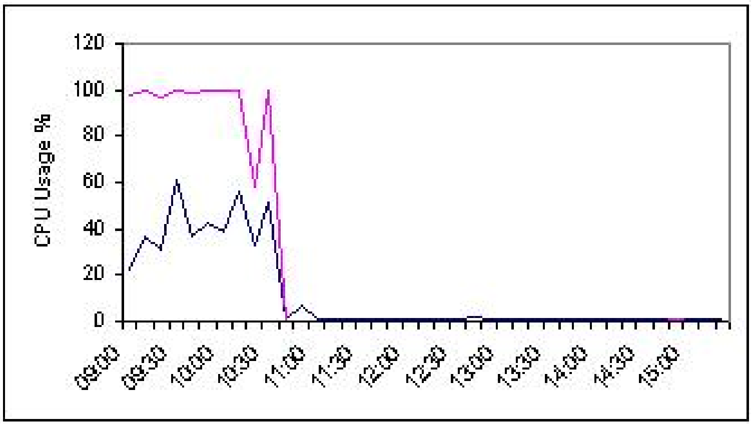

Figure 5.8: Comparison of cpu utilization on z/vm and linux side by one user during a period of 7 hours

the cpu usage has dropped to the level of 38.9%. The reason behind is same that guest is busy in I/O and I/O wait value is too high at that time. The following graphs shows the periodic cpu utilization on daily basis for four days. The time periods are divided into three groups for four days. The graphs shows that cpu utilization follows a similar pattern on daily basis except for one user who is showing high cpu usage for a short period of time. The second graph in the first figure shows a high standard deviation for the third day last period that is remarkable because the usage is almost same as for other periods.

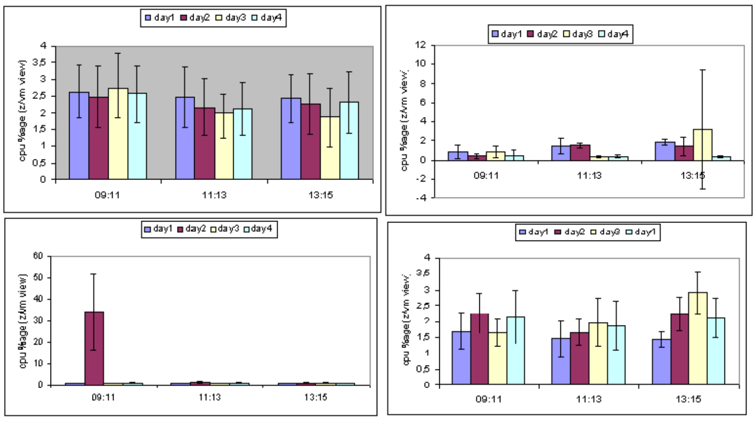

Figure 5.9: Periodic utilization trends by users

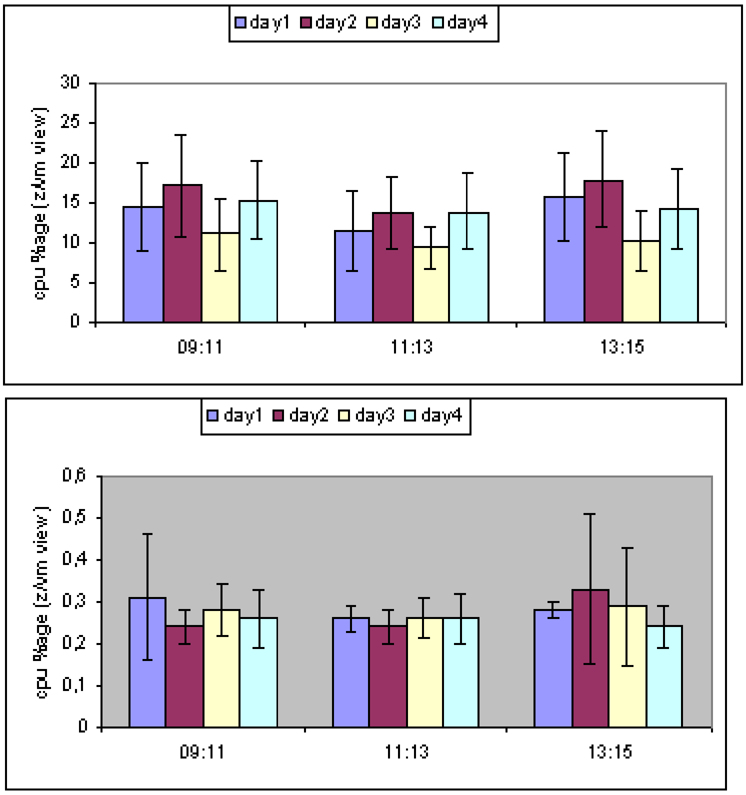

Figure 5.10: Periodic utilization trends by users

5.4.2 Relationship Between CPU utilization on Z/VM and on Linux Side

Pearson product-moment correlation was used to see any relationship between the cpu usage on both sides. The following table contains the calculated values for the selected users. The values of r and r^2 shows that the relationship

User Id	Mean (z/vm)	Mean (Linux)	σ (z/vm)	σ (Linux)	r	r^2
GJFTEAP2	2,33	3,08	0,854	1,488	0,06	0,004
GJFTEAQ2	1,08	2,12	2,066	9,223	0,18	0,03
GJFTLDP2	2,96	5,69	9,382	22,226	0,80	0,64
GJFTWSP2	1,96	2,34	0,723	1,534	0,25	0,061
GJFTWSP4	13,72	15,77	5,367	7,234	0,22	0,05
GJFTWSQ2	0,27	0,37	0,090	0,522	0,02	0,0002

Table 5.2: table showing Mean, standard deviation and correlation coefficient values of cpu data on z/vm and linux side

between cpu utilization on Z/VM and on linux side is very week. If we look at the scatter plots for all the users the fact is visible very clearly.

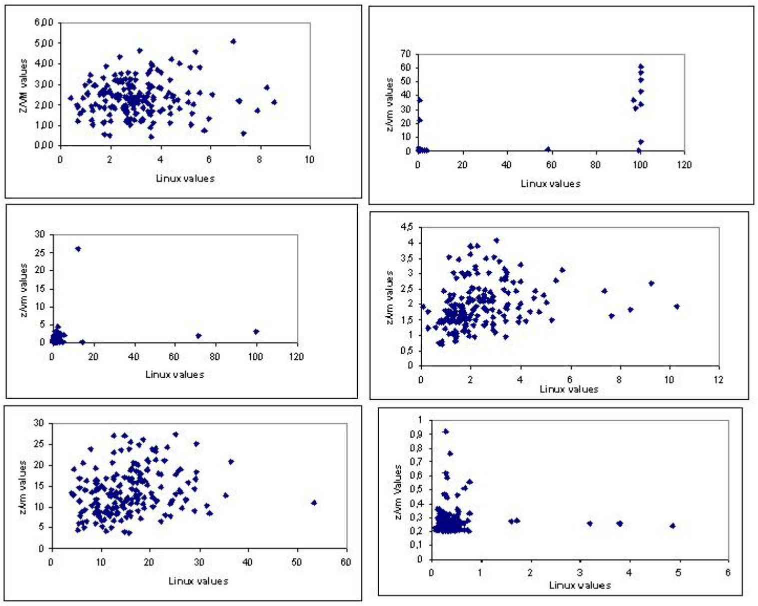

Figure 5.11: Scatter plots for all users

5.5 Memory usage By Linux

The memory usage data on Linux side provides a complete picture of how much memory it has been allocated, how much it is using and how much it has allocated to the buffers and caches. The following graph shows the memory allocated, buffer, cache and projected working set size for one particular Linux guest user. It is clear that although the Linux guest is using only a part of the memory allocated but it has allocated the rest of the memory for buffers and caches. The projected working set size is a bit higher value as compared to the memory used during that interval of time. Linux is basically designed in a way to use the main memory in a very efficient way. First it allocates memory to the kernel and applications and the unused memory is allocated to the buffers and caches. In the graphs the Linux has been assigned 1024 MB of memory except the last graph where the allocated memory is 3512 MB, and memory left after kernel and application allocation has been assigned to the caches and buffers.

The last graph is somewhat interesting because the memory used, allocated to buffers and cache is almost same through out that period but the Projected working set size estimated by the system is very low. The reason is unknown due to lack of application level data.

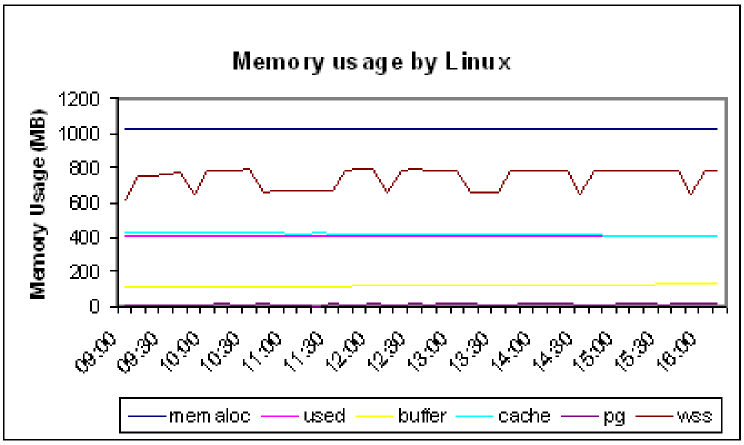

Figure 5.12: Memory Allocation and usage by Linux

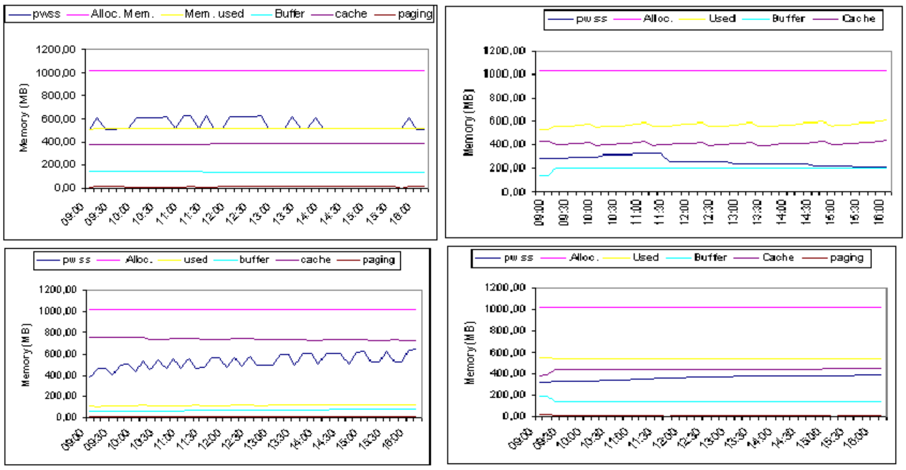

Figure 5.13: Memory Allocation and usage by Linux

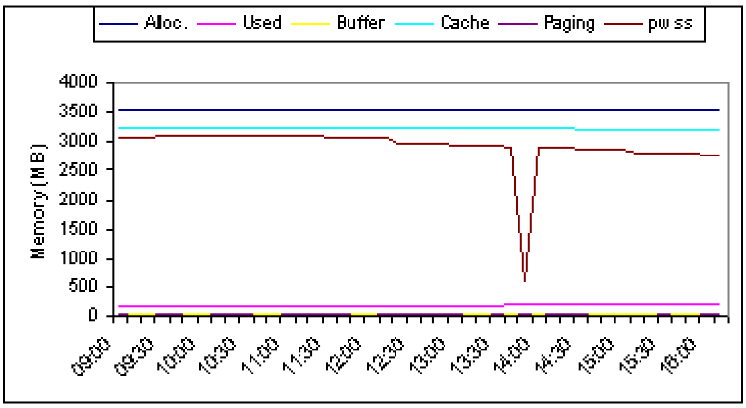

Figure 5.14: Memory Allocation and usage by Linux

Chapter 6

Discussion

6.1 Accounting Data Availability

The complete list of available accounting data has been discussed in the 3rd chapter. Although resource utilization data for different resources is available but the selection of data depends upon the policy of the data center providing the computing services to the customers. Whether the data center is providing certain resources on a fixed cost basis and some resources on the usage basis. So resources provided on usage basis are metered and resource utilization data of these resources is used for cost calculations. In this work only the CPU utilization data was selected for metering.

6.2 Value of Accounting data

In the first instance the resource utilization data seems to be important for just calculating the costs to interface with the users and to incur the setup, maintenance and operational costs. But this data contains a lot of information in it that can be used by the management to take necessary decisions for system wide capacity planning and to meet the SLA. With a slight modification to the processing virtual machine configuration it is possible to run the scripts for data collection automatically every day at specified timings. And in this way data for longer periods can be made available for making business wide decisions about hardware upgrading, outsourcing in the peak timings if resources consumption is showing very high usage values over a longer period of time. Some data centers do not use this accounting data for charging the users but they use this data for future decisions about hardware upgrades and new software licences. In this project the cpu utilization over a period of six days was collected and this data was used to estimate the cpu utilization trends of the users during this period. From these trends as shown in the graph 5.3 it is obvious that one user is running more cpu intensive work load as compared to the other users. This fact provides us an important information that by increasing the weight (share) of this customer running more cpu intensive work load, resource utilization can be improved. As the overall load run by these users is not too much so increase in the cpu share of that particular user will not effect the overall system performance. Although in shared resource environment when one user is idle then the one running more intensive cpu workload gets more cpu cycles. But if all the users are active at the same time then the user running cpu intensive workload has to wait for their turn to use the cpu cycles. If the weight of this user is increased then it will get more cpu cycles as compared to the others.

This accounting data can also be used to have a look on the overall system cpu load, and to know which user is a heavy user of the system resources (CPU) as compared to the others. This information can help in workload management and load balancing. If one LPAR is showing more users running cpu intensive workload than the other LPARs then some of the cpu intensive users can be migrated to other LPARs having low cpu intensive users, thus maximizing performance of the overall system. The data collected in this work shows actually no heavy cpu workload because the users are not running cpu intensive jobs. And the nature of their workload is almost consistent through out this period. An unusual resource consumption from the normal routine can be an indicator of something abnormal and can be drilled down to know whether this was a bug driven or business driven. A general rule is that it is not always easy to recognize a problem if we do not have a starting point for reference. So if we have this historical resource consumption data then this can serve as a reference data for diagnosing any problems in future. Any deviation from normal behavior can be diagnosed and can be fixed.

6.3 Performance Monitoring

6.3.1 CPU usage Monitoring

The performance data collected shows the CPU utilization trend of different users along with the information about which processes are using more cpu and memory. For performance monitoring we must keep in mind, from what aspect we are looking the monitored data. Whether we are looking for Reactive analysis or Predictive analysis. For reactive analysis we look only the monitored data of that user who has reported the performance problem. But in case of predictive analysis we analyze the whole monitored data for all users to look the overall system health and performance. One important thing while examining the performance monitored data is that we need some reference data that shows normal behavior. So that we can make any predictions for the system performance. If all the users are showing normal cpu usage and no alarming values are noticed in the usage data then process level data is not required but graph 5.6 in the results part shows an unusual resource usage by a user so we need to know what processes the user is running at that time to further know about whether the usage is bug driven or user is doing some cpu intensive work. The CPU utilization data on Linux side provides information about the workload nature of different users. Which process is using how much cpu amount and how much percent is spent in kernel mode and how much in the user mode means performing user work. As the linux guest is running DB at that time and doing heavy DASD I/O so processes running are data base processes and almost 23% of the total cpu usage on linux side is kernel driven means kernel is using high cpu at that time. It means the cpu usage is not bug driven but user driven. To have a further look on the user behavior of cpu usage on period basis we can use the graphs 5.9 and 5.10. In these graphs the user behavior is showing consistency with minor deviations. So such pattern recognition on period basis can help in identifying the users with specific usage pattern and this information can be used to group the users showing high cpu usage in one period with those who are using less cpu in that specified period. And such grouping or distribution of users in different LPARS can increase the overall performance of the system.

6.3.2 Projected Working Set Size

The graphs 5,12, 5,13 and 5,14 explains the fact that Linux has the tendency to use all of allocated memory by assigning it to buffers and caches. This policy works well for servers with dedicated memory because buffers and cache is preferred over unused memory. But this policy can have some consequences when Linux is running under Z/VM [14]. In Z/VM systems it is possible to allocate virtual memory to the guests that exceeds the actual physical memory of the system. This is possible in z/vm because z/vm allocates real memory pages to only active guest virtual memory pages. Idle memory pages are moved to the expanded storage or paging space. But in case of Linux guests most or all of the virtual memory pages are active because it allocates unused memory to buffers and caches. So when Linux guests are running under z/vm this over-commitment of memory can lead to a situation of degraded performance. Because z/vm will start stealing pages from active guests if physical memory is constrained. So in z/vm environment memory allocation should be done with extreme care and the size of the virtual memory allocated should be according to the kernel and application requirement running on the guest with a minimum size required by the buffer and cache. The interesting point in the memory figures is that all the guests are showing some Paging activity. But this is not a sign of caution as when an application first starts its executable image and data are paged in. And secondly page in is not problematic as compared to page out. Because linux page out memory when kernel finds that memory is very low it starts freeing up memory by paging out the pages of memory. This can happen time to time briefly. If in the collected data the value of page out is high constantly then it can cause a performance degradation because the kernel will spend more time in paging activity then to spend time for applications running at that time. The best estimate of memory for a linux guest running different applications (web as well as database) requires application level data. In the results part it has been shown in the graphs that projected working set size is a higher value as compared to the memory used by the Linux guest. The reason is that the Z/VM estimates the Projected working set size on the basis of the total memory allocated to the guest and considers active for the guest. So in the graph the guest has been assigned a memory of 1024 MB and it has used only 406 MB rest of the memory has been allocated to the cache and buffers. As shown in the collected data in Appendix A the memory allocated to buffers is 117 MB and to cache is 420 MB. And free memory is 6 MB. To calculate the projected working set size Z/VM actually considers the memory allocated to buffers and cache as being used by the Linux guest and therefore predicts a higher estimate of memory requirement for the guest as compared to the memory used by guest. Similarly in the figure 5,13 and graph 2 and 3 the used memory values are changing but there is no change in the buffer and cache memory values because the Linux guest is using free memory when memory use is increasing and in the same way releases memory to the free memory when memory usage has decreased. So it is clear that if Linux guest has some free memory it chooses to use it first. But again as we can see in the figure 5,14 most of the memory is allocated to buffer and used memory value is very low. The reason is that system has ample amount of memory and there is no resource shortage so almost all of the users have been allocated extra memory. But for future when more users will be added this requires some tuning and some memory can be reclaimed back from these guests. The best way to reduce the working set size is to reduce the buffer and cache memory and this can be done by allocating less virtual memory to the linux guests. Memory reports generated as above in the results parts for longer periods of time can help in determining how much memory of a particular guest can be reduced. From these results it is also visible that system is running Linux guests with over-committed memory but the system is running limited number of linux guests and number of total users is not too high therefore system has ample amount of free resources available. But in future if number of users increase then with the help of such memory usage reports as shown in results part this over-committed memory can be reclaimed back by the system to meet the new users requirements.

6.4 Critical Comparison with Vendor Supplied Tools

The charge back and monitoring tools discussed in the 2nd chapter are highly professional tools providing complete solution for charge back and monitoring in a Z/VM based virtual environment. These tools, Tivoli Usage and accounting manager and Unicenter:VM Account integrate within the system and provides intelligent business reporting facilities to interface with the customers. These tools use the system MONITOR facility to collect resource usage data and have built in analyzing modules that provides trend files and help in identifying any resource contention, if system wide load is heavy. The work presented in this report is although unable to provide any professional reporting but the required data can be collected and reports can be generated manually depending upon the need of the business, whether the management require daily usage trend reports or long term usage trend reports. Report generating front end can be designed for accounting data to produce usage reports automatically. But in spite of these missing parts the concept prototype is easy to design and requires no configuration changes if system is functioning already. Further no extra staff is required to run and report generation. But the above mentioned tools require user training and some configurations are required to make them running.

6.5 Validity Of Results

Performance monitoring is used to tune the system for better performance by finding any user problems who are using the computing services of the system. User problems can be in the form of increased response time for user commands, unable to login, system is not responding and many more. To diagnose such problems a short time interval is used like in seconds. The time interval used in the project does not support that aspect of the performance monitoring because 5 minutes are a large interval for this kind of monitoring but if system is performing well then this interval can be used to have a look on the overall system load for business wide decisions about capacity planning, outsourcing and any hardware upgrades if system is performing at its maximum load. And the second aspect that the author thinks important for this project is that data for longer periods is required to make any business wide decisions. But in this project data for only four days was collected due to time constraint. System optimization is not a single step work but it requires time and patience and sometimes optimizations require same step many times again and again to reach the goal of good performance. So according to author this work can serve as a road map with some minor modifications to reach the ultimate goal of developing a good charge back and monitoring system in future.

6.6 Repeatable Choices

According to author the approach was good but due to time shortage and some system constraints the author was unable to complete the work in the required and planned manner. In future the same setup and approach could be used for accounting data because it worked according to the requirement of the work. But in future instead of using performance toolkit the choice will be to use the rexx language for data retrieval script development on the monitoring side. Because the Performance toolkit just provides the timely snapshot of the system along with the work load nature of the users logged on to the system. It can be used to analyze the historical data if the data is saved already by some other means (using MONWRITE facility). The choice of performance toolkit in this project was based on due to system constraints. Similarly on the monitoring side same data along with some extra parameters can be a good choice for resource allocation estimation, like application level data.

Chapter 7

Conclusion

The aim of this project was to provide a concept study for development of an efficient charge back and monitoring system. It was tried in the project to answer the basic questions related to charge back and monitoring system so that a road map can be prepared for the development of an efficient charge back and monitoring system. Most of the project time was spent in learning Z/VM, what accounting and performance data is available on Z/VM and in finding different ways for collecting accounting and performance data. Basically the resources that are mostly considered important for any system include:

1. CPU
2. Memory
3. Direct Access Storage Device

More complex systems running many customer servers can include other resource as well like I/O channel path usage and paging activity. In this work only cpu usage data was used for calculating the costs to charge the users for computing services they are using because mostly the customers running linux servers on Z/VM and linux has the tendency to utilize all of the allocated memory so memory and DASD was considered constant means charged on a fixed amount basis depending upon the amount allocated to the customers. The accounting data collected not only serve for calculating costs but it can also serve for decision making. Daily, weekly and monthly resource usage reports can be generated from the collected data. Having historical resource usage data can also serve as a reference data to see any variations in the work load nature of different users. Similarly on the monitoring side performance data is monitored for fine tuning the system and to identify any abnormalities in the resources usage. The monitored data provide a picture of the system status at that particular time when it was collected. The data was collected with an interval of 5 minutes between each collection. But actually for performance monitoring this interval is too high because a performance problem can come and go silently without being noticed. Similarly the possibility that the reports generated by using this data contain some misleading information, is also considerable due to this high time interval used for data collection. The reason is that if the guest starts a cpu intensive process that consumes too much cpu on z/vm side say in 3rd minute of the 5 minutes interval and this process runs for 2 minutes and after 2 minutes it goes into I/O wait state then when we will start data collection after this interval the cpu utilization on linux side will be showing high cpu usage while on z/vm side it will be very low due to wait state. So due to this scenario there will be no match between the usage values on linux and z/vm side and will be considered as abnormal usage values leading to wrong inferences.

7.1 Future Work

This work puts only a slight light on the charge back and monitoring systems. There are some reasons that hindered in the proper design of the planned work. But future work can include application level data to know how much resources that application is using for estimating the resource allocation of similar applications in the future. Similarly in the performance context data about I/O activity along with CPU and memory utilization data can be included because I/O could also be a cause of degraded performance. And more robust ways of collecting data can be designed so that collected data can be stored more easily for future analysis. There exists many packages that collect and store data on linux side for detailed analysis like SYSSTAT and VMSTAT. Similarly instead of using Performance toolkit scripts can be designed that collects monitored data regularly. More over along with sample data event data can be included in the performance monitoring. And these events can be defined for which event data should be collected. The delay between the data collection requires to be small so future study can include a delay of 2 seconds which is considered default in z/vm. Future work can also include front end GUI and report generating module design so that a complete solution for charge back and monitoring system can be provided.

Bibliography

[1] Vizioncore. Attaining real cost tracking when going virtual. website. `http://www.vizioncore.com/WhitePapers/CostTracking+Chargeback_WP_SFS.pdf`.

[2] Thomas C. Ellett. Implementing charge back system. `http://www.charge-back.com/_ARTICLE/article1.htm`.

[3] Accenture. Creating and operating an effective and equitable shared services charge back framework. website. `http://www.accenture.com/NR/rdonlyres/012C02ED-E0D2-450D-BF71-51192B799024/0/fpm_sspov_creating.pdf`.

[4] ProfitPlus. Implementing charge-back systems. website. `http://www.profitplusabc.com/case_studies/studies/sharedservices.pdf`.

[5] B. Bitner. What is good monitor data. Technical report, IBM corp., 1992.

[6] L. Parziale, E. L. Alves, Eli M. Dow, K. Egeler, J. J. Herne, C. Jordan, E. P. Naveen, M. S. Pattabhiraman, and K. Smith. *Introduction to the New Mainframe: z/VM Basics*. IBM corp., NY, 2007.

[7] B. Darmawan, J. Siglen, L. Lundgren, and R. Catterall. *IBM Tivoli Usage and Accounting Manager V7.1 Handbook*. IBM corp., New York, 2007.

[8] Inc. Computer Associates International. *Unicenter VM:Account System Administrator Guide 1.6*, page 446. CA, 2002.

[9] V. Agarwal, N. Karnik, and A. Kumar. An information model for metering and accounting. Technical report, IBM India Research Lab, 2003.

[10] Stockinger H., Pagni M., Cerutti L., and Falquet L. Grid approach to embarrassingly parallel cpu-intensive bioinformatics problems. *Second IEEE International Conference on e-Science and Grid Computing*, page 58, 2006.

[11] IBM. *z/VM CP Planning and Administration, version 5 release 2*. IBM corp., 2005.

[12] G. Geiselhart, B. Dailey, D. DiTommaso, B. R. Degieux, K. Egeler, M. Held, L. Sousa, and R. Thibault. Linux on ibm eserver z series and s/390:performance toolkit for vm, 2004.

[13] O. Benke. Linux for zseries and s/390 performance measurement with rmf pm. Technical report, IBM corp., 2003.

[14] A. Erich, A. Ronald, and T. Arwed. Accounting and monitoring for z/vm linux guest machines. Technical report, IBM corp., 2004.

[15] S. Agarwala, R. Routray, and S. Uttamchandani. Chargeview: An integrated tool for implementing chargeback in it systems. *IEEE, 2008. proceedings*, 2008.

[16] G. Geiselhart, Y. Hara, B. Hinson, K. Smith, and C. Young. *IBM z/VM and Linux on IBM System z: Virtualization Cookbook for Red Hat Enterprise Linux 4*, chapter 12, Monitoring systems. IBM corp., NY, 2006.

[17] IBM corp. *Z/VM Getting Started with Linux on zSeries version 5 release 10*. IBM corp., NY, 2004.

[18] IBM corp. *Linux on system z: Device Drivers, Features, and Commands*. IBM corp., 2007.

[19] B. Gomolski and S. Mingay. Best practices for global chargeback. website. `http://www.businessengine.com/news/media/Gartner_Global_Chargeback_1106.pdf`.

[20] B. Abrahao and A. Zhang. Characterizing application workloads on cpu utilization for utility computing. 2004.

[21] S. O. Howard and J. O. Hallstrom. Specifying and enforcing resource utilization constraints. 2006.

[22] Mark Wisniewski and Ed Mekeel. Large memory performance studies. Technical report, IBM Systems and Technology Group, 2007.

[23] Gary King. Running ibm system z at high cpu utilization. Technical report, IBM, 2007.

Appendix A

Source Code

A.1 Accounting Script

```
/*************rexx**************/

/* Pipe command to display the account files and select the latest*/

'pipe cms listfile account * w ! sort 11.8! take first ! var record'

call lineout , record /*routine to write the data into variable*/

DISKACC = 'DATA 'substr(record,10,8)' A' /* defining the data file */

/* display different headings for data */

Line = 'userid''''Date' '' 'Connected' '' 'TotCpu''''VirtualCpu'

/* write the above line into the data file */

call lineout DISKACC , Line

/* main pipe part that reads the variable data and select the specified */

/* fields to put into array. */
```

```
'pipe ¡' record,
'! locate 80 /1/',
'! SPECS 1-8 1 17-22 10 29-32 C2X 20 33-36 C2X 30 37-40 C2X 40',
'! stem cpu.'
do i = 1 to cpu.0
conn = 0 /* Initializing the variable */
tcpu = 0
vcpu = 0;
Date = substr(cpu.i,10,6) /* reading date from the records */
/* If the some fields are empty then iterate */
if substr(cpu.i,20,8) = ' ' ! substr(cpu.i,40,8) = ' ' then iterate
else do
/* Converting hexadecimal data into decimal format */
conn = conn + X2D(substr(cpu.i,20,8))
/* converting cpu time into seconds */
```

```
tcpu = tcpu + X2D(substr(cpu.i,30,8)) / 1000

vcpu = vcpu + X2D(substr(cpu.i,40,8)) / 1000

/* write this data into a variable */

Line = substr(cpu.i,1,8)' ' Date ' ' conn' ' tcpu ' ' vcpu'

/* Write this variable into data file */

call lineout DISKACC , Line

end
end
call lineout DISKACC /* close data file
```

Appendix B

Performance toolkit Screens Used In This Work

B.1 Main Screen

This is the main screen that appears after login through web interface.

```
FCX124          Performance Screen Selection  (FL440 VM63447)     Perf. Monitor

General System Data        I/O Data                    History Data (by Time)
1. CPU load and trans.     11. Channel load            31. Graphics selection
2. Storage utilization     12. Control units           32. History data files*
3. Storage subpools        13. I/O device load*        33. Benchmark displays*
4. Priv. operations        14. CP owned disks*         34. Correlation coeff.
5. System counters         15. Cache extend. func.*    35. System summary*
6. CP IUCV services        16. DASD I/O assist         36. Auxiliary storage
7. SPOOL file display*     17. DASD seek distance*     37. CP communications*
8. LPAR data               18. I/O prior. queueing*    38. DASD load
9. Shared segments         19. I/O configuration       39. Minidisk cache*
A. Shared data spaces      1A. I/O config. changes     3A. Paging activity
B. Virt. disks in stor.                                3B. Proc. load & config*
C. Transact. statistics    User Data                   3C. Logical part. load
D. Monitor data            21. User resource usage*    3D. Response time (all)*
E. Monitor settings        22. User paging load*       3E. RSK data menu*
F. System settings         23. User wait states*       3F. Scheduler queues
G. System configuration    24. User response time*     3G. Scheduler data
H. VM Resource Manager     25. Resources/transact.*    3H. SFS/BFS logs menu*
                           26. User communication*     3I. System log
I. Exceptions              27. Multitasking users*     3K. TCP/IP data menu*
                           28. User configuration*     3L. User communication
K. User defined data*      29. Linux systems*          3M. User wait states

            Pointers to related or more detailed performance data
            can be found on displays marked with an asterisk (*).
Select performance screen with cursor and hit ENTER
Command ===>
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

Figure B.1: Performance screen selection menu

Option 21 and 29 was used in this work to retrieve the user resource usage data on Z/VM side and Linux resource usage on linux side. These screens are shown below.

B.2 Z/VM User Resource Usage Screen

The following screens shows the cpu utilization values on the Z/VM side by users running guest operating systems on the Z/VM.

```
FCX112      CPU 2084  SER 96A3A  Interval 14:41:47 - 14:42:07    Perf. Monitor
.         ____    .     .     .     .     .     .     .     .     .
          <----- CPU Load -----> <------ Virtual IO/s ------>
               <-Seconds->   T/V
Userid    %CPU  TCPU  VCPU Ratio Total DASD Avoid Diag98   UR Pg/s  User Status
>System<  3.34  .668  .508   1.3   102  101   141     .0   .0   .0  ---,---,---
LNXSU1    51.6 10.32 8.855   1.2  2267 2267  2267     .0   .0   .0  ESA,CL3,DIS
LNXRH3    24.0 4.796 4.066   1.2   4.7  4.7  1429     .0   .0   .0  ESA,CL3,DIS
LNXSU2    15.2 3.042 1.511   2.0    .0   .0    .0     .0   .0   .0  ESA,CLO,DIS
LNXRH1    14.1 2.825 2.333   1.2  1381 1376  1376     .0   .0   .0  XC, CLO,DIS
TCPIP     13.0 2.598 1.165   2.2    .0   .0    .0     .0   .0   .0  ESA,CLO,DIS
TCPMAINT   .27  .054  .050   1.1   1.2  1.2    .0     .0   .0   .0  ESA,CL3,DIS
DIRMAINT   .23  .046  .033   1.4    .0   .0    .0     .0   .0   .0  ESA,CL3,DIS
LNXSU6     .22  .044  .030   1.5    .0   .0    .0     .0   .0   .0  ESA,CL3,DIS
LNXRH2     .22  .044  .031   1.4    .0   .0    .0     .0   .0   .0  ESA,CL3,DIS
LNXRH4     .22  .044  .031   1.4    .0   .0    .0     .0   .0   .0  ESA,CL3,DIS
LNXSU3     .21  .041  .027   1.5    .1   .1    .0     .0   .0   .0  ESA,CL3,DIS
LNXSU4     .21  .041  .028   1.5    .0   .0    .0     .0   .0   .0  ESA,CL3,DIS
LNXSU5     .19  .037  .023   1.6    .0   .0    .0     .0   .0   .0  ESA,CL3,DIS
PERFKLA    .12  .024  .023   1.0    .4   .3    .2     .0   .0   .0  ESA,---,DOR
PERFLIV    .11  .022  .021   1.0    .4   .3    .1     .0   .0   .0  ESA,---,DOR
PERFROG    .11  .022  .020   1.1    .7   .6    .3     .0   .0   .0  ESA,CLO,DIS
PERFSVM    .10  .020  .019   1.1    .4   .3    .1     .0   .0   .0  ESA,---,DOR
PERFAPP    .08  .016  .015   1.1    .4   .4    .2     .0   .0   .0  ESA,---,DOR
Select a user for user details or IDLEUSER for a list of idle users
Command ===>
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F10=Left  F11=Right  F12=Return
```

Figure B.2: Resource usage screen on Z/VM side

By clicking on the particular user, detailed data about that user is displayed as shown in the next screen.

```
FCX115      CPU 2084   SER 96A3A   Interval 10:40:04 - 10:40:37    Perf. Monitor

Detailed data for user LNXSU1
Total CPU    :   114%     Storage def. :     128MB     Page fault rate:    .0/s
Superv. CPU :  34.4%     Resident <2GB:   32208       Page read rate :    .0/s
Emulat. CPU :  80.0%     Resident >2GB:       0       Page write rate:    .0/s
VF total     :  ....%     Proj. WSET   :   32198       Pgs moved >2GB>:    .0/s
VF overhead :  ....%     Reserved pgs :       0       Main > XSTORE  :    .0/s
VF emulation:  ....%     Locked pages :      10       XSTORE > main  :    .0/s
VF load rate:  ..../s    XSTORE dedic.:      0MB      XSTORE > DASD  :    .0/s
I/O rate     :  6343/s    XSTORE pages :       0       SPOOL pg reads :    .0/s
DASD IO rate:  6343/s    DASD slots   :       0       SPOOL pg writes:    .0/s
UR I/O rate :     .0/s    IUCV X-fer/s :      .0/s     MDC insert rate:    .0/s
Diag. X'98' :     .0/s    Share        :     100       MDC I/O avoided:   3.7/s
*BLOCKIO     :     .0/s    Max. share   :     ...

#I/O active :      0      Active     :100%     PSW wait :  3%     I/O act. : 20%
Stacked blk :     ..      Page wait  :  0%     CF wait  :  0%     Eligible :  0%
Stat.: ESA,P02,RNBL       I/O wait   :  0%     Sim. wait:  0%     Runnable : 98%

Proc.   %CPU  %CP  %EM   %VECT %VOHD %VEMU VLD/S   IO/S   Status
 01      55.9 18.3 37.6    ..,.  ..,.  ..,.  ..,.   3377   ESA,P02,PSWT
 02      58.5 16.1 42.4    ..,.  ..,.  ..,.  ..,.   2965   ESA,P02,RNBL

Data Space Name            Size Mode  PgRd/s PgWr/s XRd/s XWr/s Migr/s Steal/s
BASE                      128MB Priv      .0     .0    .0    .0     .0      .0

Device activity and status:
0009 3215   .0                         000C 254R       CL *, EOF    NOH NCNT
000D 254P      CL A, CO 01, NOH NCNT   000E 1403       CL A, CO 01, NOH NCNT
0190 3390   .0 1510,RR, 107Cyl,--->0   0191 3390   .0 15D2,WR,  20Cyl,--->0
019D 3390   .0 1510,RR, 146Cyl,--->0   019E 3390   .0 1510,RR, 250Cyl,--->0
0201 3390   .0 150D,WR, 200Cyl,--->0   0202 3390  3.7 150D,WR, 98% MDC eff.
0203 3390   .1 1552,WR,100% MDC eff.   0300 VDSK 6339 ----,WR,350000B,RS/RL
0592 3390   .0 1550,RR,  67Cyl,--->0   3000 OSA    .0 QDIO->SW1       INT.MISS
3001 OSA    .0 QDIO->SW1      SYSTEM   3002 OSA    .0 QDIO->SW1       INT.MISS
Enter 'STOrage Display' for storage details
Command ===>
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

Figure B.3: Detailed resource usage by selected user on Z/VM side

B.3 Linux Side Resource Usage Screen

The first screen is the main screen to select the linux guest for looking resource usage inside the linux.

```
FCX223      CPU 2084  SER 96A3A          Linux Systems         Perf. Monitor


     Selectable Linux Systems
     LNXSU1      LNXSU2      LNXSU3      LNXSU4      LNXSU5











Select a system for Linux details
Command ===>
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

Figure B.4: Linux guest selection screen

The following screen was used to choose cpu and linux memory usage screens.

```
FCX224      CPU 2084  SER 96A3A  Interval 18:49:00 - 18:50:00    Perf. Monitor

 Linux Performance Data Selection for System LNXSU1

   System Data
   Processes created per second       0.116
   Context switches per second        20.01
   Apache: Requests per second        0.014
           Bytes per request            478
           Busy threads                   1
           Idle threads                   2
           404 Errors per minute          3

 S Perform. Reports    Description
 _ LXCPU    LNXSU1     CPU utilization details

 _ LXMEM    LNXSU1     Memory utilization & activity details

 _ LXNETWRK LNXSU1     Network activity (overall & by device)

 _ LXFILSYS LNXSU1     File system size and utilization

 Command ===>
 F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

Figure B.5: Detailed data selection screen on Linux side

```
FCX230      CPU 2084  SER 96A3A  Interval 18:48:00 - 18:49:00     Perf. Monitor

 Linux CPU Utilization for System LNXSU1

                              <--- Percent CPU Utilization ---->  <-Accumulated (s)->
 Processor                    Total   User  Kernel   Nice   Idle  TotTm UserTm KernTm
 >>Mean>>                     50.51  50.23    0.27      0  49.48    ---    ---    ---
 cpu0                         82.78  82.51    0.26      0  17.21    ---    ---    ---
 cpu1                         18.26  17.98    0.28      0  81.73    ---    ---    ---

 Process Name
 mcf_base.s390_1.845          100.2  100.0    0.16      0    ---  76.61  76.51    0.1
 gpmddsrv.599                  0.85   0.65     0.2      0    ---   6.24   5.06   1.18
 procgat.589                   0.23      0    0.23      0    ---   0.14      0   0.14
 netgat.586                     0.1      0     0.1      0    ---   0.07      0   0.07
 bash.430                      0.03    ...    0.03    ...    ---   0.12    0.1   0.02
 sshd.429                      0.01    ...    0.01    ...    ---   0.88   0.65   0.23
 atd.388                          0      0       0      0    ---    ...      0      0
 cron.394                         0      0       0      0    ---    ...    ...      0
 keventd.5                        0      0       0      0    ---      0      0      0
 kjournald.24                     0      0       0      0    ---   0.02      0   0.02

 Command ===>
 F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

Figure B.6: Linux side Cpu usage screen

```
FCX229      CPU 2084  SER 96A3A  Interval 18:49:00 - 18:50:00     Perf. Monitor

Linux Memory Util. & Activity Details for System LNXSU1

Total memory size          1008MB     Swap space size           209MB
Total memory used           149MB     % Swap space used            0%
   Used for buffer            5MB     Swap-in rate               0/s
   Used for shared            0MB     Swap-out rate              0/s
   Used for cache            28MB     Page-in rate               0/s
Total free memory           859MB     Page-out rate            7.6/s


                         <----- Size -----> <------- Page Fault Rate/s -------->
                          (Bytes)      (kB)    Minor    Major <-Incl.Children->
Process Name             VirtSize  ResidSet MinPgFlt MajPgFlt MinPFltC MajPFltC
mcf_base.s390_1.845      101298k     79196      254        0        0        0
runspec.789              9981950      7700        0        0        0        0
sshd.429                 5165060      2404     ....     ....     ....     ....
sshd.760                 5074940      2392        0        0        0        0
gpmddsrv.594            37204000      2252        0        0        0        0
gpmddsrv.595            37204000      2252        0        0        0        0

Command ===>
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

Figure B.7: Linux side Memory Usage screen

Appendix C

Collected Data Sample

C.1 CPU Utilization data

GJFTEAP2(vm)	GJFTEAP2(linux)	GJFTLDP2(vm)	GJFTLDP2(linux)
1,04	5,2	0,7	0,2
2,20	1,79	0,87	0,05
2,94	2,72	0,72	0,15
2,98	2,83	0,77	0,06
1,70	3,47	0,77	0,36
4,37	2,34	0,75	0,23
2,42	2,15	1,02	0,13
3,26	2,76	0,81	0,23
2,44	2,37	0,75	0,21
2,79	3,56	0,78	0,32
3,47	2,15	0,8	0,16
2,50	2,24	0,84	0,11
2,33	0,88	0,74	0,23
2,22	3,36	0,75	0,33
2,34	2,65	0,72	0,2
2,26	2,86	0,75	0,13
2,38	2,61	0,75	0,1
1,90	4,36	0,68	0,3
2,88	3,66	0,71	0,18
3,03	3,5	0,72	0,13
1,14	1,7	0,78	0,15
1,22	0,68	0,81	0,21
2,50	1,56	0,68	0,32
3,24	2,98	0,73	0,31

C.2 Memory Utilization Data For User (GJFTEAP2)

Memory Allocated	Memory Used	Buffer	Cache	Working Set Size
1024	409	113	427	622,054
1024	408	113	424	761,156
1024	408	114	425	761,156
1024	408	114	425	766,570
1024	409	114	425	773,172
1024	407	115	423	649,582
1024	407	115	424	781,429
1024	406	116	422	785,004
1024	407	116	422	787,769
1024	407	117	422	792,711
1024	406	117	420	663,691
1024	408	118	421	668,762
1024	406	118	419	670,457
1024	407	118	419	668,125
1024	407	119	418	667,344
1024	406	119	418	789,535
1024	407	119	416	793,1875
1024	405	120	416	790,359
1024	406	120	417	660,125
1024	406	121	416	789,191
1024	405	121	416	790,344
1024	405	121	414	789,00
1024	404	122	414	785,992
1024	404	122	415	788,230