Understanding IT Cost Components – How to Maximize your IT Investments

Ray Jones, Vice President

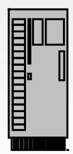
System z Worldwide Software Sales, IBM Software Group

Many Cost Components

80:20 rule helps to achieve reasonable results in a short time

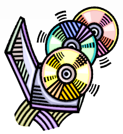
Components

Hardware



List vs Discounted
Fully configured vs. basic, Prod. vs. DR
Refresh / upgrade, Solution Edition…

Software



IBM and ISV, OTC and Annual maint (S&S)
MLC, PVU, RVU, ELA, core, system

People

FTE rate, in house vs. contract

Network

Adapters, switches, routers, hubs
Charges, Allocated or apportioned, understood or clueless

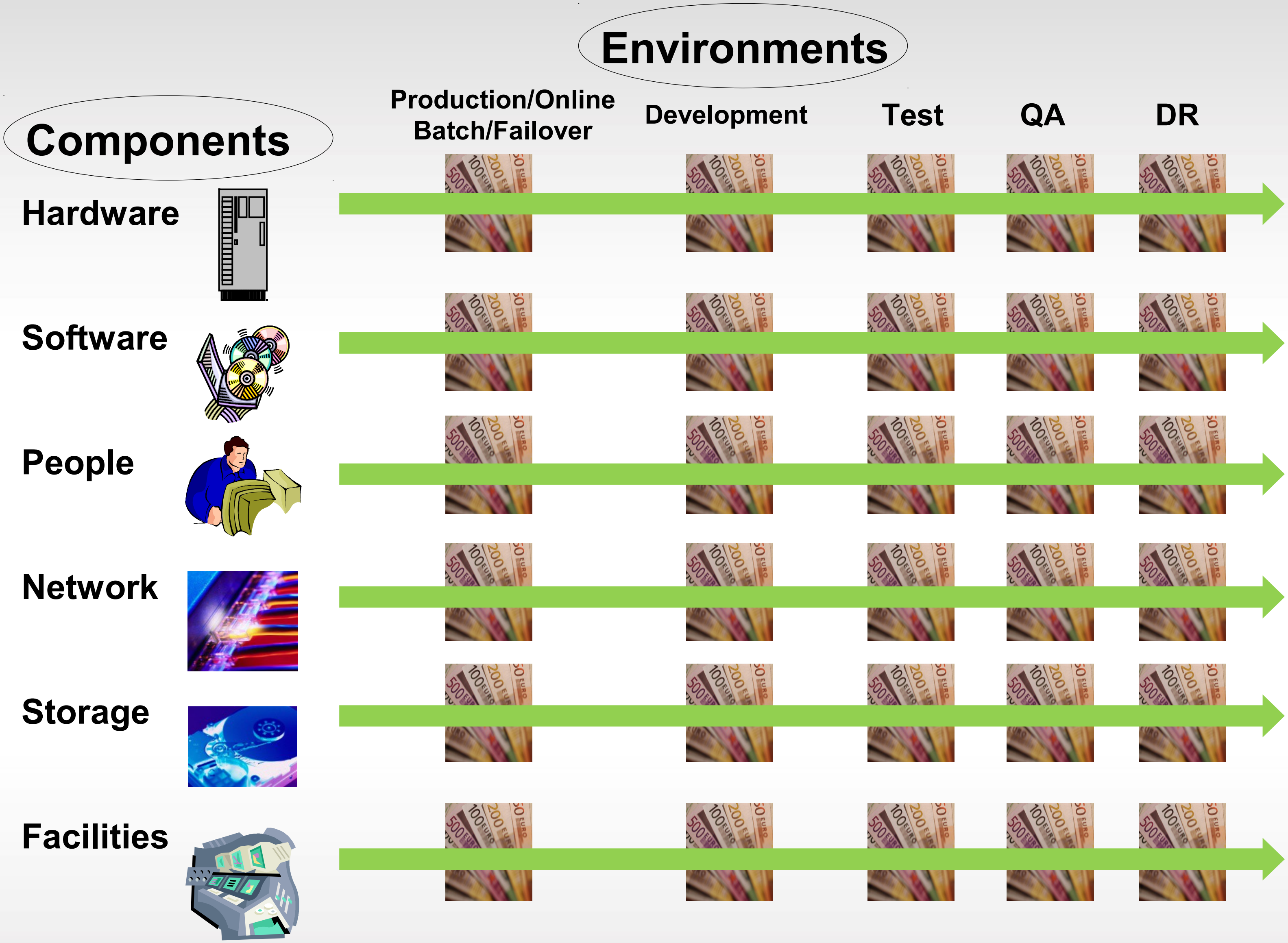
Storage

ECKD, FBA, SAN, Compressed, Primary, secondary
Disk (multiple vendors), tape, Virtual, SSD

Facilities

Space, electricity, air cooling, infrastructure including UPS and generators, alternate site(s), bandwidth

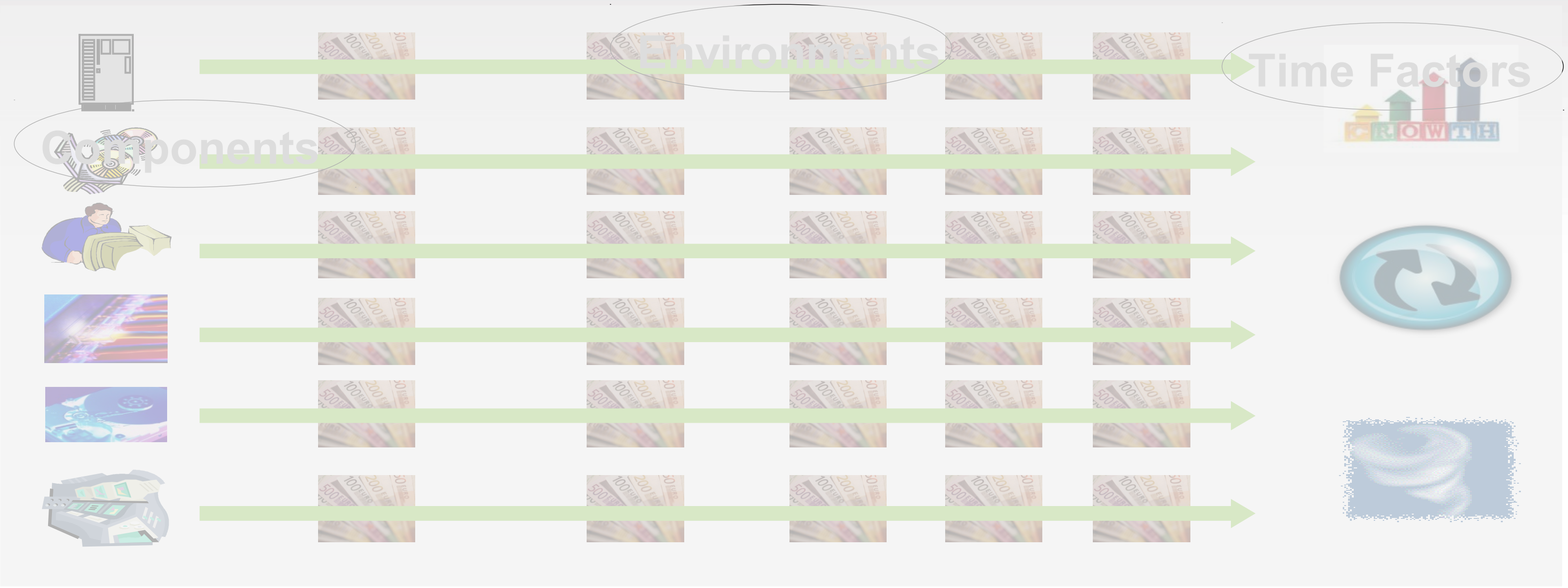
Environments Multiply Components



Time Factors Drive Growth And Cost

- Migration time and effort
- Business organic growth and/or planned business changes affect capacity requirements
  - e.g. Change of access channel or adding a new internet accessible feature can double or triple a components workload
  - Link a business metric (e.g. active customer accounts) to workload (e.g. daily transactions) and then use business inputs to drive the TCO case
- Other periodic changes – hardware refresh or software remediation

Non-Functional Requirements Can Drive Additional Resource Requirements



Availability …Security …Resiliency …Scalability …

Qualities of Service, Non-Functional Requirements

What Happens In a TCO Study?

Workload identified for analysis

frontends

databases

other components

Application XYZ (Prod, Dev, QA)

Deployment Choices

- **Do nothing**
- **Optimize current environment**
- **Deploy on other platforms**

Key steps in analysis

1. Establish equivalent configurations
- Needed to deliver workload

2. Compare Total Cost of Ownership
- TCO looks at different dimensions of cost

Approaches To Establishing Equivalent Configurations

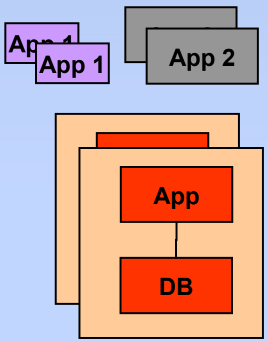
- Bottom up approach
  - Atomic benchmarks
  - Counting cycles, CPI comparisons …
  - IO, memory, cache, co-location effects …
  - Tends to show smaller core expansion factors

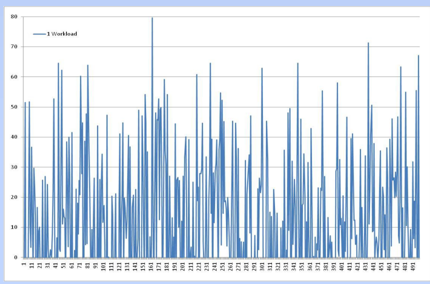
- Top down approach
  - "Real world" observations
  - Tends to show much larger core expansion factors

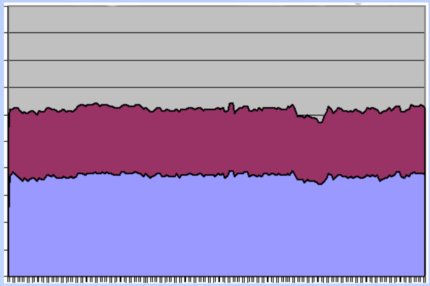
- When atomic benchmarks are assembled to represent "real world", bottom up numbers approach top down numbers

How Can We Determine Equivalent Configurations?

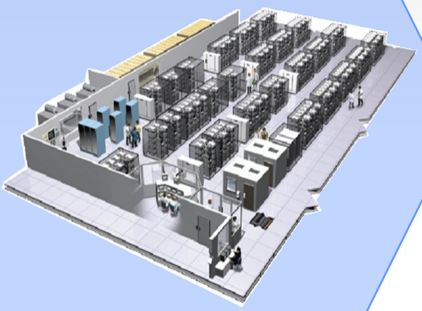
Real world aspects determine accurate equivalence







....



Platform factors

GHz, CPI, IO, co-location etc

Variability in demand

Different size servers

Workload Management

Mix workloads with different priorities

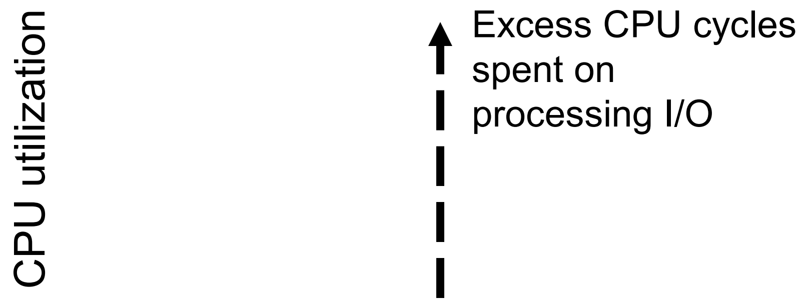
....

Top Down approach

What we see in customer environments

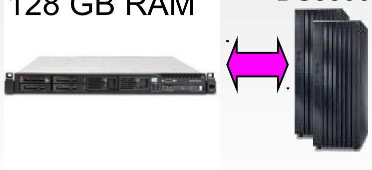
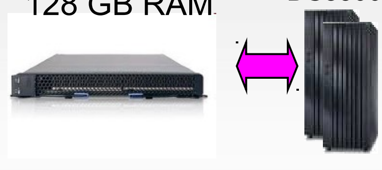
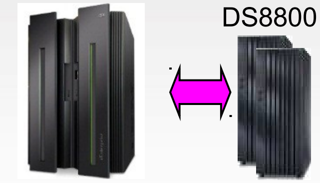
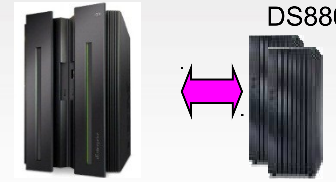
Example Of a Low Level Factor: I/O Load

- Intel's performance degrades as I/O demand increases
  - No dedicated I/O subsystem
- Test case scenario: Run multiple virtual machines on x86 server
  - Each virtual machine has an average I/O rate
  - x86 processor utilization is consumed as I/O rate increases



Benchmarks Show System z And z/OS Are Optimized For Batch Processing

Intel x3550	Power PS701	Linux on z	z/OS
12 processors 128 GB RAM, DS8300	8 processors 128 GB RAM, DS8300	8 processors 128 GB RAM, DS8800	8 processors 128 GB RAM, DS8800

	Intel x3550	Power PS701	Linux on z	z/OS
	Sorting Average CPU 89%	Sorting Average CPU 92%	Sorting Average CPU 90%	Sorting Average CPU 72%
SORT *Job: Sort a 3 GB transaction file – Repetitions: 300*				
Total Time (secs)	7,680	6,900	2,590	644
Concurrency	12	20	18	45
Rate (MB/sec)	240	280	746.2	3,000
MERGE *Job: Merge 30 sorted files into a 90 GB master file – Repetitions: 10*				
Total Time (secs)	11,709	7,920	2,799	558
Concurrency	10	10	10	10
Rate (MB/sec)	157	244	690.5	3,460

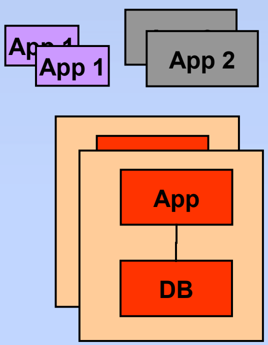
Results:

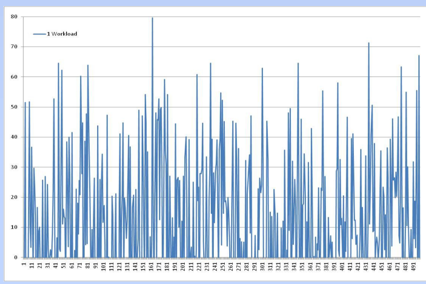
1. Running same software, x86 batch window is 3.6x greater than System z
2. On System z, Linux batch window is 4.5x greater than z/OS
3. Off-loading batch from z/OS to x86 leads to as much as 16x increase in batch window

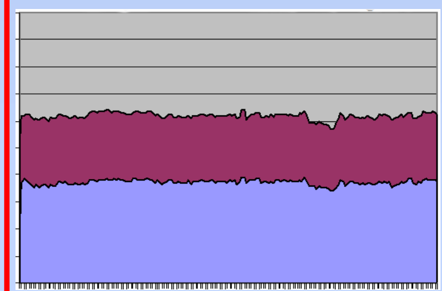
Source: Internal IBM study

How Can We Determine Equivalent Configurations?

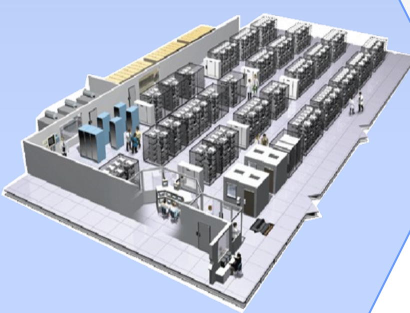
Real world aspects determine accurate equivalence







....



Platform factors

GHz, CPI, IO, co-location etc

Variability in demand

Different size servers

Workload Management

Mix workloads with different priorities

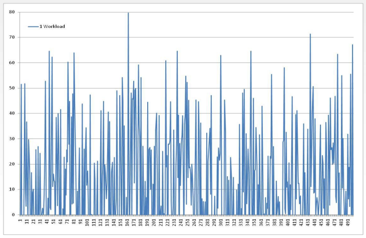
....

Top Down approach

What we see in customer environments

Larger Servers With More Resources Make More Effective Consolidation Platforms

- Most workloads experience variance in demand

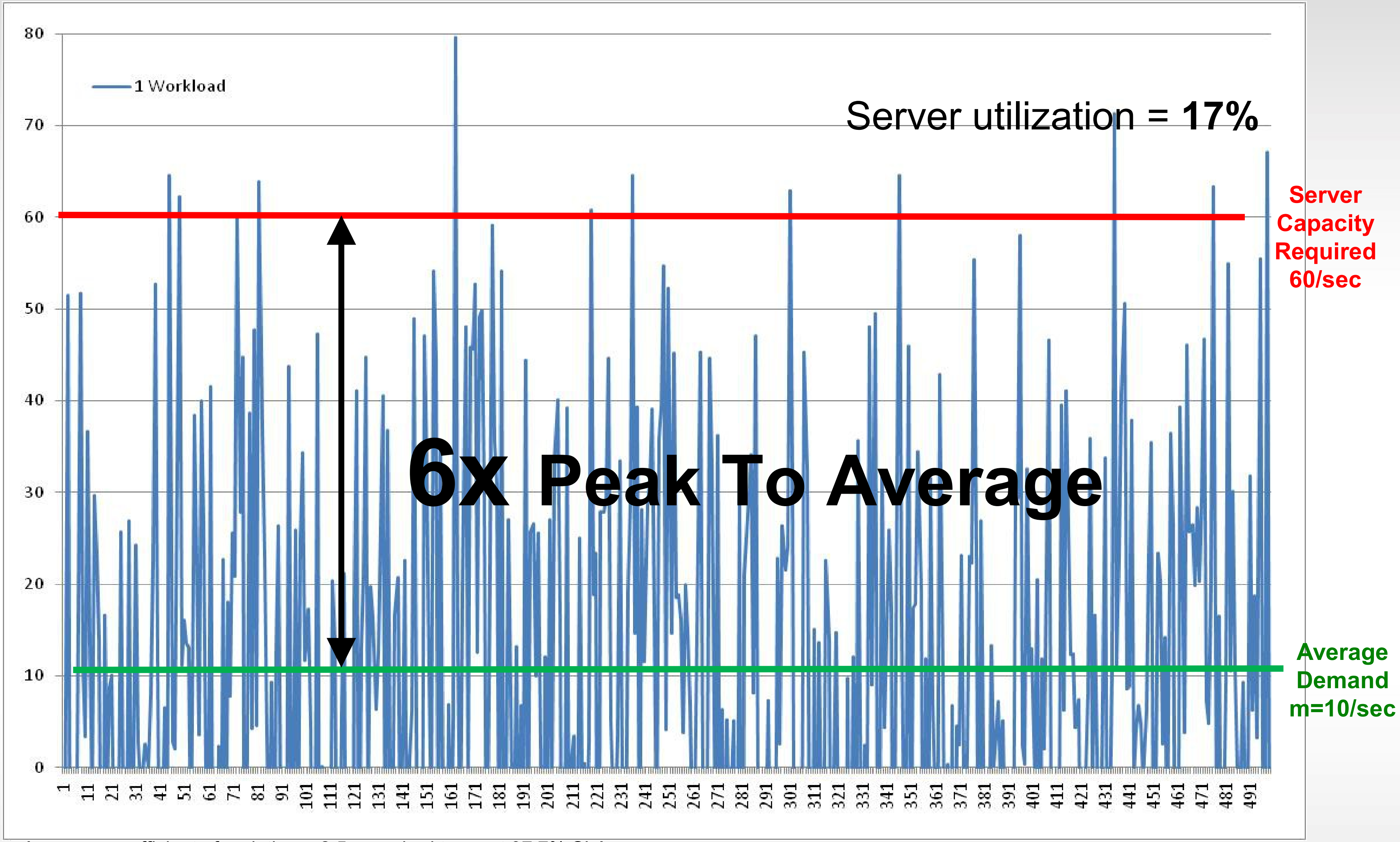


- When you consolidate workloads with variance on a virtualized server, the variance of the sum is less (statistical multiplexing)

- The more workloads you can consolidate, the smaller is the variance of the sum

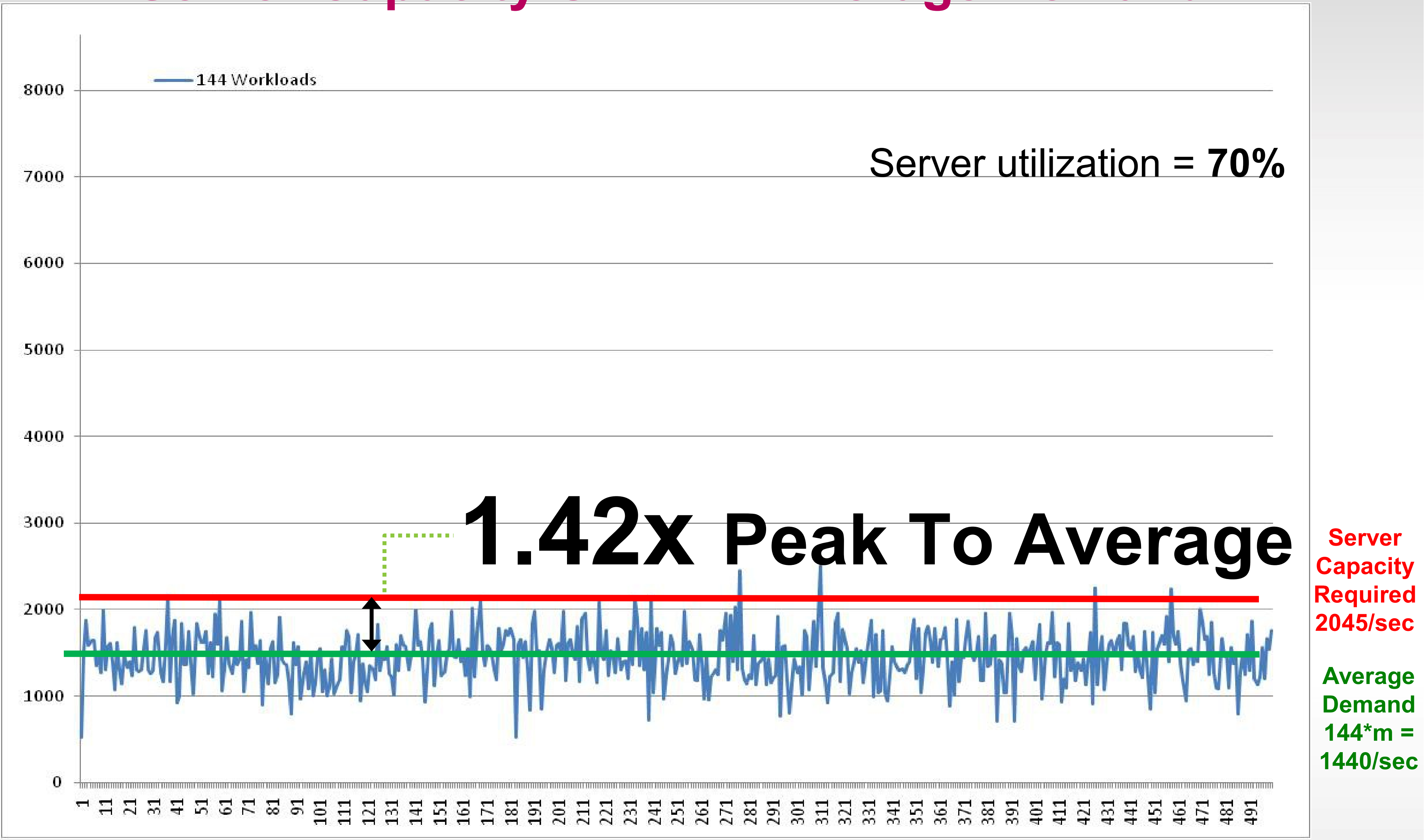
- Consequently, bigger servers with capacity to run more workloads can be driven to higher average utilization levels without violating service level agreements, thereby reducing the cost per workload

A Single Workload Requires a Machine Capacity Of 6x the Average Demand



Assumes coefficient of variation = 2.5, required to meet 97.7% SLA

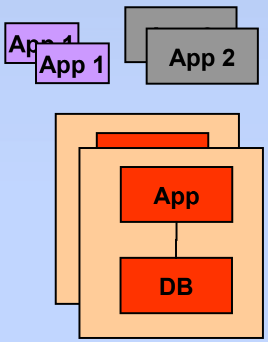
Consolidation Of 144 Workloads Requires Server Capacity Of 1.42x Average Demand

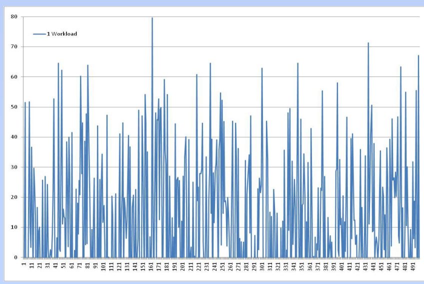


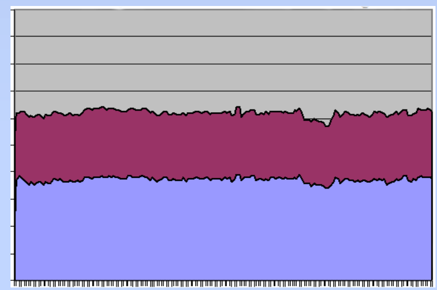
Assumes coefficient of variation = 2.5, required to meet 97.7% SLA

How Can We Determine Equivalent Configs?

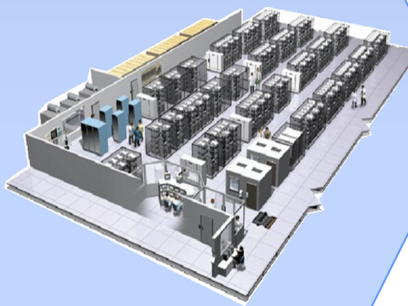
Real world aspects determine accurate equivalence







....



Size of the workload	**Variability in demand**	**Workload Management**	....	**Top Down approach**
Same software on Same size servers	Different size servers	Mix workloads with different priorities		What we see in customer environments

Workload Management

- Hosting platforms must be able to support high priority and low priority workloads together when sharing resources
  - Enables maximum utilization of the hosting platform

- Particularly relevant in a Private cloud environment
  - Multiple tenants with different priorities

- Desired behavior when mixing workloads
  - Low priority workloads “give up” resources to high priority workloads when required, soak up unused resources when available
  - High priority workload performance must not degrade

Priority Transactional Workload With Constant Demand Running Standalone On z/OS

Capacity Used
High Priority Steady State - 85.2% CPU Minutes
Unused (wasted) - 14.8% CPU Minutes

Priority Workload Metrics
Total Throughput: 417.8K
Maximum TPS 129.7

Priority Transactional Workload On System z Does Not Degrade When Low Priority Donor Workload Is Added

NO
Steady state CPU usage leakage
1%
Total Transaction leakage

Capacity Used
High Priority Steady State - 85.3% CPU Minutes
Unused (wasted) - 0% CPU Minutes

Priority Workload Metrics
Total Throughput: 414.7K
Maximum TPS 128.1

Priority Workload With Varying Demand Running Standalone On System z PR/SM

High Priority Workload
Demand Curve

% CPU Usage

Priority Workload

Time (mins.)

Capacity Used
High Priority - 72.2% CPU Minutes
Unused (wasted) - 27.8% CPU Minutes

Priority Workload Metrics
Total Throughput: 9.125M
Avg Response Time: 140ms

Priority Workload On System z Does Not Degrade When Low Priority Donor Workload Is Added

Run High Priority
And Low Priority
Workloads Together

% CPU Usage

Time (mins.)

NO
throughput leakage
NO
response time increase

Capacity Used
High Priority - 74.2% CPU Minutes
Low Priority - 23.9% CPU Minutes
Wasted – 1.9% CPU Minutes

Priority Workload Metrics
Total Throughput: 9.125M
Avg Response Time: 140ms

Priority Workload With Varying Demand Running Standalone On x86 Hypervisor

High Priority Guest
CPU Demand

% CPU Usage

Priority Workload

Time (mins.)

Capacity Used
High Priority - 57.5% CPU Minutes
Unused (wasted) – 42.5% CPU Minutes

Priority Workload Metrics
Total Throughput: 6.47M
Avg Response Time: 153ms

Priority Workload On x86 Hypervisor Degrades Severely When Low Priority Workload Is Added

Run High Priority
And Low Priority
Workloads Together

% CPU Usage

30.7%
throughput leakage
45.1%
response time increase
21.9%
wasted CPU minutes

Time (mins.)

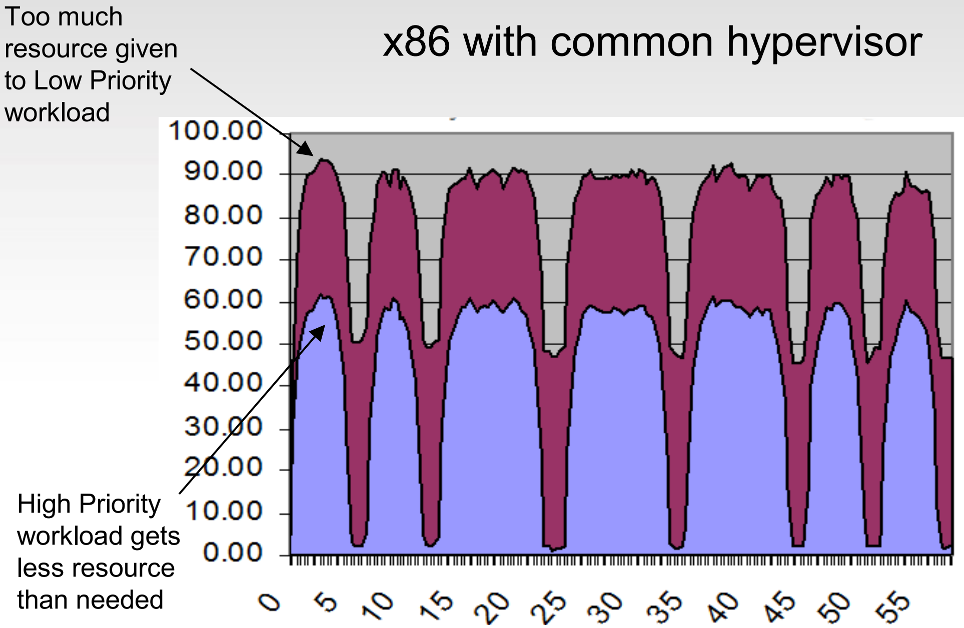
Capacity Used
High Priority - 42.3% CPU Minutes
Low Priority – 35.8% CPU Minutes
Wasted – 21.9% CPU Minutes

Priority Workload Metrics
Total Throughput: 4.48M
Avg Response Time: 220ms

System z Virtualization Enables Mixing Of High And Low Priority Workloads Without Penalty

System z

x86 with common hypervisor



- Perfect workload management
- Consolidate workloads of different priorities on the same platform
- Full use of available processing resource (high utilization)

- Imperfect workload management
- Forces workloads to be segregated on different servers
- More servers are required (low utilization)

Deliver High And Low Priority Workloads Together While Maintaining SLA

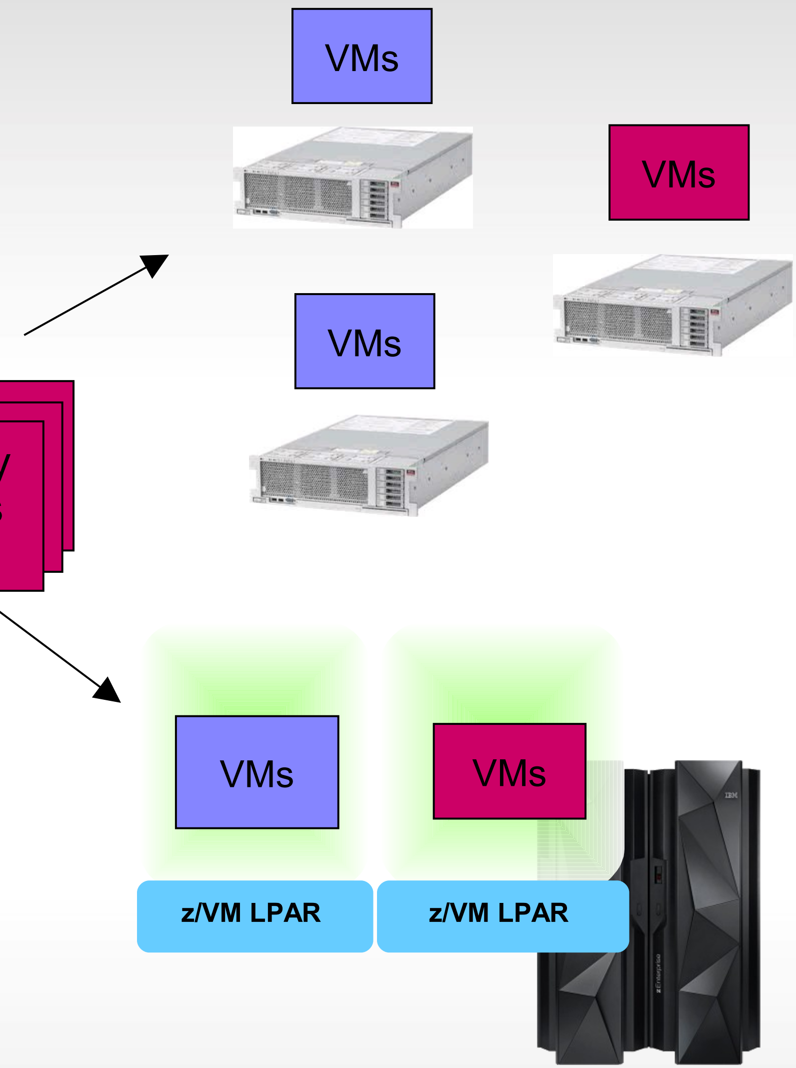
Comparison to determine which platform provides the lowest TCA over 3 years

High priority workloads

Low priority workloads

- **IBM WebSphere 8.5 ND**
- **IBM DB2 10 AESE**
- **Monitoring software**

High priority online banking workloads driving a total of **11.89M** transactions per hour and low priority discretionary workloads



Virtualized on 3 Intel 40 core servers

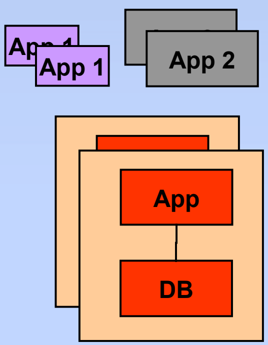
3.75x *more cores*

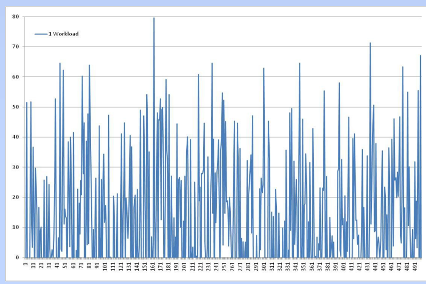
z/VM on zEC12
32 IFLs

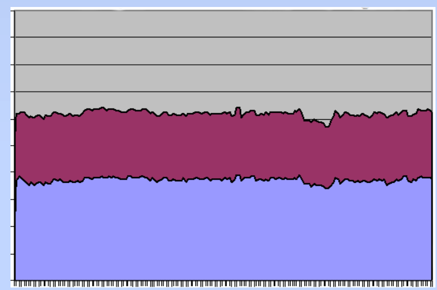
Consolidation ratios derived from IBM internal studies.. zEC12 numbers derived from measurements on z196. Results may vary based on customer workload profiles/characteristics. Prices will vary by country.

How Can We Determine Equivalent Configs?

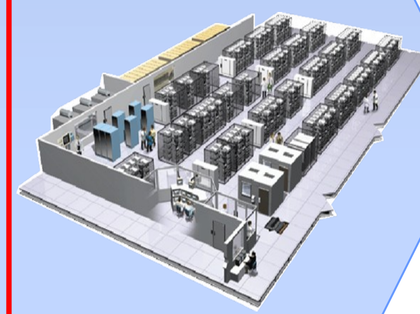
Real world aspects determine accurate equivalence







....



Size of the workload	**Variability in demand**	**Workload Management**	**....**	**Top Down approach**
Same software on Same size servers	Different size servers	Mix workloads with different priorities		What we see in customer environments

Core Proliferation For A Very Large Workload

Configurations for equivalent throughput (10,716 Transactions Per Second)

16x 32-way HP Superdome
App. Production / Dev / Test

8x 48-way HP Superdome
DB Production / Dev /Test

zEC12 41-way Production / Dev / Test

41 GP processors
(38,270 MIPS)

896 processors
(3,668,600 Perf Units)

***22x** more cores!*

Core Proliferation For A Mid-sized Workload

6x 8-way HP DL Production / Dev
2x 64-way p595 Production / Dev
Application/MQ/DB2/Dev partitions

2x z900 3-way Production / Dev / QA / Test

6 processors
(1,660 MIPS)

176 processors
(800,072 Performance units)

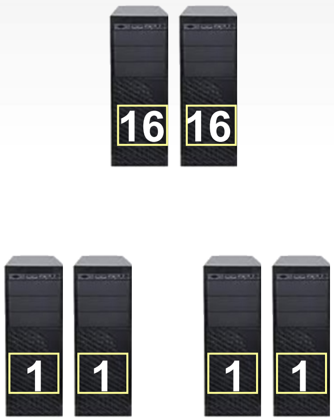
***29x** more cores!*

482 Performance Units per MIPS

Core Proliferation For A Small Offload Project

2x 16-way Production / Dev / Test / Education
App, DB, Security, Print and Monitoring
4x 1-way Admin / Provisioning / Batch Scheduling

z890 2-way Production / Dev / Test / Education
App, DB, Security, Print, Admin & Monitoring



0.88 processors

(332 MIPS)

36 Unix processors

(222,292 Performance Units)

41x more cores

Almost 5 Year Migration

670 Performance Units per MIPS

1 CICS region in production!!
CICS/IDMS migrated to CICS/DB2.
Accessing DB2 thru mapping layer

No Disaster Recovery

Core Proliferation For A Smaller Offload Project

z890 Production / Test

4x p550 (1ch/2co)
Application and DB

0.24 processors
(88 MIPS)

8 Unix processors
(43,884 Performance Units)

33x more cores

3 Year Migration

499 Performance Units per MIPS

Just Completed x86 Offload

3x HP DL580 (2ch/20co)
Production / Dev / Test
(2011 x86 technology)

z800 Production /
Dev / Test
(2002 mainframe technology)

2.1 processors

(499 MIPS)

60 Linux processors

(383,022 Perf Units)

29x more cores

(despite the 9 year technology gap!)

1.5 Year Migration

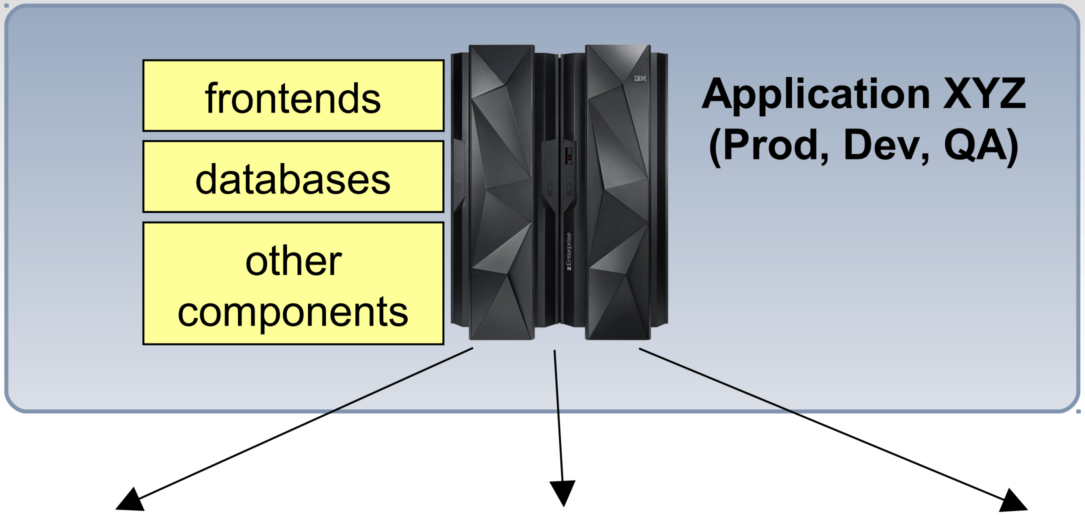
768 Performance Units per MIPS

So What Were The Total Costs In The Core Proliferation Cases We Saw Earlier?

Case	RPE/MIPS	Z Total Cost	Distributed Total Cost	Factor
Large benchmark	95	**$111M** (5 yr. TCA)	**$180M** (5 yr. TCA)	1.62x
Mid size offload	482	**$17.9M** (5 yr. TCO)	**$25.4M** (5 yr. TCO)	1.42x
Small offload	670	**$4.9M** (4 yr. TCO)	**$17.9M** (4 yr. TCO)	3.65x
Even smaller offload	499	**$4.7M** (5 yr. TCO)	**$8.1M** (5 yr. TCO)	1.72x

What Happens In a TCO Study?

Workload identified for analysis



Deployment Choices

- Do nothing
- Optimize current environment
- Deploy on other platforms

Key steps in analysis

1. **Establish equivalent configurations**
   - Needed to deliver workload
2. **Compare Total Cost of Ownership**
   - TCO looks at different dimensions of cost

Lessons Learned Can Be Grouped Into Three Broad Categories

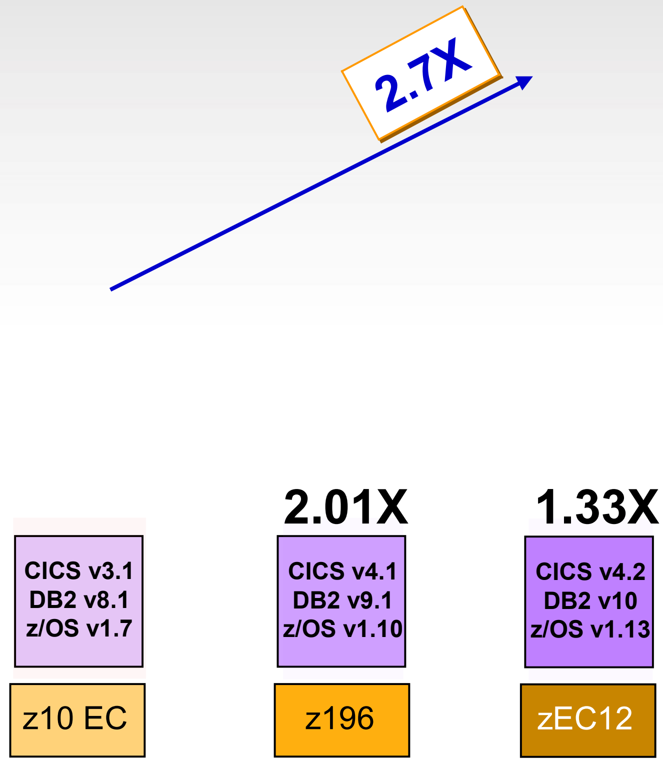
- Always compare to an optimum System z environment
- Look for not-so-obvious distributed platform costs to avoid
- Consider additional platform differences that affect cost

Currency Reduces Cost – Hardware

2 generations,
from z9 to z196

- Typical customer (European bank) hardware refresh scenario
  - 2M investment pays back >1M savings every year – most cases positive in a 3 year period
  - Savings from VWLC->AWLC and specialty processor upgrades
- Comparing latest technology servers to old mainframes is unfair but often done

Performance Improvements Can Lower MLC Costs And Free Up Hardware Capacity



Customer examples:

(1) Large MEA bank

- Delayed upgrade from z/OS 1.6 because of cost concerns
- When finally did upgrade to z/OS 1.8
  - Reduced each LPAR's MIPS by 5%
  - Monthly software cost savings paid for the upgrade almost immediately

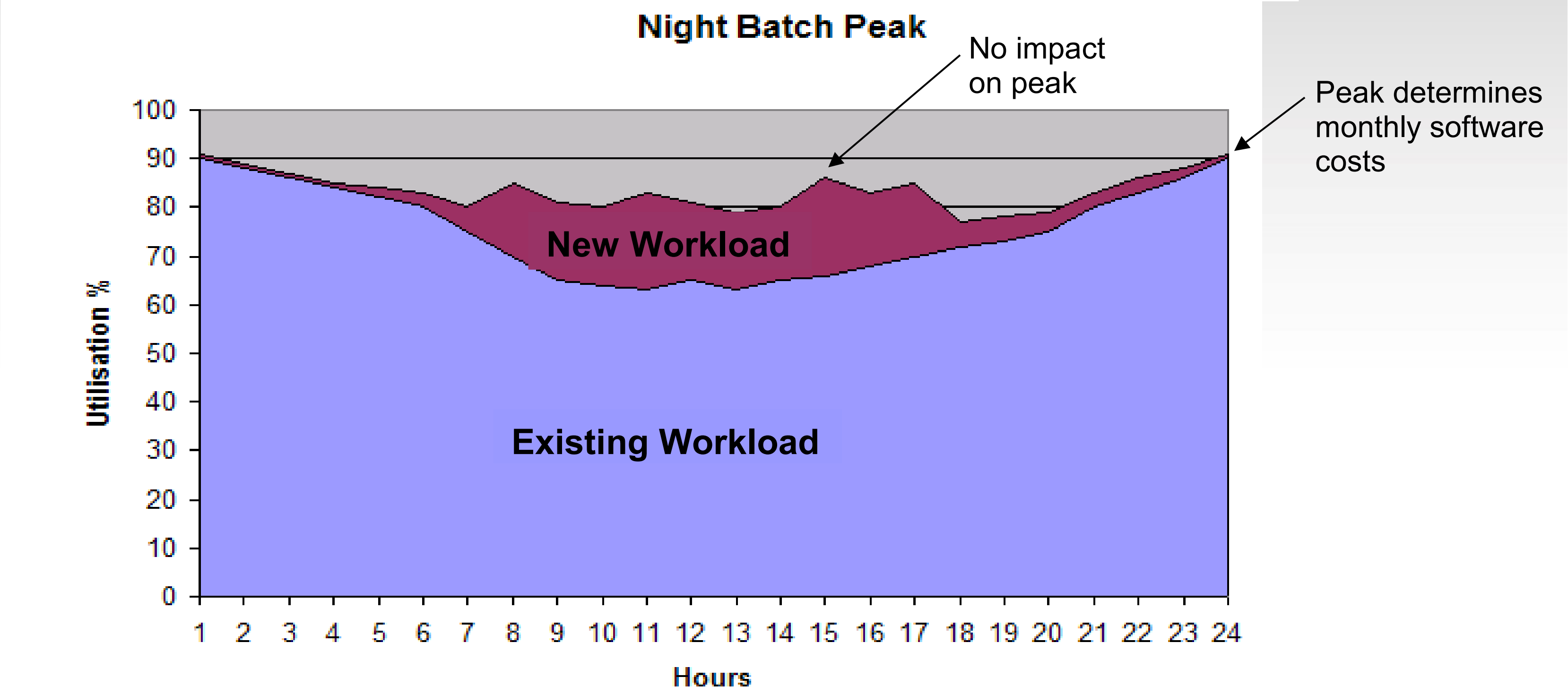
(2) Large European Auto company

- Upgraded to DB2 10
- Realized 38% pathlength reduction for their heavy insert workload
  - Other DB2 10 users saw 5-10% CPU reduction for traditional workloads

Additionally, save costs by moving to newer compilers and tuning

IBM internal core banking workload (Friendly Bank). Results may vary.

Sub-Capacity May Produce Free Workloads



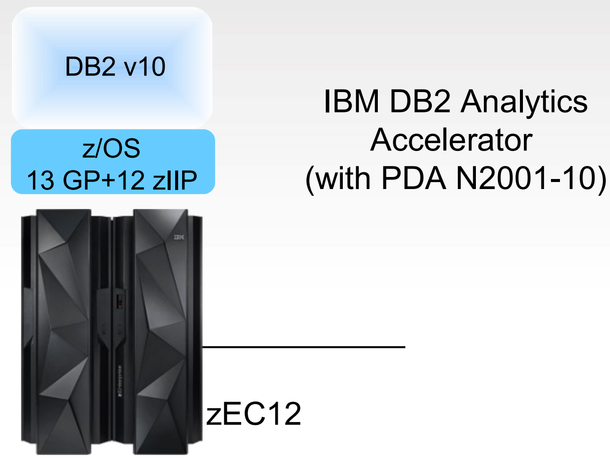
- Standard “overnight batch peak” profile – drives monthly software costs
- Hardware and software are free for new workloads using the same middleware (e.g. DB2, CICS, IMS, WAS, etc.)
- Ensure you exploit any free workload opportunities, and conversely, avoid offloading free applications!

Leverage Accelerators Where Relevant

Standalone Pre-integrated Competitor V3

Quarter Unit

IBM zEnterprise Analytics System 9700



Unit Cost
$51/Reports per Hour

Workload Time	141 mins
Reports per Hour	68,581
Total Cost (3 yr. TCA) (HW+SW+Storage)	$3,530,041

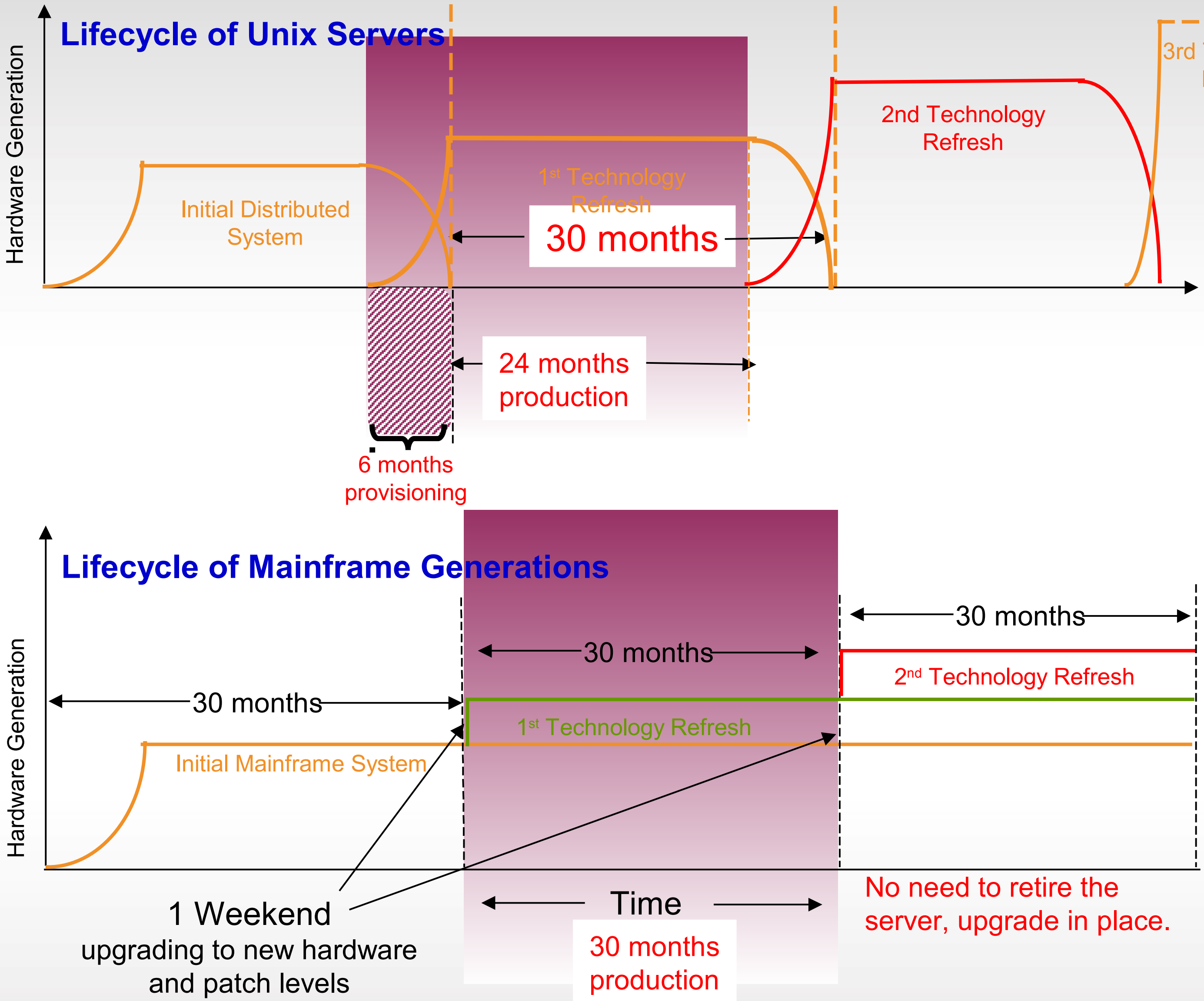
Unit Cost
$17/Reports per Hour

Workload Time	25 mins
Reports per Hour	386,798
Total Cost (3 yr. TCA) (13 GP + 12 zIIP, HW+SW+ Storage + Accelerator V3.1 with PDA N2001-10 hardware)	$6,464,849

***3x** price performance!*

Source: Customer Study on 1TB BIDAY data running 161,166 concurrent reports. Intermediate and complex reports automatically redirected to IBM DB2 Analytics Accelerator for z/OS. Results may vary based on customer workload profiles/characteristics. Note: Indicative 9700 pricing only internal to IBM, quotes to customer require a formal pricing request with configurations.

Distributed Servers Need To Be Replaced Every 3 To 5 Years



Refresh is normally even worse than just re-purchasing existing capacity as this real customer demonstrates:

Non-mainframe systems must co-exist for months at a time while being refreshed, requiring space, power, licenses etc. In this case only 24 months of productive work is realized for each 30 month lease period and the leases overlap up to 6 months

The mainframe by contrast is upgraded over a weekend and is fully productive at all times

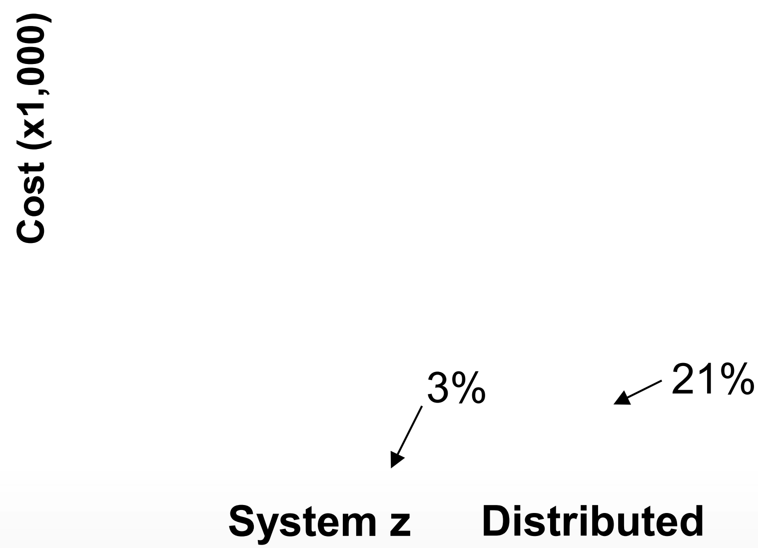
Disaster Recovery On System z Costs Much Less Than On Distributed Servers

A large European insurance company with mixed distributed and System z environment at :

Disaster Recovery Cost as a percentage of Total Direct Costs:

System z – **3%**

Distributed – **21%**



Two mission-critical workloads on distributed servers had DR cost > 40% of total costs

Disaster Recovery Testing Is Typically More Expensive On Distributed Platforms Too

- A major US hotel chain
  - ~ 200 Distributed Servers (LinTel, Wintel, AIX, and HP-UX)

	Person-hours	*Elapsed days*	*Labor Cost*
Infrastructure Test (7 times)	1,144	7	$89,539
Full Test (4 times)	2,880	13	$225,416
Annual Total – Distributed	14,952*	73	$1,170,281
Mainframe Estimate	2,051*	10	$160,000

* Does not include DR planning and post-test debriefing

- Customer Recovery Time Objective (RTO) estimates:
  - Distributed ~ 48 hours to 60 hours
  - Mainframe ~ 2 hours

- Conclusion: Mainframe both simplifies and improves DR testing

Large Systems With Centralized Management Deliver Better Labor Productivity

Large US Insurance Company

HP Servers + ISV

Production Servers
HP 9000 Superdome RP4440
HP Integrity RX6600

Dev/Test Servers
HP 9000 Superdome RP5470
HP Integrity RX6600

Claims per year **327,652**

$0.12 per claim →

← **$0.79 per claim**

Mainframe support staff has 6.6x better productivity

IBM System z CICS/DB2

Total MIPS **11,302**

MIPS used for commercial claims processing prod/dev/test **2,418**

Claims per year **4,056,000**

Accumulated Field Data For Labor Costs

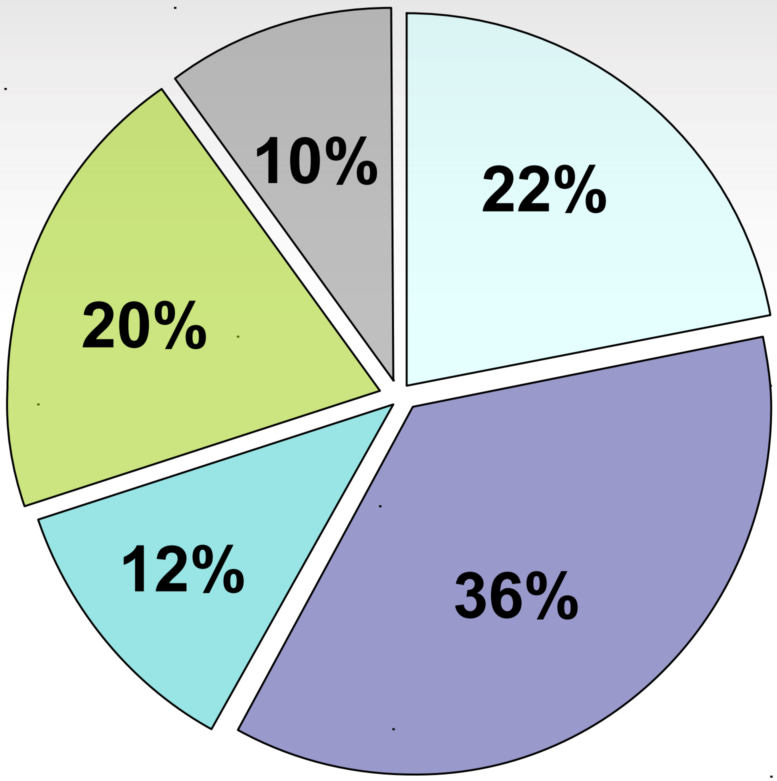
- Average of quoted infrastructure labor costs
  - **30.7** servers per FTE (dedicated Intel servers)
    - **67.8** hours per year per server for hardware and software tasks
  - **52.5** Virtual Machines per FTE (virtualized Intel servers)
    - **39.6** hours per year per Virtual Machine for software tasks and amortized hardware tasks
    - Typical 8 Virtual Machines per physical server

- Best fit data indicates
  - Hardware tasks are **32** hours per physical server per year
    - Assume this applies to Intel or Power servers
    - Internal IBM studies estimate 320 hours per IFL for zLinux scenarios
  - Software tasks are **36** hours per software image per year
    - Assume this applies to all distributed and zLinux software images

Labor model based on customer data from IBM studies

Five Key IT Processes For Infrastructure Administration

Time spent on each activity



- **Deployment Management**
  - *Hardware set-up and software deployment*
- **Incident/Capacity Management**
  - *Monitor and respond automatically*
- **Asset Management**
  - *Hardware and software asset tracking*
- **Security Management**
  - *Access control*
- **Change Management**
  - *Hardware and software changes*

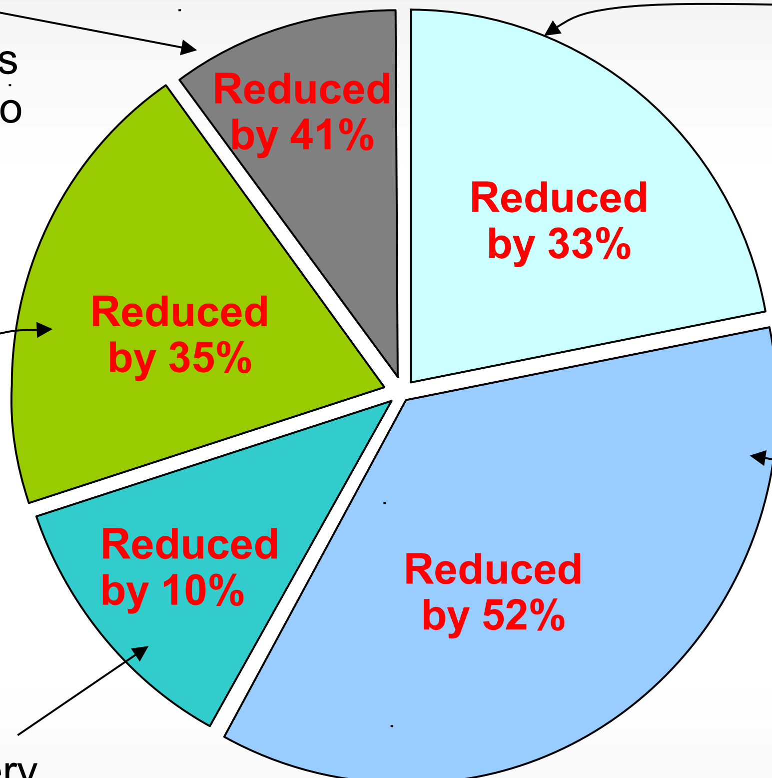
Allocation based on customer data from IBM study

zManager Labor Cost Reduction Benefits Case Study

5032 total hours per year reduced by 38% to 3111 hours per year

Change Management
Standardization of images and firmware, visibility into relationships among resources

Deployment Management
Automatic setup and configuration of the hypervisor and out-of-the-box networks



Security Management
Centralized fine-grain administrator access control

Incident/Capacity Management
Automation to isolate and fix issues

Asset Management
Automated discovery, entitlement management

TCO: Understand The Complete Picture

Thank you.

Cost Ratios in all TCO Studies

Average Cost Ratios (z vs Distributed)

		z	Distributed	z vs distributed (%)
Offload	**5-Year TCO**	**$16,351,122**	**$31,916,262**	**51.23%**
	Annual Operating Cost	$2,998,951	$4,405,510	68.07%
	Software	$10,932,610	$16,694,413	65.49%
	Hardware	$3,124,013	$3,732,322	83.70%
	System Support Labor	$3,257,810	$4,429,166	73.55%
	Electricity	$45,435	$206,930	21.96%
	Space	$59,199	$154,065	38.42%
	Migration	$438,082	$10,690,382	4.10%
	DR	$854,266	$2,683,652	31.83%
	Average MIPS	3,954	[redacted]	
	Total MIPS	217,452		
Consolidation	**5-Year TCO**	**$5,896,809**	**$10,371,020**	**56.86%**
	Annual Operating Cost	$716,184	$1,646,252	43.50%
	Software	$2,240,067	$6,689,261	33.49%
	Hardware	$2,150,371	$1,052,925	204.23%
	System Support Labor	$1,766,403	$2,395,693	73.73%
	Electricity	$129,249	$365,793	35.33%
	Space	$84,033	$205,860	40.82%
	Migration	$678,449	$0	
	DR	$354,735	$411,408	86.22%
	Average MIPS	10,821	[redacted]	
	Total MIPS	292,165		

(1) Always Compare To An Optimum System z Environment

- Updating hardware and software reduces cost
- Sub-capacity may produce free workloads
- Replace ISV software with IBM software
- System z Linux consolidation saves money
- Changing database can impact capacity requirements
- Specialty processors reduce mainframe cost

(2) Look For Not-so-obvious Distributed Platform Costs To Avoid

- Distributed servers refresh every 3 to 5 years
- Distributed server disaster recovery is typically at 100%
- Non-production environments require fewer resources on System z
- Customers often overlook significant tools replacement costs

Distributed Servers Need To Be Replaced Every 3 To 5 Years

- IT equipment refreshed 2 – 7 year intervals, normally 3 or 4 years
- Distributed servers re-purchased each time
  - Normally with some additional growth capacity (CPU, memory, I/O and other specialty cards like cryptographic offloads)
- With a growing mainframe, customers normally only have to purchase the additional (new) MIPS capacity
  - Existing MIPS are often carried over to the new hardware
  - Existing memory, I/O facilities and specialty processors / cards are also normally carried over to the new hardware
- Five year studies show this effect, short time periods do not

(3) Consider Additional Platform Differences That Affect Cost

- Mainframe blockade effects
- Cost of adding incremental workloads to System z is less than linear
- Offloading chatty applications introduces latency
- Batch challenges non-mainframes
- Cost of administrative labor is lower on System z
- System z responds flexibly to unforeseen business events
- System z cost per unit of work is much lower than distributed

Linux On System z Consolidation Usually Has Lower Costs

Which platform provides the lowest TCA over 3 years?

Oracle DB workload

3 OLTP Database Workloads, each supporting 18K tps

Oracle Enterprise Edition

Oracle Real Application Cluster

3 Oracle RAC clusters
4 server nodes per cluster

12 total HP DL580 servers (192 cores)

$13.2M (3 yr. TCA)

3 Oracle RAC clusters
4 nodes per cluster
Each node is a Linux guest
zEC12 with 27 IFLs

$5.7M (3 yr. TCA)

***Half** the cost*

TCA includes hardware, software, maintenance, support and subscription.
Workload Equivalence derived from a proof-of-concept study conducted at a large Cooperative Bank.

Cost Of Adding Incremental Workloads To System z Is Less Than Linear

- Mainframes are priced to deliver a substantial economy of scale as they grow
- Doubling of capacity results in as little as a 30% cost growth for software on z/OS
- Average Cost is significantly more than incremental cost