



Capacity Estimation for Linux Workloads



David Boyes
Sine Nomine Associates

Agenda

- General Capacity Planning Issues
- Virtual Machine History and Value
- Unique Capacity Issues in Virtual Machines
- Empirical Impact and Observations
- Measurements, Benchmarks, and Methods, Oh My!
- Case Study of a Real Deployment
- References

Caveats

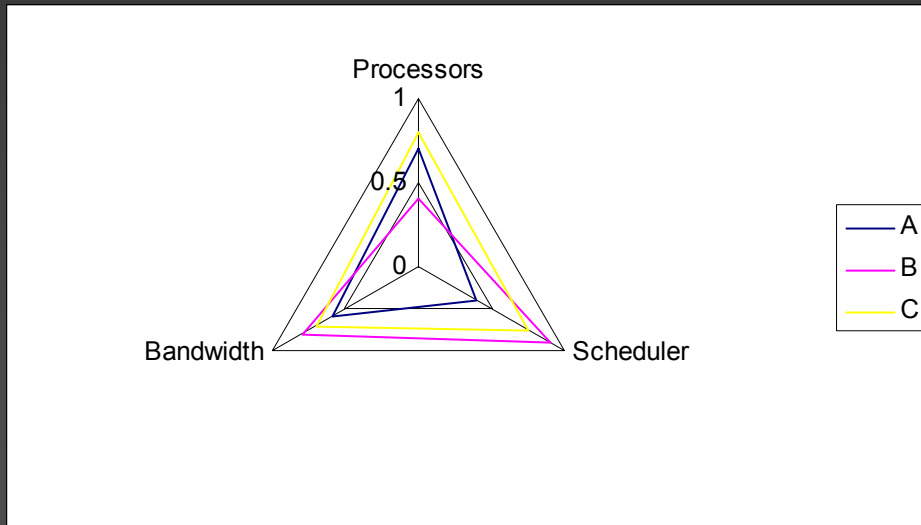
- Capacity planning is at best an educated guess about what might happen if something happens, plus an error term
 - Statistics are voodoo
 - Work in progress
 - Your results will vary
 - Etc, etc, etc
-

General Capacity Planning Issues

- Relative vs Absolute
- Measuring Workload
- Workload Composition
- Utilization
- “White Space”

Relative Vs Absolute

- Balance of:
 - Processor power (effective work per instruction)
 - Internal bandwidth (data rate between cache and RAM)
 - Scheduler efficiency (processor utilization typical to this system type)



Perfect World:
If equal stress in
test, relative
capacity is
geometric mean
of measurements

Measuring Workload

- If workload shifts away from balance, relative capacity is always affected.
- Server workload tends to be skew-rich
- CPU intense applications tend to favor more/faster CPUs (skewless)
- Mixed workloads/non uniform usage patterns favor higher internal bandwidth and better scheduling (skew/spiky)

Workload Summary

- Relative capacity of machines differs by workload:
 - There is no single solution
 - “Right Tool, Right Job”

Utilization

- Typical server farm is deliberately underutilized most of the time (peak load allowance, or “white space”)
- White space is 1 minus the utilization.
- Sources of White Space
 - Spikes
 - Headroom
 - Redundancy
 - Fragmentation
 - Partitioning
 - Skew

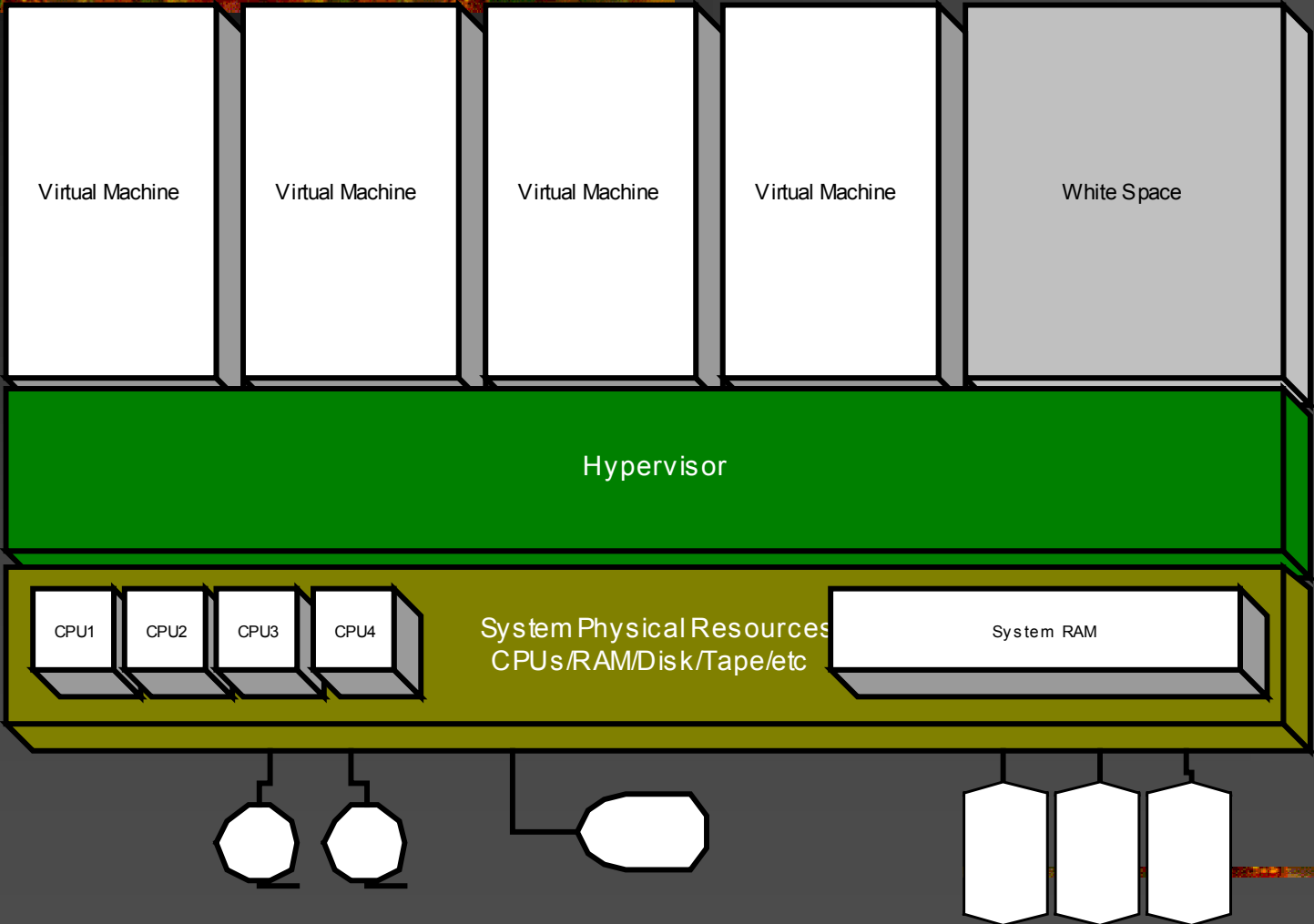
Utilization

- Be careful when you talk about utilization (peak vs average).
 - Utilization figures are a statistical sample, thus subject to manipulation
 - Short time samples tend to poorly represent the utilization of a machine
 - Fine granularity impacts figures (Heisenburg)
 - Corollary: utilization often depends on workload – combined workload smoothing often can optimize utilization over time

Virtual Machines: History and Value

- Not a new idea
 - OS/8 for the PDP-8
 - IBM VM family (1964)
- Commonplace in large systems deployments for more than 3 full decades
- Simple concept: allow single physical machine to emulate a dedicated system for each individual user
 - Each emulated system receives a portion of the physical resources of the underlying hardware
 - The emulated system does NOT have to match the physical configuration of the system
 - Emulated systems are independent of each other.

Graphical View of Virtual Machine Environment



Unique Capacity Issues

- I/O Overlapping
- CPU and FPU Overlapping
- Memory Overlap
- Virtual Device Emulation Overhead

I/O Overlap

- Virtual machines allow overlapping usage of I/O devices based on individual virtual machines being statistically unlikely to be performing simultaneous I/O.
- This overlap can affect performance positively in most cases – in most applications, the lions share of cycles are wasted waiting on external I/O to complete.

CPU/FPU Overlap

- CPU overlap occurs when a virtual machine suspends to complete a blocking operation such as I/O and cannot use its full timeslice, releasing it to be available for other virtual machines to use as needed.
- Unused space allows additional “overlap” cycles to be utilized according to demand

Memory Overlap

- This is not particularly different from any other virtual memory value proposition, however, the scale is somewhat larger – entire systems instead of processes within a single system.
- Storage overload/overlap factors are generally much higher (60-80%) in virtual machines

Device Emulation Overlap

- Permitting devices to be sub-allocated (simulating full disks) allows significant cost savings
- Consider normal workstation deployment: Every machine has:
 - Operating system files – all the same in most environments, so why not share?

Empirical Impact for System Sizing

- Standard sizing models for Unix and traditional large systems grossly overestimate the requirement for CPU and memory.
 - Most server farm machines operate at less than 10% nominal utilization
 - Most server farms have less than 20% of systems active at any specific interval
- There is no valid direct comparison for disparate architectures – the numbers don't necessarily mean the same for different systems
 - Meaningless/Misleading Indicator of Processor Speed (MIPS)
 - Overall instruction throughput for effective work differs on RISC/CISC

Empirical Observations

■ Smaller is Better

- For most virtual machine hypervisors, the contents of the virtual machine is a opaque black box.
 - Optimizing for minimum size impermeable boxes allows better control of hypervisor resources
 - Smaller scheduling units
 - Less storage/disk paging fragmentation
 - Less paging/swapping latency
 - Maximum benefit for block paging algorithms
 - Best opportunity for maximum I/O and CPU overlap
-

Empirical Observations

- Scaling via Horizontal Additions
 - Corollary of Smaller is Better
 - Applications best suited to virtual machine environment scale by addition of multiple virtual instances
 - Statistical sampling of workload spreads impact of load across multiple virtual CPU engines
 - Minimizes deadlock opportunities
 - Increases requirement for synchronization
 - Does awful things to CPU cache consistency in hypervisor
 - Augments demand for sophisticated system/configuration management tools

Empirical Observations

- Hypervisor impact biases application selection toward I/O intensive applications
 - Self-overlapping effect of I/O operations as unit transactions at the hypervisor level
 - Allows dispatch of next VM at I/O request block boundary and more effective “fair share” algorithms
 - Rapid context switch capability imperative for efficient hypervisor management
 - Intel and Sparc not strong in this area without careful thought
 - Strong benefit for S/390 architecture

Empirical Observations

- Virtual machines demand LESS hardware resources for most server-class tasks due to overlap.
 - Many applications spend up to 60% of CPU time in I/O wait/spin locks
 - Context switch/dispatch/CPU pipeline effect can demonstrate 35-45% overlap for CPU
 - Corollary: You don't need 900 Mhz systems to deliver equivalent performance – apples/pears comparison
 - Corresponding impact on effective transaction throughput – more transactions with 35-45% smaller systems
 - Reduced hypervisor overhead

What About Benchmarks?

- Benchmarks are by definition arbitrary, and are usually misleading across disparate hardware.

What Can We Rely On?

- Workload Analysis
- Overlap Analysis
- Error Estimation Terms

“In capacity planning, the answer is always ‘*it depends*’. With virtual systems and Linux, ‘*it depends even more*’.”

-- Bill Bitner, IBM

Analysis Framework

- Instrumentation
- Application Classification
- Data Model
- Overlap Analysis
- Error Estimation

Instrumentation

- Internal Instrumentation
 - vmstat/iostat over multiple runs
 - 'sar'
 - Best for detail of individual task implementation
- Hypervisor Instrumentation
 - Surrounds virtual systems
 - Gives overall summary of usage for system

Application Structure

- Application Classes
 - I/O Intensive
 - CPU/FP Intensive
 - Memory Intensive
 - Network Intensive
 - Composite Distribution
- To some degree, all applications are composite, but a consistent judgement call generally gives better results

Overlap Analysis

- Transaction Borders
- Preemption
- Timeslice
- Multitasking/Multiprogramming Level
- Threading

Error Estimation Terms

- While we can't cover this in detail here, the Merrill book in the reference section does a very detailed job of providing error estimation processes.
- Major areas:
 - Measurement overhead
 - Workload estimation errors over time
 - Imprecise transaction definitions

A More Comprehensive Example

- Global Insurance Company
 - Windows File and Print Application
 - Replacing 180 800 Mhz 1-way NT4 servers
 - Driving 850 network printers
 - Consolidate to IBM mainframe and VM

Analysis Framework

- File/print primarily I/O intensive, little CPU involved
- Much of existing capacity dedicated to redundancy (~40% for failure & clustering requirements)
- Strong transactional model (file update/print job)

Overlap Analysis

- CPU utilization rarely over 10% on any individual server
- Overlay of all server processes under 10% nominal, peak 62% during cluster failover; normal idle percentage of 60-70% of theoretical capacity
- Network and disk I/O rarely idle

Error Estimation Terms

- CPU measurement introduces +/- 2.5-3% error
- Disk usage sampling imposes +/- 5-7% error
- Inefficiency of MS performance tools: priceless...

Sizing Possibilities

■ Simplistic

■ 1800 MIPS

- 12 (11.25) way mainframe
- Next generation only growth path
- Some operational savings, but not compelling

■ Sized

■ 900 MIPS

- 6 way mainframe
 - 1+ million saved
 - Room to grow with single asset
 - Capacity on demand
-

References

- Linux for S/390: ISP/ASP Solutions
RedBook, IBM
- Generalized Capacity Planning Models for
Large Systems, David Merrill

Contact Info

David Boyes

Sine Nomine Associates

+1 703 723 6673

dboyes@sinenomine.net

dboyes@de.sinenomine.net
