z/VM Charge Back Models and Methods

v7

Bill Bitner
z/VM Development Client Focus & Care
IBM Endicott – bitnerb @ us.ibm.com

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM*
IBM Logo*
DB2*
Dynamic Infrastructure*
GDPS*
HiperSockets
Parallel Sysplex*
RACF*
System z*
System z10*
Tivoli*
z10 BC
z9*
z/OS*
z/VM*
z/VSE
zEnterprise*
System z196
System z114
System zEC12
System zBC12

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

OpenSolaris, Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
INFINIBAND, InfiniBand Trade Association and the INFINIBAND design marks are trademarks and/or service marks of the INFINIBAND Trade Association.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs):

Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”).

No other workload processing is authorized for execution on an SE.

IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Agenda

- Introduction
  - Total Cost of Acquisition, Ownership, and Computing
  - Profit or Cost
  - What is a charge back model?
  - What are considerations to creating one?

- Survey of Different Models

- Methods and approaches to capturing the required data
  - z/VM Accounting Data
  - z/VM Monitor Data

Total Cost

- Charge back models should be based on costs
  - Total Cost of Acquisition (TCA): one time cost of acquiring capital equipment and resources for the solution.
  - Total Cost of Ownership (TCO): on-going costs to operate and maintain the solution.
  - Total Cost of Computing (TCC): the cost on a per unit of work, cost of transaction, cost of a job, cost of a unit manufactured or shipped.

- There is value in Total Cost analysis even if not used to do charge back.
  - Knowing where the money goes
  - Understanding return on investment
  - Understanding where the potential for cost efficiencies lay

- Value from Total Cost in Charge Back Models
  - Total costs can then be used to help determine the charge back rate
  - Helps validate fairness by associating area of greatest cost with those using it

The Big Question: Share Wealth?

- As more work is brought onto the platform the TCO on a per virtual machine or database drops. Do you pass that on to your users?

1. Set it up so rates go down as tenancy increases.
   - Fair to the first ones on the platform?
2. Compute what the costs would be for fully loaded system and charge that on day one, eating the other costs.
3. Continue to charge the initial rate and be profitable.
4. Continue to charge the initial rate and invest the *profits* in funding advancements and growth that bring in new workload or benefit existing workloads.

Questions to Consider

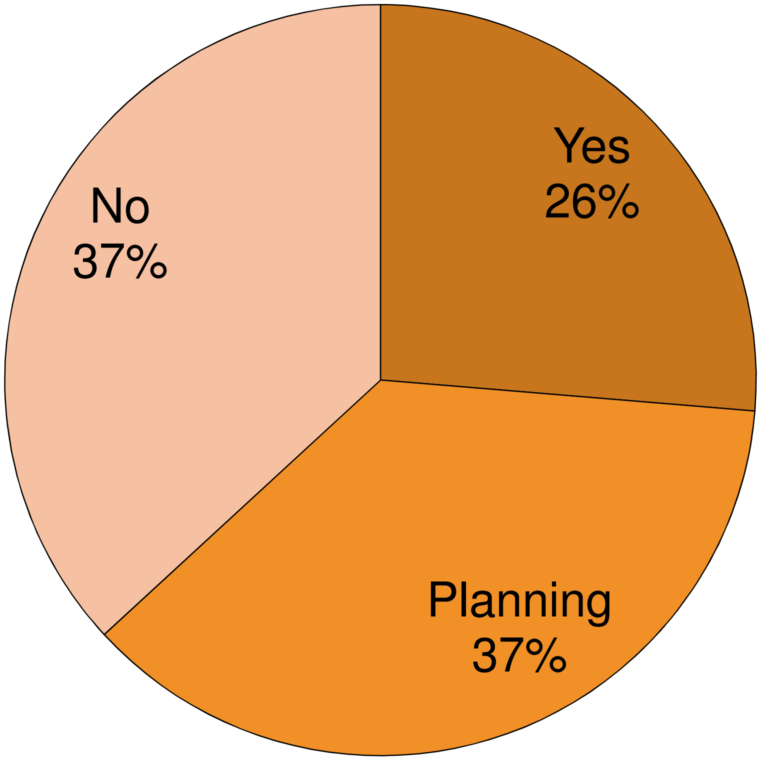
- Other reasons to do charge back?
  - To be fair to different groups
  - To capture value of your services
  - To control or influence behavior

- What's important to your users?
  - Predictability and consistency
    - What constitutes "fairness"?
  - Flexibility
  - Ease
  - Control

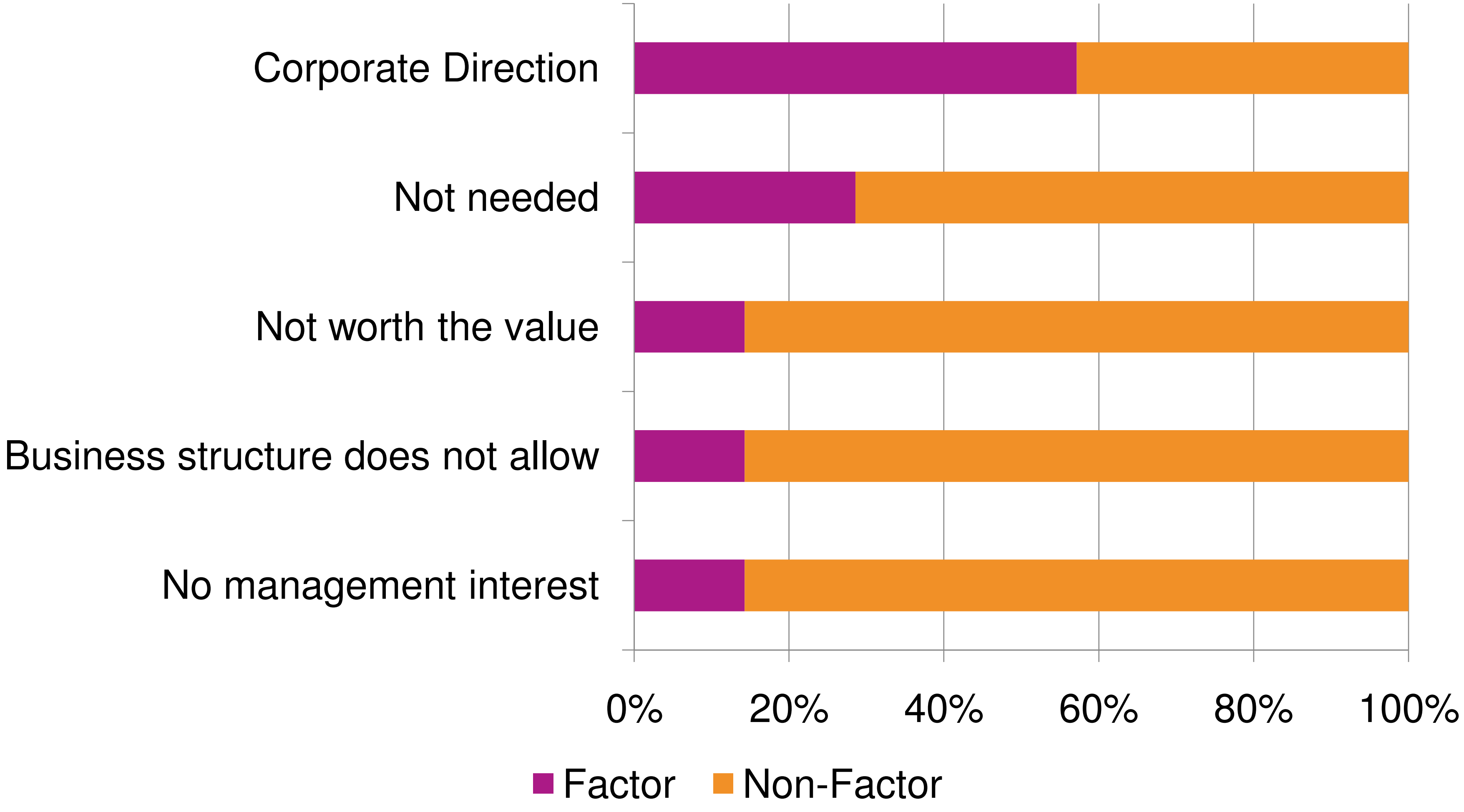
- IT strategy?
  - Cost recovery
  - Profit

What are Customers Doing?

Customer Charge Backs

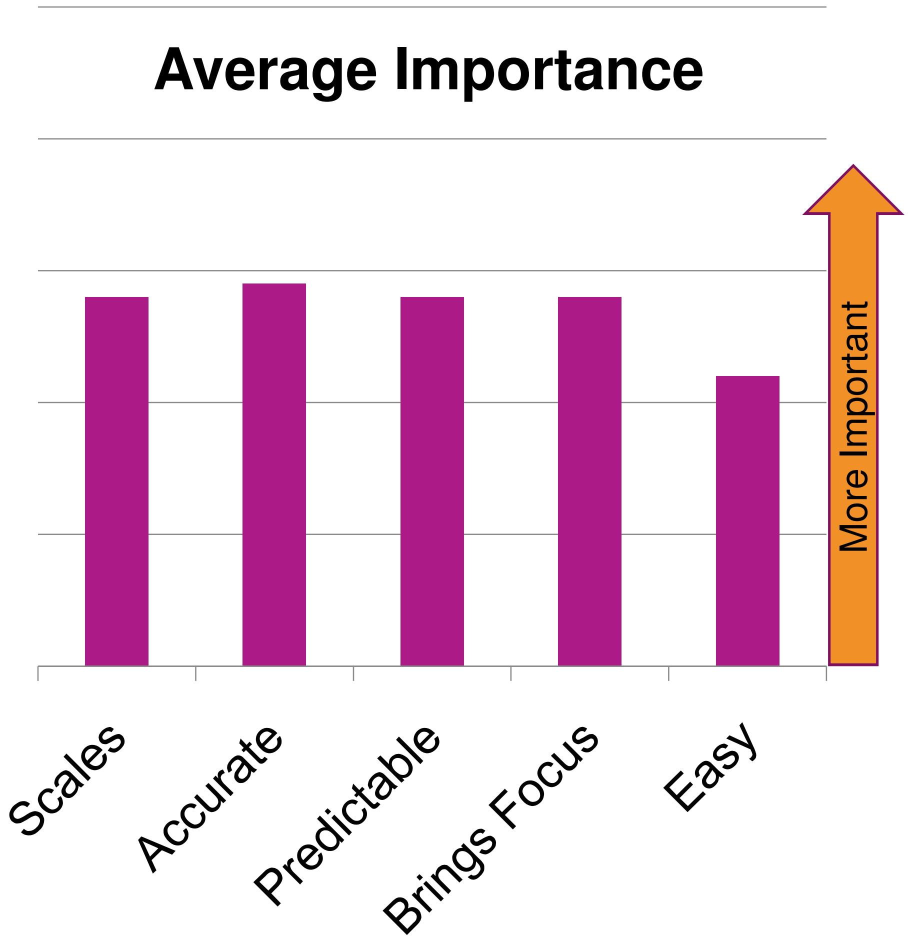


If Not, Why Not?

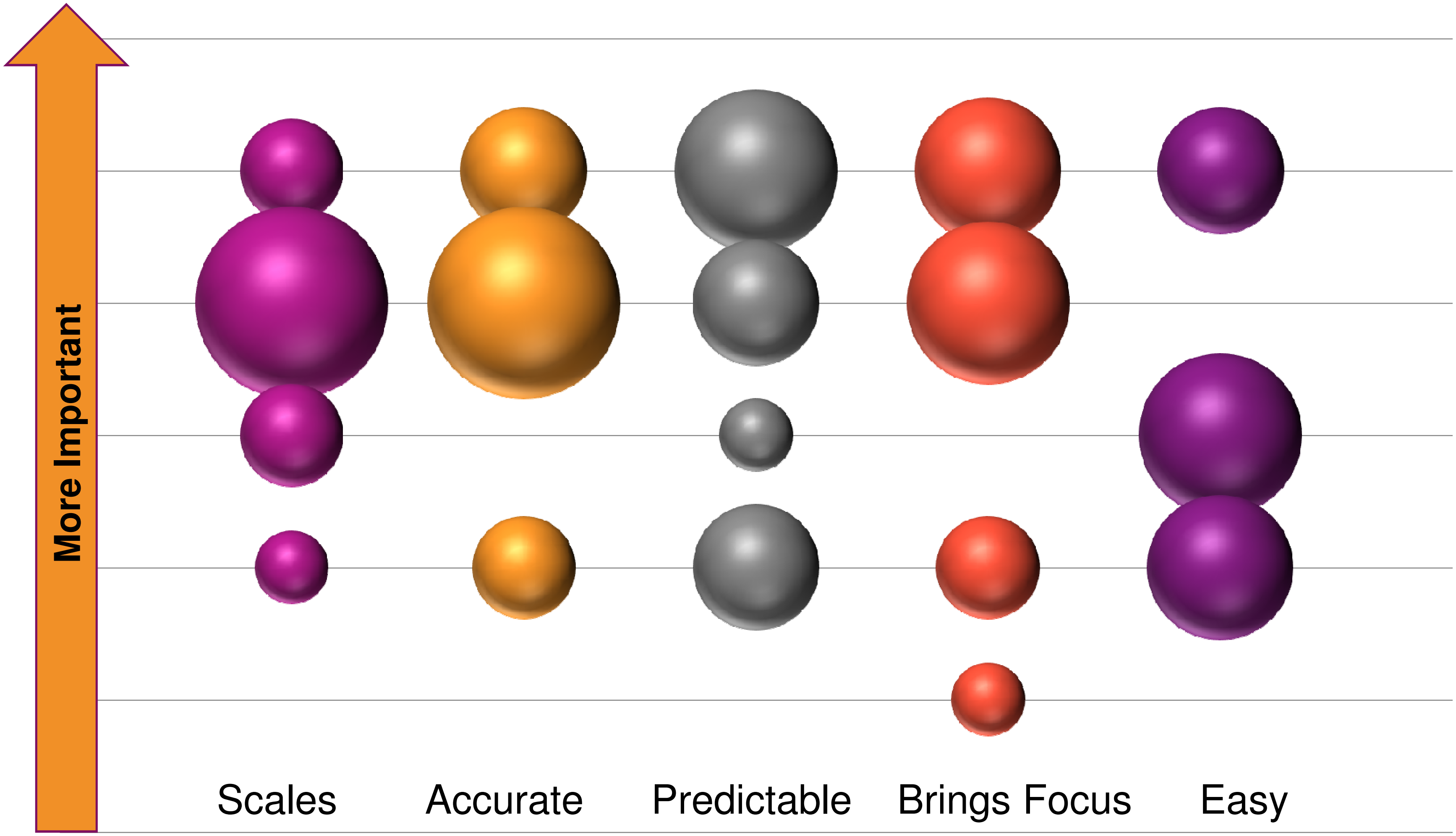


What’s Important to a Model Choice?

- **Scales:** will the model allow for dealing with 100s or 1000s of virtual machines? What’s the overhead?
- **Accurate:** does it associate the costs with the proper virtual machines or applications?
- **Predictable:** Is it predictable for end users? Will they have some knowledge or expectation of their bill for each month or quarter?
- **Brings Focus:** Does it help to bring focus to the value of improving efficiency and TCO?
- **Easy:** Is it a model that is easy to implement in effort or in time?



Breakdown of Criteria



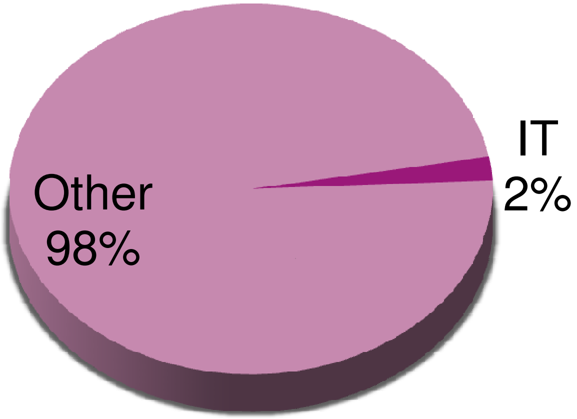
Charge Back Models

Models Examined

1. Pure Cost Center
2. Service Cost Allocation
3. Virtual Machine Flat Charge
4. T-Shirt Size Charge
5. Virtual Machine Usage Based
6. Linux Process Usage Based
7. Fix Resource Contract

Pure Cost Center - Description

- Treats IT as a pure Cost Center
- No charge back or recovery
- Collect costs for overall financial reporting
  - “IT is n% of Corporate Expenses”



Pure Cost Center – Considerations

- Strengths
  - Little for IT to implement
  - Very predictable for users

- Weaknesses
  - Little incentive for users or systems to be efficient
  - Focus on strategic decisions is often on expenses, and not on the potential value
  - Actual cost of end user services not as accurate
  - Can make IT a target for cuts
  - Can make initial z Systems projects look expensive due to larger base costs

Service Cost Allocation - Description

- View IT as a cost center, that is you provide services but don’t generate revenue directly

- Add up all the costs: HW, SW, Personnel, etc.

- Distribute across the users of the services

- How to distribute those costs for departments or lines of business?
  - Percentage of virtual machines
  - Percentage of business or revenue
  - Percentage of users supported
  - Percentage of overall processor time used
  - Some other straight forward calculation

Service Cost Allocation – Considerations

- Strengths
  - Fairly simple to implement
  - Allows some focus on value versus expense

- Weaknesses
  - Agreement on distribution method and fairness
  - Incentives for efficiencies not as obvious
  - Initial projects could be more expensive as base costs are distribute across fewer users

Virtual Machine Flat Charge - Description

- Create standard virtual machines with a fixed number of virtual processors, memory, and storage.
- Treats virtual servers like commodity servers or hardware.
- The 'basic' virtual machine would need to be large enough to contain largest workloads
  - One size has to fit all or performance has to be limited
- No need to track usage of virtual machines for charge back reasons
- Responsibility to ensure sufficient resources

$1 per Machine

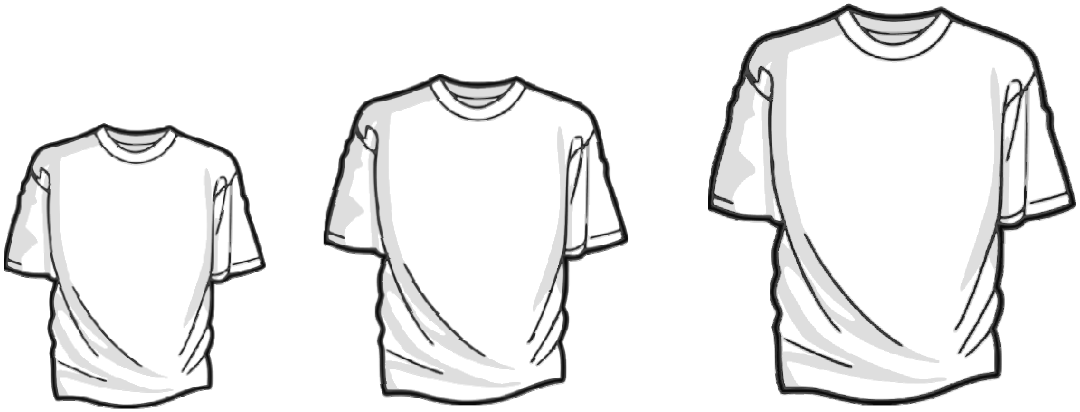
Virtual Machine Flat Charge - Considerations

- Strengths
  - Implementation fairly straight forward
  - Internal revenue for virtual resources instead of real resources
  - Simple and predictable for user

- Weaknesses
  - No incentive by users for efficient use of resources or to tune their systems
  - Loophole of no limit for how much work you put into a virtual machine
    - Example: Just keep adding JVMs in same virtual machine
  - Difficult to do capacity planning
  - Need to oversize virtual machines
    - Can cause other problems

T-Shirt Sizes - Description

- Variation of the Virtual Machine Flat Size
- Have various ‘sizes’ of virtual machines to select with a charge varies with the size
- Sizes normally reflect both processor and memory requirements.



T-Shirt Sizes - Considerations

- Strengths
  - Implementation fairly straight forward
  - Internal revenue for virtual resources instead of real resources
  - Simple and predictable for user
  - More fair than virtual machine flat charge

- Weaknesses
  - No incentive by users for efficient use of resources or to tune their systems
  - Loophole of no limit for how much work you put into a virtual machine
    - Example: Just keep adding JVMs in same virtual machine
  - Difficult to do capacity planning

Variation for Flat and T-Shirt Sizes

- Earlier models were based on virtual machines built
- Variation would be to charge based on logged on time
- Fairly easy to capture and gives incentive for virtual machines to be shutdown when not in use.

Virtual Machine Usage Based - Description

- Charge back based on absolute real resource usage
  - Processor
  - Memory
  - Network
  - Storage

- Virtual Machine Level
  - Typically done at this level
  - Data is easily gathered from z/VM

Virtual Machine Usage Based - Considerations

- Strengths
  - Flexible – pay as you go
  - Tracks back to cost recovery
  - Various tools exist to facilitate this approach
  - Incentive for tuning and improving efficiency

- Weaknesses
  - More complex than other approaches
  - Need method to adjust for loops or runaway scenarios

Linux Process Usage Based - Description

- Charge back based on absolute real resource usage
  - Processor
  - Memory
  - Network
  - Storage

- Linux Process Level
  - Valuable when multiple applications in a virtual machine (though that has other downsides)
  - Requires pulling Linux process data

Linux Process Usage Based - Considerations

- Strengths
  - Flexible – pay as you go
  - Tracks back to cost recovery
  - Handles virtual machines with mixed applications
  - Incentive for tuning and improving efficiency

- Weaknesses
  - More complex than other approaches, need to capture process level information
  - Need method to adjust for loops or runaway scenarios

Individual Resource Limit Based - Description

- Paid for a fixed limit or capacity of resources, typically processor usage

- Create a cap or limit for individual virtual machines via SHARE ABSOLUTE LIMITHARD

- Determine the correct ‘limit’
  - MIPS
  - Percentage of an IFL

Individual Resource Limit Based - Considerations

- Strengths
  - Predictable for the end user
  - Could ‘purchase’ different amounts based on heavier periods of the activity across the year

- Weaknesses
  - Computing a value (MIPS) on which everyone can agree
  - Having to set individual LIMITHARD for virtual machines
    - Percentage varies if you change number of logical processors
    - Multiple virtual machines may be needed or not needed at times
  - Difficult for workloads that vary greatly
  - Unused resources, while paid for, are wasted

Group Resource Limit Based - Description

- Like the Individual Resource Limit Model but for a set of virtual machines on a given z/VM system

- Paid for a fixed limit or capacity of resources, typically processor usage

- Create a cap or limit for individual virtual machines via CPU pools

- Determine the correct ‘limit’
  - MIPS
  - Percentage of an IFL

Group Resource Limit Based - Considerations

- Strengths
  - Predictable for the end user
  - Could ‘purchase’ different amounts based on heavier periods of the activity across the year
  - Purchase for a group of virtual machines without need to know which will be used

- Weaknesses
  - Computing a value (MIPS) on which everyone can agree
  - Difficult for workloads that vary greatly
  - Unused resources, while paid for, are wasted

Variations and Other Options

- Provide ‘discount’ for resources used off-shift or off-peak hours

- T-shirt model with service goals
  - Example:
    - $ Discount Machine: Low Limithard Share Setting
    - $$ Standard Machine: Limitsoft Share Setting
    - $$$ Gold Machine: No limit share

Quality of Service Charges

- Is there special technology or quality of service that costs you more to provide?

- Is there special technology or quality of service that your end users would pay to have?

- Examples:
  - 365 x 7 Support
  - Disaster Recovery
  - Business Continuity
    - Up during planned outages (Live Guest Relocation)
  - Secure end of life cycle management

Quality of Service Based Charge - Considerations

- Strengths
  - Allows recovering costs for perhaps expensive services
  - Allows highlighting of services perhaps not available on other platforms
- Weaknesses
  - Incentive for people to avoid services that could be important due to price

Other Considerations

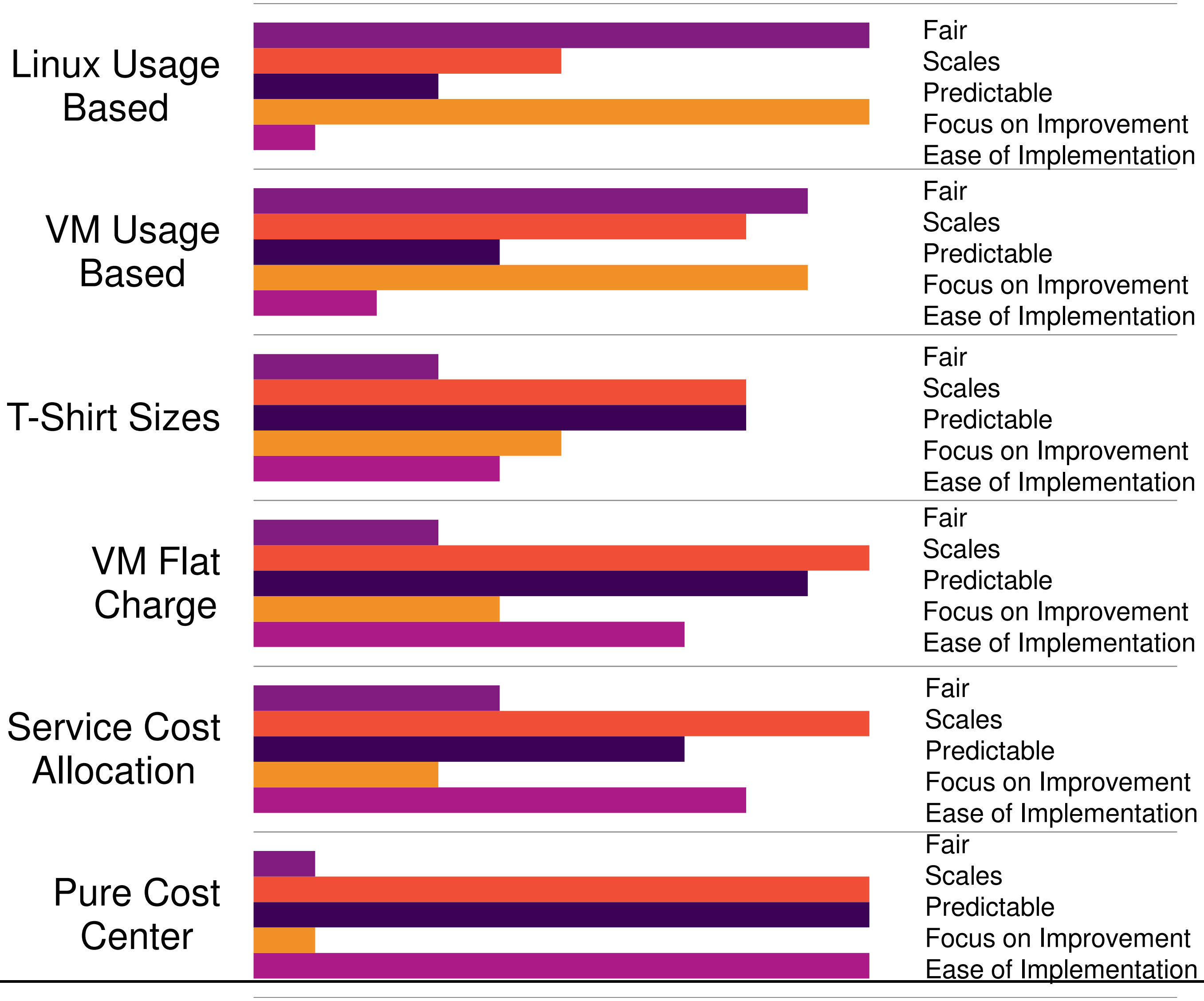
- Put yourself in your user's position. How would you (they) try to work the system? What loop holes are there? What would you not like about it?

- Where are your costs in terms of managing the system, particularly in people time?
  - Build? Capacity Planning? Network modifications?

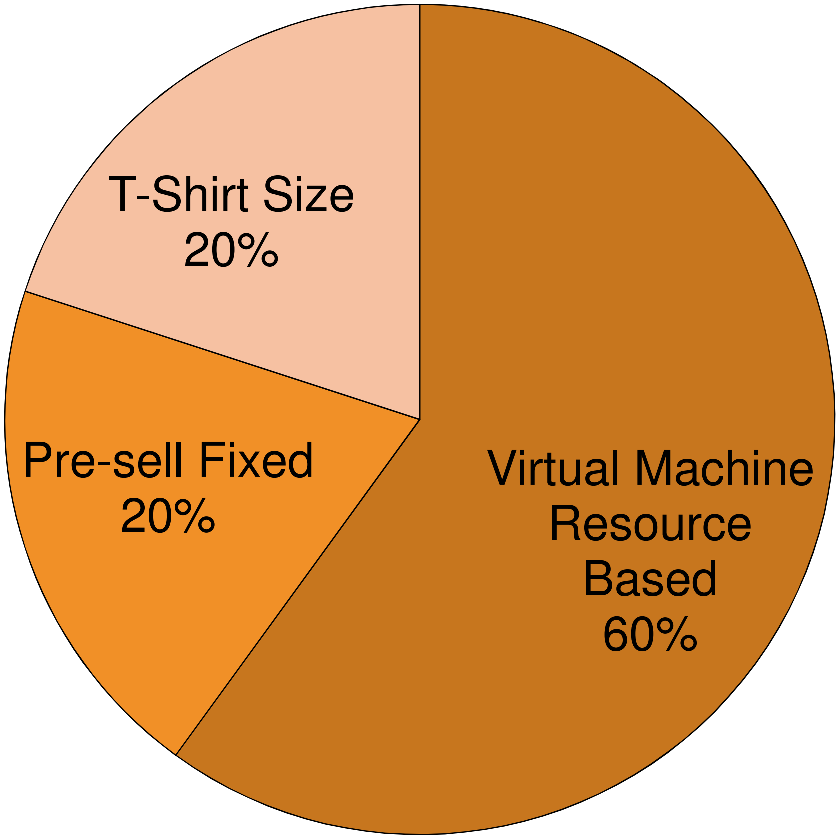
- Ask about the dynamic nature of the environment.
  - If cost recovery from the systems team is important, than understanding what can change and how frequently is valuable.

- How large will the environment scale?
  - Some approaches are easy for a few dozen guests, but become more challenging when talking about 100s of virtual machines.

Summary of Models



What Models are Customers Using?



Methods

Billing or Charge Back in z/VM Environment

- z/VM has interfaces that allow for performance and accounting data to be generated and collected. Examples of this information includes:
  - Processor time consumed
  - Processor or capacity available
  - Memory usage and activity
  - I/O activity
  - Network activity and capacity

- IBM and other ISVs provide tools and products to reduce and report this information with various degrees of sophistication

- Information can be grouped or aggregated via accounting numbers or other information.

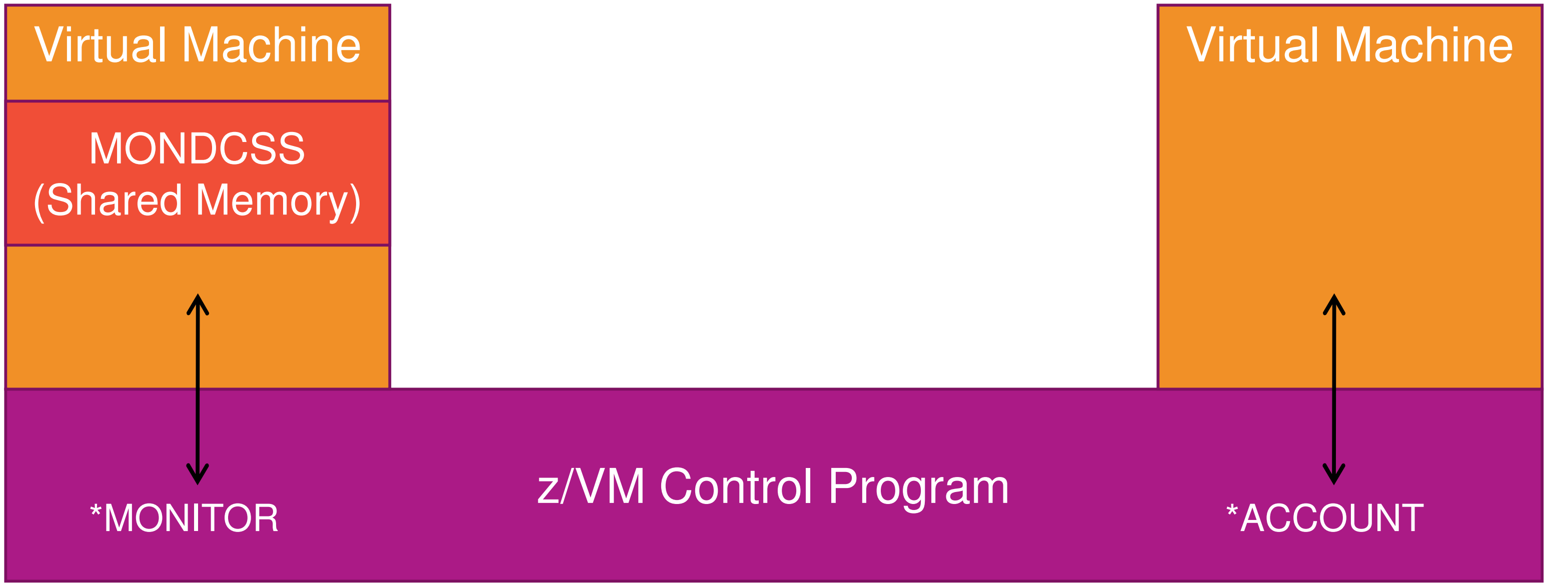
Primary Data Sources

- z/VM Monitor System Service
  - Monitor Records
  - Scales really well
  - Lots of data

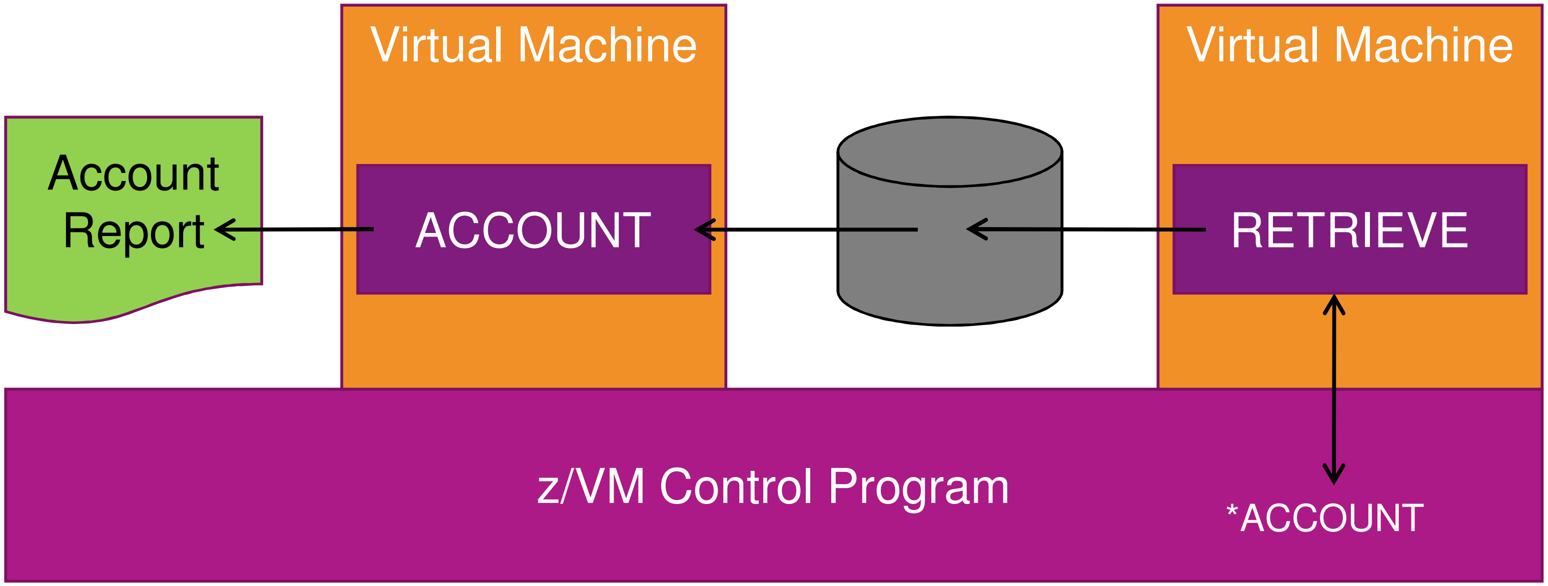
- z/VM Accounting System Service
  - Accounting Records
  - Exact, uses interlocked instructions
  - Smaller amount of data

- Linux Performance or Accounting Data

Overview

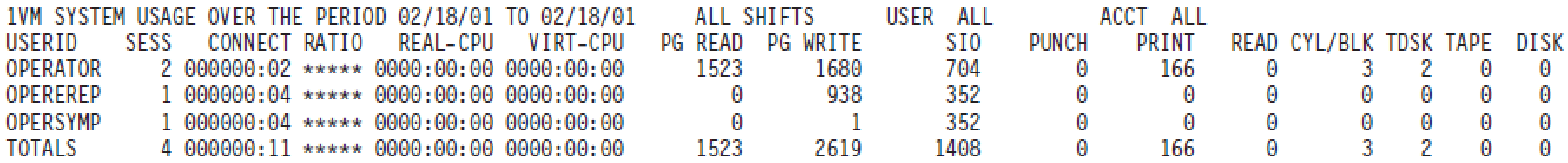

Virtual Machine
MONDCSS
(Shared Memory)
Virtual Machine
*MONITOR
z/VM Control Program
*ACCOUNT

Accounting Records



ACCOUNT Utility

- Included in z/VM and documented in CMS Commands and Utilities Reference
  - http://publibz.boulder.ibm.com/epubs/pdf/hcsd8c20.pdf

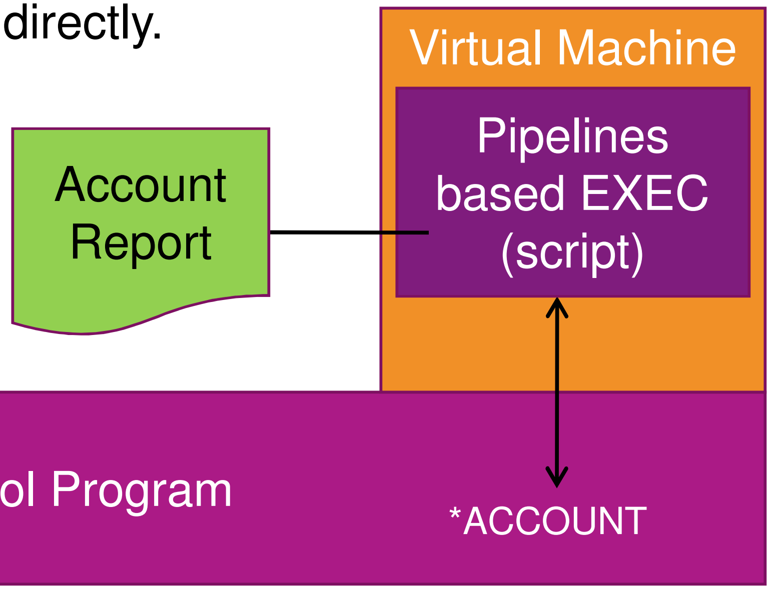


```
1VM SYSTEM USAGE OVER THE PERIOD 02/18/01 TO 02/18/01     ALL SHIFTS      USER  ALL         ACCT  ALL
USERID    SESS   CONNECT RATIO   REAL-CPU   VIRT-CPU   PG READ  PG WRITE       SIO    PUNCH    PRINT   READ CYL/BLK TDSK TAPE  DISK
OPERATOR     2 000000:02 ***** 0000:00:00 0000:00:00      1523      1680       704        0      166      0       3    2    0    0
OPEREREP     1 000000:04 ***** 0000:00:00 0000:00:00         0       938       352        0        0      0       0    0    0    0
OPERSYMP     1 000000:04 ***** 0000:00:00 0000:00:00         0         1       352        0        0      0       0    0    0    0
TOTALS       4 000000:11 ***** 0000:00:00 0000:00:00      1523      2619      1408        0      166      0       3    2    0    0
```

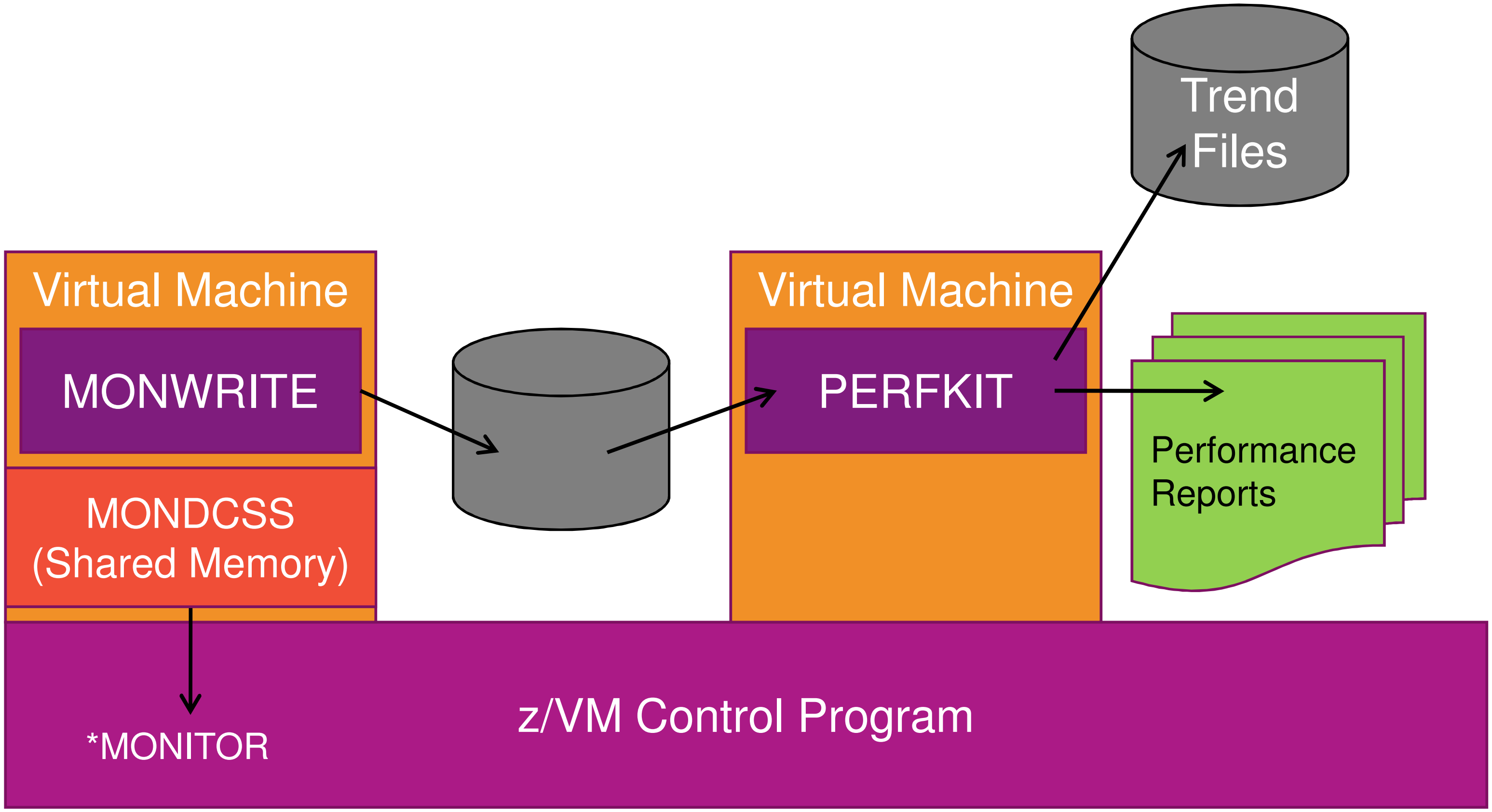
Accounting Data through CMS PIPELINES

- ➢CMS Pipelines is a powerful programming model in z/VM that can be used in REXX programs.

- ➢Included in Pipelines is a stage called STARSYS that allows one to connect to *ACCOUNT directly.



Monitor Record Processing



Performance Toolkit for z/VM

- Priced featured of z/VM
- Real time monitor
  - Optionally create human readable reports on regular intervals (each shift, once a day, etc)
- Batch processor of raw Monwrite files
- Human Readable Reports
- Summary/Trend/History files
- Very robust data covering z/VM level in great detail

CSVGEN for Performance Toolkit

- Available on z/VM Home Page

- Converts Performance Toolkit Summary/Trend/History files to CSV files for import into spreadsheets or databases

- http://www.vm.ibm.com/related/perfkit/csvgen.html

- Process:
  - Take Toolkit files
  - CSVGEN *format toolkit_file   output_fm   prefix (options*
    - *Format:* **S**ummary or **T**rend
    - *Prefix:* prefix the output file names

IBM Smart Cloud Cost Management

- Formerly known as ITUAM – IBM Tivoli Usage and Accounting Manager
- SCCM supports both z/VM and Linux for z Systems as well as:
  - Windows®, VMware, AIX®, Linux, z/OS
- Uses z/VM Accounting records to create CSR (Common Source Resource) files which is its standard input
- z/VM based reports cover:
  - Connect Time
  - CPU Time
  - Virtual I/O (traditional SSCH based)
  - Unit record information (spool)
- Linux data includes process information:
  - System and user processor usage
  - Blocks read/written
  - Average memory usage

OMEGAMON XE

- Tivoli Data Warehouse (TDW) component captures and stores performance data

- Various reporting tools available with TDW

- Any report can be exported as a CSV file

Summary

Summary

- If you'd like to add to the charge back survey, see:
  - https://ibm.biz/bitchargebacksurvey
- Various models exist
  - Need to understand the pros and cons of each
  - Need to think how the model will influence behaviour
- Various methods exist to implement the models
  - Create your own
    - z/VM Monitor Data
    - z/VM Accounting Data
    - Linux Performance and Accounting Data
- IBM Products and Tools
  - ACCOUNT Utility
  - Performance Toolkit for z/VM
  - OMEGAMON XE
  - IBM Smart Cloud Cost Management

References

- Red paper on Accounting and Monitoring for z/VM Linux guest machines
  - http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp3818.html
  - A little dated but includes some examples for doing your own thing

- Red book on ITUAM
  - http://publib-b.boulder.ibm.com/abstracts/sg247404.html?Open