Linux on z/VM
Chargeback and Accounting

Barton Robinson
Barton@VelocitySoftware.com
Clifford Stoll’s “The Cookoo Egg”, Tracking a Spy Through the Maze of Computer Espionage

Why chargeback?
Accounting Issues
Accounting Models
Data Requirements
zVPS Accounting data

My first accounting program:
- written for IBM datacenter, 1976,
- Cobol, challenge: to correct CPU time for DOS/VS under vm/370
Clifford Stoll’s “The Cuckoo’s Egg”
1986, computer time at LBL cost $300/hour
Accounting had 75 cent discrepancy
The Process took almost a year:
- Set up logging
- Find the hacker’s entry point (LBL, 1200 baud dial up modem)
- Access to a dozen networks, 1000’s of computers
- Show the “defense contractors” that had been compromised

Result: Hackers (students in Germany), stealing computer time, sold information to KGB to support drugs, phone bills

When doing accounting, the numbers must add up!
IT as a business - Profit center vs Cost center

Cost center
- Corporate line item
- One price, percent of gross
- Easily outsourced to more efficient “profit organizations”
- Often inherent waste, decisions made based on religion
- Less regard to new initiatives (big data?)

Profit center
- “free” to corporation
- Funded by the users, they have choices
- More efficient because it has to be
- Platform decision results measurable by the users
Why chargeback?

Capital Expenditures “Large” vs “small”
- Capital Expenditures seen by executive committee
- Justification of $100K IFLs different than servers
- Distributed / Rack servers often under the “sign-off” limit

Justification for hardware
- Justify IFLs – knowing who is using them
- Knowing what application is paying for the new ones

In profit center model, IFLs are justified by user requirements and funding
Supporting Corporate Interests and service levels
- Understanding costs promotes competitiveness
- Understand service and service level requirements
- Encourages higher service levels at lower cost

Chargeback Encourages::
- Lower resource requirements
- Tuning applications
- Workload scheduling
- Better service levels if heavy CPU spikes moved off shift

Accurate accounting models takes religion out of platform decisions
- Bad decisions are not buried in a corporate line item
Charge structure resolves issues

A favorite “cron” job has a measurable cost
- Many sysprogs have their personally developed “cron job”
- It runs across many servers at “low cpu cost each”
- It may cost 3 IFLs in aggregate or:
- It may cost $300,000 per year or more to operate – is it worth it?

Putting a structured cost to operations changes perspectives

How much of the system resources are wasted because costs are unknown or not understood?
Cloud Challenges

Cloud:
- Cloud: Old concept, new name, same requirements
- Time Sharing existed in the 70’s, same requirements
- Dynamic access to resource and service
- Often 100’s or 1000’s of servers (z)

Cloud as a business
- What service is being provided
- What to charge for a service?

Cloud REQUIRES accounting
Amazon charges

Amazon

- Charge by IP address
- Charge by bandwidth
- Storage (100tb -- $8400/month)
- DB: 1st 25 cpu hours per month free
- DB: row selects, adds, ….
- Email requests
- Data base loads
- 1TB of HPC: $28,000 / year

Everything is measured
Google

- **Google Cloud Storage pricing** is based on usage. Project storage usage and bandwidth usage are calculated in gigabytes (GB), Storage (100tb -- $8400/month)

- All machine types are charged a minimum of 10 minutes. For example, if you run your instance for 2 minutes, you will be billed for 10 minutes of usage.

- After 10 minutes, instances are charged in 1 minute increments, rounded up to the nearest minute.

- A 10TB persistent disk $400/month
Pricing model

- 10x10 cage
- 30 Amp redundant circuit (15 amps used)
- A/C included in price of circuit
- 20 Amp circuit
- 16 IP addresses
- High bandwidth connection
- Max 1 gigabit / month, incremental charges after that

Services include

- UPS, Diesel generators (2 weeks of diesel on site)
- Security – badge in/badge out
- Internet
Performance Management Requirements

Chargeback implies service levels as well:

Capacity Planning
- Methodology needed to project future requirements

Performance Analysis
- Tools/education needed to resolve current service issues

Operations
- Alerts for service issues
- If a process loops, who is responsible for detection and correction?
Define amount of money to be recovered
Charges for “z” must be “reasonable” as compared to “x”

Cost allocations must be correct
  - Allocating data center costs these days to “Z” does not make sense
  - Run as a business, departments don’t pay to support other departments

Honest allocations
  - Electricity, A/C
  - Floor space

System programmer who says: “I will quit if I have to support chargeback”
Every data center has mixed technologies, other virtualization shared resource should be considered.

If adding one more distributed server requires a new A/C Chiller, and one more chiller on the roof requires a new building, who pays the true costs? (True example, converted to “z” and saved a building)

If new data center is required to support more racks (or networking, or…) who pays?
- Incremental costs of the racks can be buried
Defining Costs

Hardware costs
- Processor, storage, network

Software “shared” licenses, by platform
- z/VM and Linux system software (and support)
- Linux application software
- Infrastructure (management, backup)
- Don’t charge HP Openview to the mainframe

Personnel Costs, by platform:
- Systems support (including office expenses)
- Operations

Network – based on bandwidth, user requirements
- Connection costs
- Personnel
Defining Costs

Environmentals, (BY PLATFORM)
- Floor space
- Power/Air Conditioning
- Power requirements should be documented for ALL hardware

Disaster Recovery (or lack of)
- Don’t pay for it if it won’t work
- Major strength of “z”, but has associated costs

Data backup
Accounting Model Considerations

Service level definitions

Design to meet Corporate Interests
- Move workloads to low resource periods when possible
- Minimize resource spikes during prime shift
- Convince users to tune their applications

Accounting model considerations
- Different support models (critical, production, development)
- Don’t charge for off shift resource consumption
- Charge for resources relevant to dollar purchases

Off shift work – when?
- Backups
- Database loads
- Server creates
- Batch updates
Influencing the Users

Resources that are the most expensive and most limiting:

- Processor / IFLs
- Real Storage

Chargeback should convince users to

- Move work off shift
- Tune applications
- Use real storage realistically (is 20 GB SGA really needed?)
- Utilize the I/O subsystem instead of caching the world
Planning for Chargeback

Accounting Methodology:
- Charge for resource consumption
- Flat rate
- Combination

Acknowledge resources have both fixed and variable costs
- Disk space is fixed or variable?
- CPU time is variable
- Network connections are fixed
- Network use is variable
Hello! There are going to be a lot of ideas on how to do charging all the way from "we don't charge back" to "we charge by the micro-process".

Fixed-price charging is good for very well contained zLinux servers. Using SHARE ABSOLUTE and even CPU POOLING can help with this. CPU POOLING is really good if you are thinking of mixing those fixed-resource servers with high-performance or "premium services" servers.
There are pros and cons to fixed price charging. A pro is it is MUCH easier than any other kind of charging (other than no-cost!).

A con is that some folks may think they are being charged too much if they really don't use it that much or that heavy, and some may not be thrilled being capped when they want to run something heavy for a short while.
We use a variable rate based on consumption for most of our servers. We also have "lab" servers that are contained within SHARE ABS. We have processes to "cap" non-premium production servers when they get out of hand and the entire processor has been running in the high 90's for CPU.

We use zVPS to collect all the data for the processor/LPARs and the zLinux servers to evaluate what they are up to every minute. Using that data, we are able to charge-back to the business units based on what they consume by process/application.
There is a small base charge for just having a server, then the use of resources adds to it. This was all built over the last 10 years and we are still tuning it! Figuring out what your business will accept for charging is the hard part. Between the knobs in z/VM and performance monitoring/collection tools available, you can make it work the way you need it. -- *James Vincent* -- President, SHARE Inc
Planning for Chargeback

Computing Resources – traditional model
- by CPU second
- by resident storage (storage used)
- DASD I/O, disk space

Current model
- By CPU second
- By virtual machine size
- Disk space
- Network bandwidth required
- Network bandwidth utilized
Accounting Example

- Support types:
  - Premium: 7x24 technical support
  - Prime: Prime shift technical support
  - General: Technical support as available

- Charge for CPU consumption 7AM to 5PM
  - (No charge off shift)

- Charge for Average Storage residency
  - (Encourages smaller virtual machines)

- Charge for DISK space allocated

Model proven to minimize IFL and storage requirements
Accounting Model Failures

Accounting Model tried (large manufacturing company)

- Monthly flat fee per server

Flat fee encourages:

- Larger servers doing more work
- Wasting of resources

The need for additional IFLs with no additional revenue broke the model

- Fee charged needs to be the maximum to survive
Model based on Linux CPU data
- Prior to sles10, CPU data “VERY” bogus
- One test case showed Linux CPU data wrong by 100

- z/VM data is accurate to the microsecond -

Solution - prorate
Data Challenges – data objectives

- Data collection must not be prohibitive
- Spikes must be validated with historical data
- Capture Ratios (data model) must be validated
- Accurate accounting data source
Resource utilization data requirement
- CPU by z/VM Virtual Machine (Linux server)
- CPU by Linux process
- CPU by application
- Real Storage residency

Historical data required
- Customer asks “why did I spike on October 20”? 
- What processes, application was running?

Data Accuracy
- Data is accurate (Linux CPU measurements are not)
- Data must be validated
- Data collection must not be prohibitive

Capture Ratio (100%????)
- How much of resource consumed is measured?
- Validates the model
CPU Accounting

CPU consumption is key charge item
- 10-20GB ram required per IFL
- Software licenses
- IFL initial fee and maintenance

What CPU is charged?
- Infrastructure / support? (Overhead)
- System Overhead (by virtual machine)
- LPAR / CP Overhead?

Prorate function still needed?
- z/VM data very accurate
- Pre SLES10, Linux data under “VM” was bogus
- Linux data mostly valid (post sles10)
- Linux data does not include overhead
- Prorate used to include virtual machine system overhead
Traditional data model (1989)

**ZMON: Real time analysis**
- Uses Standard CP Monitor
  Real Time Analysis

**ZMAP: Performance Reporting**
- Post Processing
  Creates Long Term PDB
- PDB or monwrite data input

**PDB (Performance DataBase)**
- Complete data
- By Minute, hour, day
- Monthly/Yearly Archive

---

**Diagram:**
- **VM CP Monitor**
  - **ZMON**
    - **PDB**
    - **ZMAP**
    - **PDB**
  - **Reports**
    - **Real-Time Displays**
Linux and Network Data Acquisition

ZTCP: Network Monitor

- SNMP Data collection
- Data added to PDB
- Availability Checking

ZTCP collects data from:

- LINUX (netsnmp)
- NT/SUN/HP (native snmp)
- Printers/Routers….
Accounting using zVPS:
- CPU data from z/VM very accurate
- Linux process data accurate
- Linux application data accurate
- Historical data maintained
- Capture ratios 100%

Traditional z/VM accounting - ESAACCT
- by CPU second,
- by resident storage
- DASD I/O
- Reported on ESAACCT (accounting report)
- Extracted from performance database
- Users can see data real time
Traditional z/VM Accounting

**Accounting Definitions:**
- Service unit: Chargeable unit (compare z9 to z10)
- CPU Service Units: Service Units per CPU Second
- Storage Service Units: Service Units per page resident
- DASD I/O: Service Units per Virtual I/O

**Real Storage Accounting (example)**
- Storage costs $500 / GB per month
- Prime shift (20 days per month), storage costs $25 / GB
- Server averages 2GB resident per day, should be charged $50 day
- Choose STORAGE FACTOR

**DASD I/O – More arbitrary**
- Channels, Storage processors may be factors
Traditional z/VM Accounting

Report: ESAACCT      User Accounting Report                        Veloc
Monitor initialized: 10/19/08 at 14:00:00 on 2097 serial 274E2 First

Service Unit Virtual CPU Factor:    32989
Service Unit Overhead CPU Factor:   32989
Service Unit I/O Factor:            0.800
Service Unit Storage Factor:        0.500
Charge per Service Units:           0.00001

UserID  <---Users-->  <Service Units>  Charges  <---Resources Used--->
/Class   Logged Actv  /Sec  Total  Total        CPU  DASD I/O  Pages
--------  --------  ------  ------  -------    ------  -------  ------
14:15:00  61  40.4  67058  60353K  603.52  1551.1  337088  18M
SYSTEM    240  715.67  644103  6.44  19.52

***Key User Analysis***
RSCS      1  1.0  0.67  601.32  0.06  0.02  0  24
TCP/IP    1  1.0  21.80  19619  1.96  0.58  0  810
VMSECURE  1  1.0  1.71  1535.0  0.15  0.04  26  65

***User Class Analysis***
*Servers  29  12.4  156.69  141023  14.10  3.91  12819  3719
*Linux    23  23.0  66161  59545K  595.45  1527.0  324243  18M
*Misc     4  2.0  0.50  448.13  0.04  0.00  0  667

***Top User Analysis***
LXSG0012  1  1.0  33662  30295K  302.94  903.65  228801  604K
LXSG0020  1  1.0  6365.4  5729K   57.28  146.40  4354  1792K
LXSG0011  1  1.0  5661.6  5095K  50.95  119.10  4641  2325K
LXSG0002  1  1.0  3778.2  3400K  34.00  72.64  30204  1960K
LXSG0003  1  1.0  3111.4  2800K  28.00  72.40  3625  818K
LXSG0017  1  1.0  1325.5  1193K  11.92  29.79  2039  417K
LXSG0016  1  1.0  1319.4  1187K  11.87  24.08  3079  781K
LXSG0008  1  0.9  1104.9  994451  9.94  15.27  1790  978K
LXSG0023  1  0.5  569.82  512841  5.12  9.26  1160  413K
LXSG0015  0  0.5  522.87  470584  4.76  8.75  1347  362K
Options:
- Charge per server
- Charge by application

Must have ability to show cost of “cron”

Requirements for application, infrastructure chargeback
- Must have full process table information every interval
- Must have process parent / child relationships
- Must have capture ratio exposed

zVPS Collects full process table every 60 seconds
- Data includes parent child relationship
- When process terminates, CPU is moved to parent “children buckets”
- An application is defined as either by process name (Oracle), or by parent
- Websphere identifies the “head of application”, that owns the worker processes
### High Linux CPU capture ratio

**Report: ESALNXV**  
**LINUX Virtual Processor Analysis Report**

<table>
<thead>
<tr>
<th>Node/ Name</th>
<th>VM</th>
<th>&lt;Linux Pct CPU&gt;</th>
<th>&lt;Process Data&gt;</th>
<th>Capture Prorate</th>
<th>Ratio Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:03:00</td>
<td>NEALE1</td>
<td>LNEALE1</td>
<td>100.0 11.4</td>
<td>88.6 100.2 11.5</td>
<td>88.7 1.002</td>
</tr>
</tbody>
</table>

---

**Report: ESALNXP**  
**LINUX HOST Process Statistics Report**

<table>
<thead>
<tr>
<th>Node/ Name</th>
<th>&lt;-Process Ident-&gt;</th>
<th>Nice</th>
<th>&lt;------CPU Percents----&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:03:00</td>
<td>NEALE1</td>
<td>0</td>
<td>0 0 100 0.43 3.35 11.0 85.4</td>
</tr>
<tr>
<td></td>
<td>kswapd0</td>
<td>100</td>
<td>1 1 0.12 0.12 0 0 0</td>
</tr>
<tr>
<td></td>
<td>snmpd</td>
<td>1013</td>
<td>1 1012 -10 0.13 0.03 0.10 0 0</td>
</tr>
<tr>
<td></td>
<td>sh</td>
<td>3653</td>
<td>3652 30124 0 52.7 0 0 9.37 43.3</td>
</tr>
<tr>
<td></td>
<td>gmake</td>
<td>9751</td>
<td>9750 30124 0 43.4 0.02 0.02 1.37 42.0</td>
</tr>
<tr>
<td></td>
<td>sh</td>
<td>10129</td>
<td>9751 30124 0 0.02 0.02 0 0 0</td>
</tr>
<tr>
<td></td>
<td>sh</td>
<td>10130</td>
<td>10129 30124 0 0.63 0.03 0.23 0.28 0.08</td>
</tr>
<tr>
<td></td>
<td>ccl</td>
<td>10307</td>
<td>10306 30124 0 3.12 0.18 2.93 0 0</td>
</tr>
<tr>
<td></td>
<td>rpmbuild</td>
<td>30124</td>
<td>16382 30124 0 0.07 0.03 0.03 0 0</td>
</tr>
<tr>
<td></td>
<td>sh</td>
<td>30125</td>
<td>30124 30124 0 0.02 0 0.02 0 0</td>
</tr>
<tr>
<td></td>
<td>gmake</td>
<td>30126</td>
<td>30125 30124 0 0.02 0 0.02 0 0</td>
</tr>
</tbody>
</table>

---

**Report: ESALNXC**  
**LINUX Process Conf**

<table>
<thead>
<tr>
<th>Node/ Name</th>
<th>&lt;-Process Ident-&gt;</th>
<th>ID</th>
<th>PPID</th>
<th>GRP</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEALE1</td>
<td>init</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>init [3]</td>
</tr>
<tr>
<td></td>
<td>migratio</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>migratio</td>
</tr>
<tr>
<td></td>
<td>ksoftirq</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>ksoftirq</td>
</tr>
<tr>
<td></td>
<td>events/0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>events/0</td>
</tr>
<tr>
<td></td>
<td>khelper</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>khelper</td>
</tr>
<tr>
<td></td>
<td>kblockd/</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>kblockd/</td>
</tr>
<tr>
<td></td>
<td>cio</td>
<td>41</td>
<td>4</td>
<td>0</td>
<td>cio</td>
</tr>
<tr>
<td></td>
<td>cio_noti</td>
<td>42</td>
<td>4</td>
<td>0</td>
<td>cio_noti</td>
</tr>
<tr>
<td></td>
<td>kslowcrw</td>
<td>43</td>
<td>4</td>
<td>0</td>
<td>kslowcrw</td>
</tr>
<tr>
<td></td>
<td>appldata</td>
<td>96</td>
<td>4</td>
<td>0</td>
<td>appldata</td>
</tr>
<tr>
<td></td>
<td>aio/0</td>
<td>101</td>
<td>4</td>
<td>0</td>
<td>aio/0</td>
</tr>
<tr>
<td></td>
<td>pdflush</td>
<td>5266</td>
<td>4</td>
<td>0</td>
<td>pdflush</td>
</tr>
<tr>
<td></td>
<td>pdflush</td>
<td>26647</td>
<td>4</td>
<td>0</td>
<td>pdflush</td>
</tr>
<tr>
<td></td>
<td>kswapd0</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>kswapd0</td>
</tr>
<tr>
<td></td>
<td>kmcheck</td>
<td>158</td>
<td>1</td>
<td>1</td>
<td>kmcheck</td>
</tr>
<tr>
<td></td>
<td>syslogd</td>
<td>976</td>
<td>1</td>
<td>976</td>
<td>/sbin/syslogd</td>
</tr>
<tr>
<td></td>
<td>klogd</td>
<td>979</td>
<td>1</td>
<td>979</td>
<td>/sbin/klogd</td>
</tr>
<tr>
<td></td>
<td>snmpd</td>
<td>1013</td>
<td>1</td>
<td>1012</td>
<td>snmpd</td>
</tr>
<tr>
<td></td>
<td>portmap</td>
<td>1030</td>
<td>1</td>
<td>1030</td>
<td>/sbin/portmap</td>
</tr>
<tr>
<td></td>
<td>rpciod</td>
<td>1034</td>
<td>1</td>
<td>1</td>
<td>rpciod</td>
</tr>
<tr>
<td></td>
<td>lockd</td>
<td>1035</td>
<td>1</td>
<td>1</td>
<td>lockd</td>
</tr>
<tr>
<td></td>
<td>sshd</td>
<td>1072</td>
<td>1</td>
<td>1072</td>
<td>/usr/sbin/sshd</td>
</tr>
<tr>
<td></td>
<td>sshd</td>
<td>16272</td>
<td>1072</td>
<td>16272</td>
<td>sshd: bu</td>
</tr>
<tr>
<td></td>
<td>sshd</td>
<td>16288</td>
<td>1072</td>
<td>16288</td>
<td>sshd: bu</td>
</tr>
<tr>
<td></td>
<td>sshd</td>
<td>16290</td>
<td>16288</td>
<td>16288</td>
<td>sshd: bu</td>
</tr>
<tr>
<td></td>
<td>bash</td>
<td>16291</td>
<td>16290</td>
<td>16291</td>
<td>bash</td>
</tr>
<tr>
<td></td>
<td>python</td>
<td>16312</td>
<td>16291</td>
<td>16291</td>
<td>python</td>
</tr>
<tr>
<td></td>
<td>do-bui</td>
<td>16313</td>
<td>16312</td>
<td>16291</td>
<td>/bin/sh</td>
</tr>
<tr>
<td></td>
<td>bb_do</td>
<td>16382</td>
<td>16313</td>
<td>16291</td>
<td>/usr/bin</td>
</tr>
<tr>
<td></td>
<td>rpm</td>
<td>16415</td>
<td>16382</td>
<td>16415</td>
<td>rpmbuild</td>
</tr>
<tr>
<td></td>
<td>rpm</td>
<td>30124</td>
<td>16382</td>
<td>30124</td>
<td>rpmbuild</td>
</tr>
</tbody>
</table>
Storage map - **CAPTURE RATIOS always critical for any instrumentation:**

- CP Fixed Storage
- CP Non Pageable
  - Free storage (only VMDBLKs)
  - Frame tables
- Dynamic Paging Area (DPA)
  - System Execution Space
  - User storage, MDC, Address Space, Vdisk – **This validates user resident storage**
  - Available List (greater/less than 2gb)
Database (Performance DataBase/PDB) Extraction

- Define headings as you like
- Define variables, summary data only

EXTRACT: ; First extract system data

columns = '10 10 7 8 8 8'
TITLE = ' USERID    ACCTNBR      CPU   Pages    VIO'
TITLE = ' --------- -------- -------  ------  ------'
X = 'USERID'
Y = 'USRCON.VMDACTNO'
Y = 'USEACT.VMDTTTIME' ; cpu utilization
Y = 'USEACT.VMDCTPVR / 256' ; MB Resident
Y = 'USEACT.VMDVDSCT' ; VIO

INTERVAL = 'SU'
CRITERIA = 'USRTYPE = USER'
CRITERIA = 'USRCON.CLASSID = Linux'
CRITERIA = 'STARTTIME >= 080000'
CRITERIA = 'STOPTIME <= 170000'
## Database extract, traditional metrics

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>USERID</th>
<th>ACCTNBR</th>
<th>CPU</th>
<th>Pages</th>
<th>VIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>20100603 080000</td>
<td>REDHAT04</td>
<td>REDHAT</td>
<td>4</td>
<td>99</td>
<td>742</td>
<td></td>
</tr>
<tr>
<td>20100603 080000</td>
<td>REDHAT3</td>
<td>REDHAT</td>
<td>8</td>
<td>170</td>
<td>790</td>
<td></td>
</tr>
<tr>
<td>20100603 080000</td>
<td>REDHAT5</td>
<td>REDHAT</td>
<td>6</td>
<td>83</td>
<td>3138</td>
<td></td>
</tr>
<tr>
<td>20100603 080000</td>
<td>REDHAT5X</td>
<td>REDHAT</td>
<td>11</td>
<td>217</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>20100603 080000</td>
<td>SUSELNX1</td>
<td>SUSELNX</td>
<td>18</td>
<td>46</td>
<td>6985</td>
<td></td>
</tr>
<tr>
<td>20100603 080000</td>
<td>SUSELNX2</td>
<td>SUSELNX</td>
<td>194</td>
<td>15</td>
<td>1067</td>
<td></td>
</tr>
<tr>
<td>20100603 090000</td>
<td>REDHAT04</td>
<td>REDHAT</td>
<td>4</td>
<td>99</td>
<td>731</td>
<td></td>
</tr>
<tr>
<td>20100603 090000</td>
<td>REDHAT3</td>
<td>REDHAT</td>
<td>8</td>
<td>171</td>
<td>797</td>
<td></td>
</tr>
<tr>
<td>20100603 090000</td>
<td>REDHAT5</td>
<td>REDHAT</td>
<td>6</td>
<td>83</td>
<td>3211</td>
<td></td>
</tr>
<tr>
<td>20100603 090000</td>
<td>REDHAT5X</td>
<td>REDHAT</td>
<td>11</td>
<td>218</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>20100603 090000</td>
<td>SUSELNX1</td>
<td>SUSELNX</td>
<td>18</td>
<td>46</td>
<td>6977</td>
<td></td>
</tr>
<tr>
<td>20100603 090000</td>
<td>SUSELNX2</td>
<td>SUSELNX</td>
<td>195</td>
<td>17</td>
<td>1054</td>
<td></td>
</tr>
<tr>
<td>20100603 100000</td>
<td>REDHAT04</td>
<td>REDHAT</td>
<td>15</td>
<td>103</td>
<td>4618</td>
<td></td>
</tr>
<tr>
<td>20100603 100000</td>
<td>REDHAT3</td>
<td>REDHAT</td>
<td>8</td>
<td>173</td>
<td>792</td>
<td></td>
</tr>
<tr>
<td>20100603 100000</td>
<td>REDHAT5</td>
<td>REDHAT</td>
<td>6</td>
<td>84</td>
<td>3172</td>
<td></td>
</tr>
<tr>
<td>20100603 100000</td>
<td>REDHAT5X</td>
<td>REDHAT</td>
<td>11</td>
<td>220</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>20100603 100000</td>
<td>SUSELNX1</td>
<td>SUSELNX</td>
<td>33</td>
<td>46</td>
<td>7190</td>
<td></td>
</tr>
<tr>
<td>20100603 100000</td>
<td>SUSELNX2</td>
<td>SUSELNX</td>
<td>205</td>
<td>20</td>
<td>1288</td>
<td></td>
</tr>
</tbody>
</table>
Linux accounting by “velocity mib”
- CPU by process, capture ratio 100%
  - Requires “prorate” technology
- CPU by Linux Application
- Reported on ESALNXA
- Extracted from performance database

Other virtual platforms need accounting too!

Microsoft, SUN, P-series Accounting by “host” process
- By CPU second, capture ratio 100% for static processes
- Capture ratio for dynamic processes low without Velocity mib
- Reported on ESAHST1
- Extracted from performance database
Linux accounting by “application”

EXTRACT: ; First extract system data

columns = '10 10 10 8 8 8 8'
TITLE = 'STARTTIME  STOPTIME  NODE      APPL  PID'
TITLEc = 'CPU'
TITLE = '---------- --------- --------- -------  ------'
TITLEc = '-------'

INTERVAL = 'SU'
CRITERIA = 'USRTYPE=USER'
CRITERIA = 'USRCON.CLASSID=*Linux'
CRITERIA = 'STARTTIME >= 080000'
CRITERIA = 'STOPTIME <= 170000'
CRITERIA = 'VSINAP.TOTCPU > 0'

X = 'STARTTIME'
Y = 'STOPTIME'
Y = 'NODE'
Y = 'VSINAP.NAME'
Y = 'VSINAP.ID'
Y = 'VSINAP.TOTCPU / 100' ; Turn into CPU Seconds
**Linux Application Accounting**

By server, by application, CPU seconds consumed

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>NODE</th>
<th>APPL</th>
<th>PID</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>*Totals</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>agc2</td>
<td>29188</td>
<td>8</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>nodeage</td>
<td>318</td>
<td>7</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>automou</td>
<td>3028</td>
<td>0</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>cron</td>
<td>3313</td>
<td>1</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>kernel</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>multipa</td>
<td>24607</td>
<td>4</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>sendmai</td>
<td>3154</td>
<td>0</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>snmpd</td>
<td>30961</td>
<td>3</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>*Totals</td>
<td>0</td>
<td>170</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>.opende</td>
<td>3340</td>
<td>0</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>ace</td>
<td>28947</td>
<td>8</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>agenten</td>
<td>18597</td>
<td>7</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>auditau</td>
<td>26553</td>
<td>13</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>a2adata</td>
<td>25244</td>
<td>19</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>comm_um</td>
<td>15950</td>
<td>11</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>c2cb_mi</td>
<td>20312</td>
<td>3</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>ezacquii</td>
<td>4612</td>
<td>9</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>isodata</td>
<td>28948</td>
<td>7</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>nodeage</td>
<td>9756</td>
<td>15</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>webshee</td>
<td>22130</td>
<td>15</td>
</tr>
</tbody>
</table>
Websphere argument string used for accounting

```bash
wasadmin 27144 6846 0 Feb06 ? 00:43:13 /u01/was61/java/bin/java -Dclipse.security -Dwas.status.socket=34229 -Dosgi.install.area=/u01/was61 -Dosgi.configuration.area=/u01/was61/profiles/appsrv/configuration -Dosgi.framework.extensions=com.ibm.cds -Xshareclasses:name=webspherev61_%g,groupAccess,nonFatal -Xscmx50M -Xbootclasspath/p:/u01/was61/java/jre/lib/ext/ibmorb.jar:/u01/was61/java/jre/lib/ext/ibmext.jar:/u01/J2EEProbe/DiagnosticsAgent/classes/IBM/1.5.0/instr.jre:/u01/J2EEProbe/DiagnosticsAgent/classes/bootstrap -classpath /u01/was61/profiles/appsrv/properties:/u01/was61/properties:/u01/was61/lib/startup.jar:/u01/was61/lib/bootstrap.jar:/u01/was61/lib/j2ee.jar:/u01/was61/lib/Improxy.jar:/u01/was61/lib/urlprotocols.jar:/u01/was61/deploytool/itp/batchboot.jar:/u01/was61/deploytool/itp/batch2.jar:/u01/was61/java/lib/tools.jar -Dibm.websphere.internalClassAccessMode=allow -verbose:gc -Xms1024m -Xmx1200m -Dws.ext.dirs=/u01/was61/java/lib:/u01/was61/profiles/appsrv/classes:/u01/was61/classes:/u01/was61/lib/u01/was61/installedChannels:/u01/was61/lib/ext:/u01/was61/web/help:/u01/was61/deploytool/itp/plugins/com.ibm.etools.ejbdeploy/runtime -Dderby.system.home=/u01/was61/derby -Dcom.ibm.itp.location=/u01/was61/bin -Djava.util.logging.configureByServer=true -Duser.install.root=/u01/was61/profiles/appsrv -
```
Linux accounting by DISK Size, Space

EXTRACT: ; First extract system data

columns = '10 8 8 10 10'
TITLE = ' NODE      disk     size(MB)   Used(MB)'
TITLE = ' --------- -------  ---------  --------'

INTERVAL='SU'
CRITERIA='STARTTIME >= 080000'
CRITERIA='STOPTIME <= 170000'
CRITERIA='HSTMEM.ALLOCUN > 1024'
CRITERIA='HSTMEM.SIZE > 0'

X = 'NODE'
Y = 'HSTMEM.DESCR'
Y = 'HSTMEM.SIZE / 1024' ; Convert to MB
Y = 'HSTMEM.USED / 1024' ; Convert to MB
## Linux disk accounting metrics

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>NODE</th>
<th>disk</th>
<th>size(MB)</th>
<th>Used(MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>/</td>
<td>272</td>
<td>106</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>/usr</td>
<td>1203</td>
<td>583</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>/opt</td>
<td>157</td>
<td>2</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>/webdat</td>
<td>4032</td>
<td>3131</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>/var</td>
<td>496</td>
<td>103</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>/home</td>
<td>248</td>
<td>67</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>/usr/op</td>
<td>4032</td>
<td>3131</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>/usr/Xl</td>
<td>4032</td>
<td>3131</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>/local</td>
<td>248</td>
<td>71</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>/boot</td>
<td>51</td>
<td>18</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS009</td>
<td>/u01</td>
<td>12095</td>
<td>2467</td>
</tr>
</tbody>
</table>
Linux accounting by network activity by server

```plaintext
EXTRACT: ; First extract system data

columns = '10 8 8 10 10'
TITLE = ' NODE      Descr    OutKB/Sec  InKb/Sec  Time(Secs)'
TITLE = ' --------- -------  ---------  --------  ----------'

INTERVAL='SU'
CRITERIA='STARTTIME >= 080000'
CRITERIA='STOPTIME <= 170000'
CRITERIA='IFTAB.type = 6'
CRITERIA='IFTAB.INOCTETS > 1000'

X = 'NODE'
Y = 'IFTAB.DESCR'
Y = 'IFTAB.OUTOCTETS / 1024'
Y = 'IFTAB.INOCTETS / 1024'
Y = 'IFTAB.SECONDS'
```
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>NODE</th>
<th>Descr</th>
<th>OutKB/Sec</th>
<th>InKb/Sec</th>
<th>Time(Secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20090910</td>
<td>092200</td>
<td>TCPIP</td>
<td>ETHERNE</td>
<td>30</td>
<td>82</td>
<td>3480</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS029</td>
<td>eth1</td>
<td>16</td>
<td>4</td>
<td>3480</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXASF25</td>
<td>eth1</td>
<td>16</td>
<td>5</td>
<td>3480</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXASA09</td>
<td>eth1</td>
<td>5</td>
<td>1</td>
<td>3480</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS019</td>
<td>eth1</td>
<td>23</td>
<td>3</td>
<td>3480</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS037</td>
<td>eth1</td>
<td>6</td>
<td>4</td>
<td>3480</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS021</td>
<td>eth1</td>
<td>7</td>
<td>2</td>
<td>3481</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS013</td>
<td>eth1</td>
<td>41</td>
<td>11</td>
<td>3481</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS031</td>
<td>eth1</td>
<td>25</td>
<td>18</td>
<td>3481</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS025</td>
<td>eth1</td>
<td>8</td>
<td>2</td>
<td>3480</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS002</td>
<td>eth1</td>
<td>13</td>
<td>2</td>
<td>3481</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS005</td>
<td>eth1</td>
<td>18</td>
<td>3</td>
<td>3480</td>
</tr>
<tr>
<td>20090910</td>
<td>092200</td>
<td>LNXAS011</td>
<td>eth1</td>
<td>8</td>
<td>2</td>
<td>3481</td>
</tr>
</tbody>
</table>
Many options, must focus on business drivers
- Determine costs
- Data provided by zVPS for resources utilized

VM Based Accounting
- CPU by virtual machine/Server, capture ratio 100%
- Storage residency
- Virtual I/O

Linux based accounting
- CPU by process, application, user
- DISK size and use by server
- Network by server

Distributed server data also collected for Accounting