Internet-Scale Service Efficiency

James Hamilton

2008-09-16

JamesRH@microsoft.com

web: http://research.microsoft.com/~JamesRH

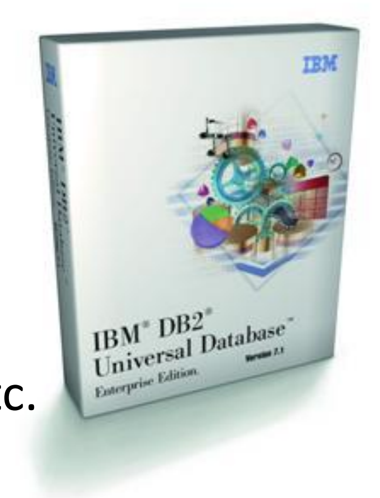
blog: http://perspectives.mvdirona.com

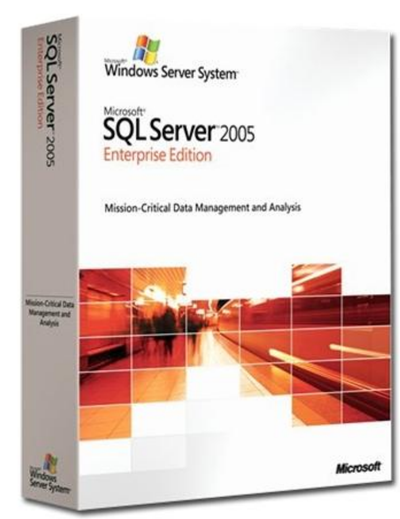
Agenda

- Internet-Scale Service Environment
  - Industry & technology trends
  - Some opportunities while others to be worked around
- Techniques & Distributed Systems Challenges
  - Approaches to scaling to, and beyond, 10^5 servers
  - Trail of interesting distributed systems problems

Background & Biases

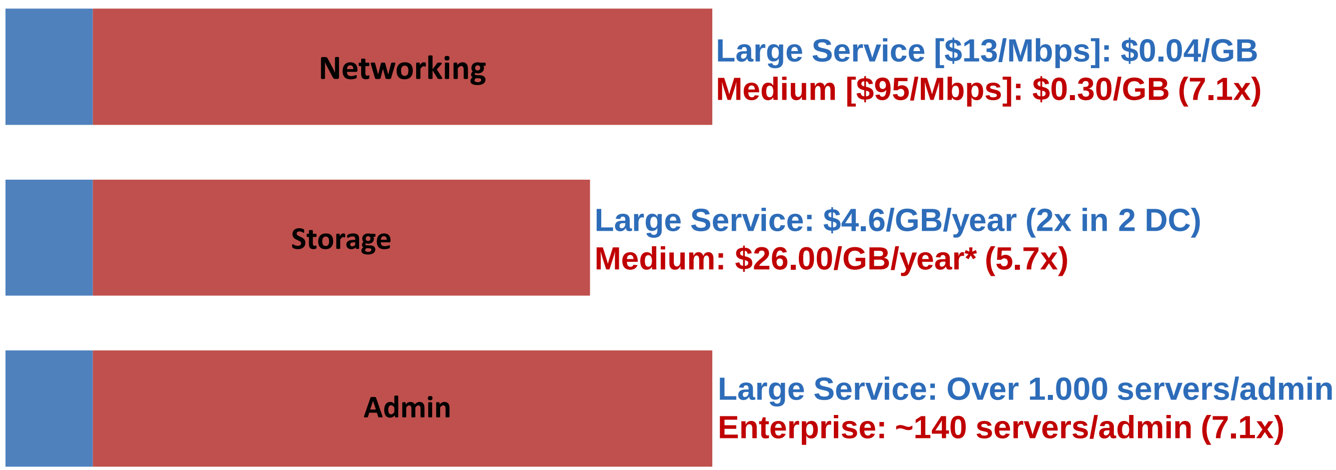
- 15 years in database engine development
  - Lead architect on IBM DB2
  - Architect on SQL Server
    - Have led: Optimizer, SQL compiler, XML, client APIs, fulltext search, execution engine, client protocols, etc.
- Led Exchange Hosted Services Team
  - Mid-sized: ~700 servers in 10 DCs world-wide
- Architect on the Windows Live Platform
  - Live Mesh, Messenger Server, Spaces backed, Live Storage, Identity Services, Groups, etc.
- Currently Data Center Futures architect





Services Economies of Scale

- Substantial economies of scale possible
- Compare a very large service with a small/mid-sized: (~1000 servers):



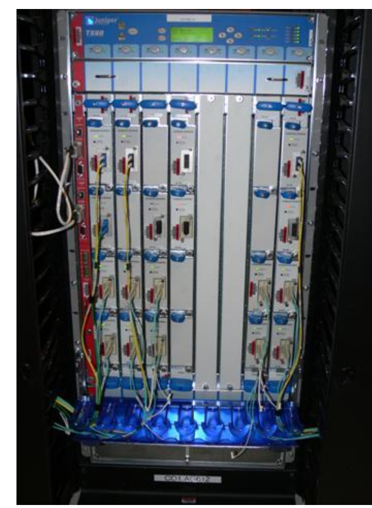
- High cost of entry
  - Physical plant expensive: 15MW roughly $200M
- Summary: significant economies of scale but at very high cost of entry
  - Small number of large players likely outcome

Automation over Scale-up Consolidation

- **Enterprise Approach:**
  - Largest cost is people -- scales roughly with servers (~100:1 common)
  - Enterprise interests center around consolidation & utilization
    - Consolidate workload onto fewer, larger systems
    - Large SANs for storage & large routers for networking
- **Internet-Scale Services Approach:**
  - Largest costs is server H/W
    - Typically followed by cooling, power distribution, power
    - Networking varies from very low to dominant depending upon service
    - People costs under 10% & often under 5% (>1000+:1 server:admin)
  - Services interests centered around work-done-per-$ (or watt)
    - Scale out over up, commodity components, exploit scale economics
- **Services continue to drive distributed systems innovation**
  - Services model starting to show up in some enterprise app areas

Limits to Computation

- Processor cycles are cheap & getting cheaper
- What limits the application of infinite cores?
  1. Power: cost rising & will dominate
  2. Communications: getting data to processor
- The most sub-Moore attributes typically require the most innovation
  - Infinite processors require infinite power
  - Getting data to processors in time to use next cycle:
    - Caches, multi-threading, ILP,…
    - All techniques consume power
- Power & communications key constraints
  - Impacts DC design, server design, & S/W architecture



Latency Lags Bandwidth

	CPU	DRAM	LAN	Disk
Annual bandwidth improvement (all milestones)	1.5	1.27	1.39	1.28
Annual latency Improvement (all milestones)	1.17	1.07	1.12	1.11

- CPU out-pacing all means to feed it
- Bandwidth out-pacing latency across all dimensions
- Additional bandwidth can be achieved via data-path parallelism
  - No joy on latency & again power limits parallelism
- Hubble's Expanding Universe:
  - Everything is getting further away from everything else [Pat Helland]
- Expect many simple, low-frequency processors with low-power sleep
  - Ironically: Applies both to data center & edge devices

Table from Dave Patterson: Why Latency Lags Bandwidth and What It Means to Computing

Power & Related Costs Will Dominate

- **Assumptions:**
  - Facility: ~$200M for 15MW facility (15-year amort.)
  - Servers: ~$2k/each, roughly 50,000 (3-year amort.)
  - Commercial Power: ~$0.07/kWhr (sometimes less)
  - On-site Sec & Admin: 15 people @ ~$100k/annual
- **Run the numbers:**
  - $2.9M/month on server amortization (w/o networking)
    - =PMT(5%/12,12*3,50000*2000,0,1) => ($2,984,653.65)
  - $1.7M/month on data center amortization, onsite security & admin
    - =PMT(5%/12,12*15,200000000,0,1)-(100000/12*15) => ($1,700,024.65)
  - $1.3M/month on power
    - =15,000,000/1000*1.7*0.07*24*31 => (1,328,040)
    - $0.9M/month @ $0.05/kWhr
    - $1.9M/month @ $0.10/kWhr
- **Observation**:
  - $3M/month from charges functionally related to power
  - Power related costs trending flat or up while server costs trending down

Where Does the Power Go?

- Assuming an average data center with PUE ~1.7
  - Power Usage Effectiveness: Total-facilities-power/critical-load-power
  - Each watt to server loses ~0.7W to power distribution & cooling
- Power losses are easier to track than cooling:
  - Transformer losses: 3 transformers at 99.7% efficiency (high)
  - UPS losses: at 94% efficiency (better available)
  - Power transmission & switching losses: 99% efficiency
  - 0.997^3*0.94*0.99 => 0.9
  - Cooling losses remainder 100-(59+9) => 32%
- Data center power consumption:
  - IT load (servers): 1/1.7=> 59%
  - Distribution Losses: 9%
  - Mechanical load(cooling): 32%

Agenda

- Internet-Scale Service Environment
  - Industry & technology trends
  - Some opportunities while others to be worked around
- Techniques & Distributed Systems Challenges
  - Approaches to scaling, to and beyond, 10^5 servers
  - Trail of interesting distributed systems problems

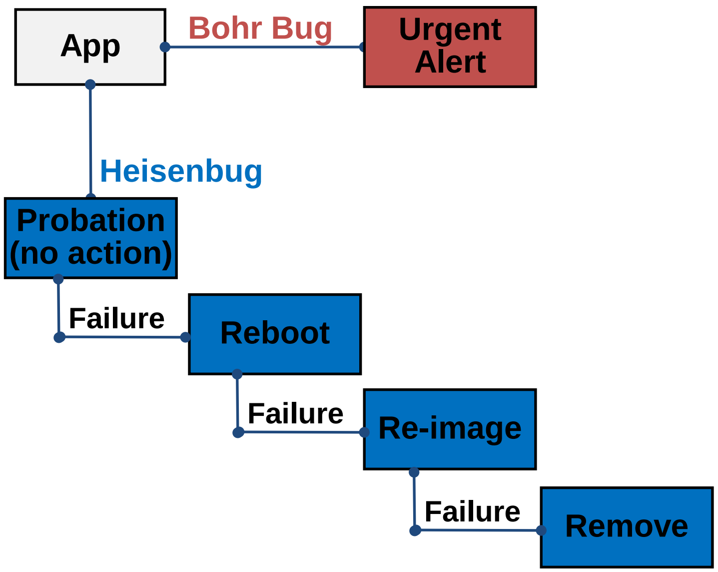
Partitioned & Redundant

- Scalability & availability only achieved through partitioning & redundancy
  - Internet-scale through partitioning
  - Over 4 nines only through redundancy
    - Best hardware never good enough
    - Highly reliable S/W evolves VERY slowly

- Lower quality hardware in large numbers more reliable in aggregate than high-quality hardware
- Repeating a trend seen before in disk
  - Expect the same trend to again play out in networking
- Reliable service built on unreliable S/W & H/W

ROC Service Design Pattern

- Recover-Oriented Computing (ROC)
  - Assume software & hardware will fail frequently & unpredictably
  - Only affordable admin model at high scale
- Heavily instrument applications to detect failures



- **Take machine out of rotation and power down**
- **Set LCD/LED to "needs service"**

Bohr bug: Repeatable functional software issue (functional bugs); should be rare in production

Heisenbug: Software issue that only occurs in unusual cross-request timing issues or the pattern of long sequences of independent operations; some found only in production

Relaxed Consistency Models

- Full ACID semantics unaffordable in real distributed systems
  - Consistency, availability, or partition-tolerance
    - Pick any two*
  - Financial transactions often used as examples of needing ACID yet two-phased commit seldom used
- Relax consistency model exploiting knowledge of application semantics
  - Caches & temporal inconsistency
- Hairball problem in social networks
  - Redundant application maintained partitioned views
  - Caching (e.g. memcached)

**** CAP Conjecture, Eric Brewer***

Some Data "Pulled" to Core And Some to Edge

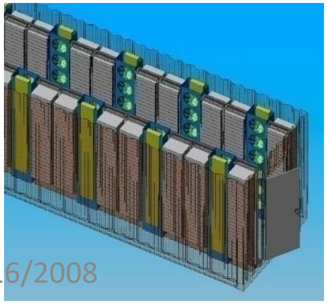
- User data pulled to the edge (close to user)
  - Highly interactive web applications
  - Social & political restrictions on data movement
    - e.g. Patriot Act concerns & jurisdictional restrictions
  - Application & data availability
  - Techniques:
    - Content Distribution Networks
    - Geo-partitioned and/or geo-redundant applications
- Aggregated data pulled to network core
  - Data mining & analysis workloads run central
    - e.g. MapReduce workloads

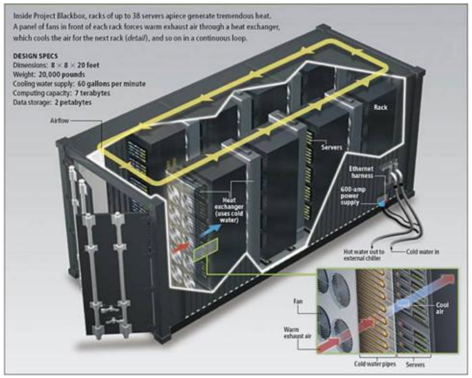
High-Scale Data Analysis

- Yield management first used by airlines
  - Airplane more expensive than computation
- Falling cost of computing allows yield-management of more resources
- Heavily used in retail:
  - Shelf-space optimization, supply-chain mgmt, …
- Financial community has widely implemented automated trading & data analysis compute farms of 1,000s of nodes
- Analysis systems dominate transactional systems
  - Transactional workload growth tend to be related to business growth
  - Analysis workload growth bounded only by decline cost of computing

Modular Data Center

- Just add power, chilled water, & networking
- Drivers of move to modular
  - Faster pace of infrastructure innovation
    - Power & mechanical innovation to 3 year cycles
  - Efficient scale-down
    - Driven by latency & jurisdictional restrictions
  - Service-free, fail-in-place model
    - 20-50% of system outages cause by admin error
    - Recycle as a unit
  - Incremental data center growth
    - Transfer fixed to variable cost





Systems & Power Density

- Estimating DC power density difficult through 15+ year horizon
  - Power + Cooling: ~70% of capex
  - Shell ~12% of DC capex
  - Better waste floor than power
  - Add modules until power is absorbed
    - Modular DC eliminates impossible to predict future power density requirements
- 480VAC to container (reduce dist losses)
- Over 20% of DC costs is in power redundancy
  - N+2 generation at over $2M each
- Instead, use more, smaller, cheaper DC

Memory to Disk Chasm

- Disk I/O rates grow slowly while CPU data consumption grows near Moore
  - Random read 1TB disk: 15 to 150 days*
- Sequentialize workloads
  - Essentially the storage version of cache conscious algorithms
    - e.g. map/reduce
  - Disks arrays can produce acceptable aggregate sequential bandwidth
- Redundant data: materialized views & indexes
  - Asynchronous maintenance
  - Delta or stacked indexes (from IR world)
- Distributed memory cache (remote memory "closer" than disk)
- I/O Cooling: Blend hot & cold data
- I/O concentration: partition hot & cold data

**** Tape is Dead, Disk is Tape, Flash is Disk, Ram Locality is King (Jim Gray)***

New layer in storage hierarchy

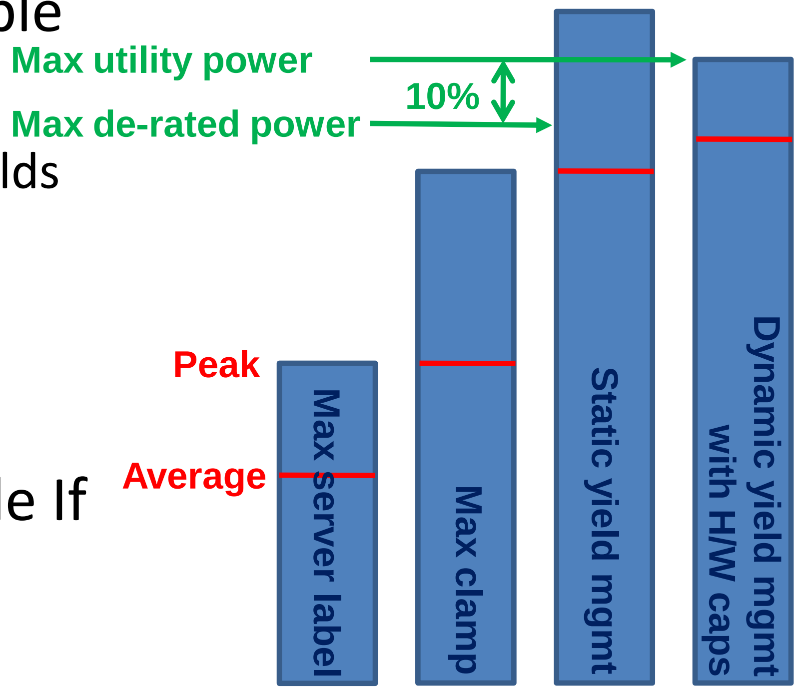
- NAND Flash as new layer in memory/storage hierarchy
- Last DIMM added to server memory costs same as first but less performance gain
  - Move some data "down" from memory to flash cache
- Disks added to get IOPS often strands capacity
  - Enterprise disk ~170 to 200 random IOPS
  - Commodity disk: ~70 to 100 random IOPS
  - Move some data "up" from disk to flash storage
- On client-side, NAND flash entirely replaces disk
  - Low power, quiet, lightweight, robust, high random IOPS,…

Graceful Degradation & Admission control

- No economic amount of "head room" is sufficient
  - Even at 25-50% hardware utilization, spikes will exceed 100%
  - EHS average-to-peak load spread over 6x
- Prevent overload through admission control
  - Service login typically more expensive than steady state
- Graceful Degradation Mode prior to admission control
  - Find less resource-intensive modes to provide degraded services

Power Yield Management

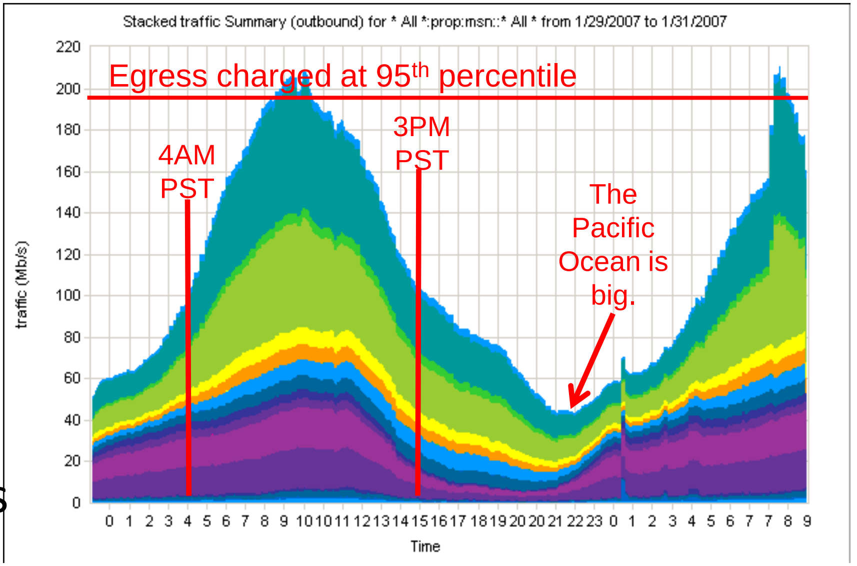
- "Oversell" power, the most valuable resource:
  - e.g. sell more seats than airplane holds
- Overdraw penalty high:
  - Pop breaker (outage)
  - Overdraw utility (fine)
- Considerable optimization possible If workload variation is understood
  - Workload diversity & history helpful
  - Graceful Degradation Mode to shed workload



Power Provisioning in a Warehouse-Sized Computer, Xiabo Fan, Wolf Weber, Luize Borroso

Resource Consumption Shaping

- Essentially yield mgmt applied to full DC
- Network charged at 95th percentile:
  - Push peaks to troughs
  - Fill troughs for "free"
    - e.g. Amazon S3 replication
  - Dynamic resource allocation
    - Virtual machine helpful but not needed
  - Charged for symmetrically so ingress effectively free



David Treadwell & James Hamilton / Treadwell Graph

- Power also charged at 95th percentile
  - Server idle to full-load range: ~65% (e.g. 158W to 230W)
  - S3 (suspend) or S5 (off) when server not needed
- Disks come with both IOPS capability & capacity
  - Mix hot & cold data to "soak up" both
- Encourage priority (urgency) differentiation in charge-back model

Summary

- Hosted services will be a large component of many enterprise solutions
- Hosted services will dominate consumer S/W
- Innovations needed throughout services stack
  - Data center design, especially modularity, power, & cooling
  - Low power server design
  - S/W infrastructure
  - Multi-device support required & should be exploited
- Integrated approach over entire H/W & S/W stack
  - Optimizations at each layer need cooperation from others

More Information

- **These slides:**
  - http://mvdirona.com/jrh/TalksAndPapers/JamesRH_Ladis2008.pdf
- **Designing & Deploying Internet-Scale Services:**
  - http://mvdirona.com/jrh/talksAndPapers/JamesRH_Lisa.pdf
- **Architecture for Modular Data Centers:**
  - http://mvdirona.com/jrh/talksAndPapers/JamesRH_CIDR.doc
- **Increasing DC Efficiency by 4x:**
  - http://mvdirona.com/jrh/talksAndPapers/JamesRH_PowerSavings20080604.ppt
- **Recovery-Oriented Computing:**
  - http://roc.cs.berkeley.edu/
  - http://www.cs.berkeley.edu/~pattrsn/talks/HPCAkeynote.ppt
  - http://www.sciam.com/article.cfm?articleID=000DAA41-3B4E-1EB7-BDC0809EC588EEDF
- **Autopilot: Automatic Data Center Operation:**
  - http://research.microsoft.com/users/misard/papers/osr2007.pdf
- **Perspectives Blog:**
  - http://perspectives.mvdirona.com
- **Email:**
  - JamesRH@microsoft.com