# zEnterprise –
# The Ideal Platform For Smarter Computing

## TCO Lessons Learned From Customer Engagements

# IBM Eagle Team Helps Customers Find The Lowest Cost Solution

- 8 senior technical IT staff performing TCO customer studies worldwide

- Over 250 TCO studies in the past 5 years

- Study categories

  ▶ Offload System z workloads to distributed

  ▶ Placement of new workloads (Fit for purpose)

  ▶ Consolidate workloads to zLinux

- Contact us via Craig Bender (csbender@us.ibm.com)

- No charge

**All the examples in this module are from actual customer studies**

# Typical Eagle TCO Study Approach

## Approach

- Establish scope of study – applications, platforms, etc.

- Gather information

- Build the cost model

- Review with customer and iterate as needed

- Produce final report

## Process

- Meet with customer face-to-face to establish scope and gather information (1/2 day).  **Kick off and set expectations.**

- Complete study (30 day target)

# To Understand <u>Total Cost</u>
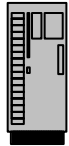# Four Dimensions Of Cost Should Be Considered

- Components

- Environments

- Time Factors

- Non-Functional Requirements / Qualities of Service

# IT Solutions Require 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**
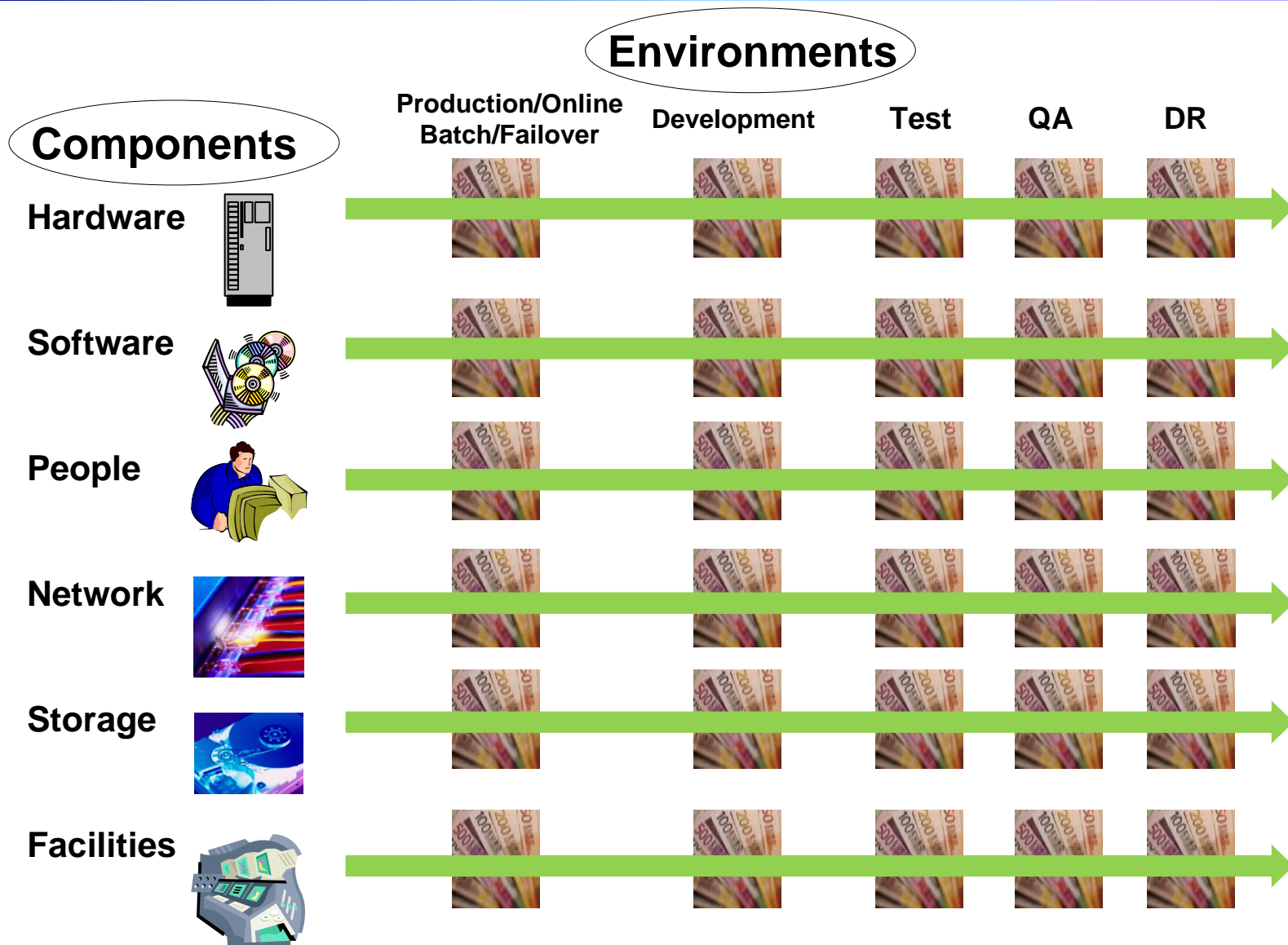
ECKD, FBA, SAN, Compressed, Primary, secondary
Disk (multiple vendors), tape, Virtual, SSD

**Facilities**

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

# Environments Multiply Components



Environments

| Components | Production/Online Batch/Failover | Development | Test | QA | DR |
|---|---|---|---|---|---|
| Hardware | | | | | |
| Software | | | | | |
| People | | | | | |
| Network | | | | | |
| Storage | | | | | |
| Facilities | | | | | |

# Time Factors Drive Growth And Cost

- **Migration time and effort**

- **Business organic growth and/or planned business changes effect 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
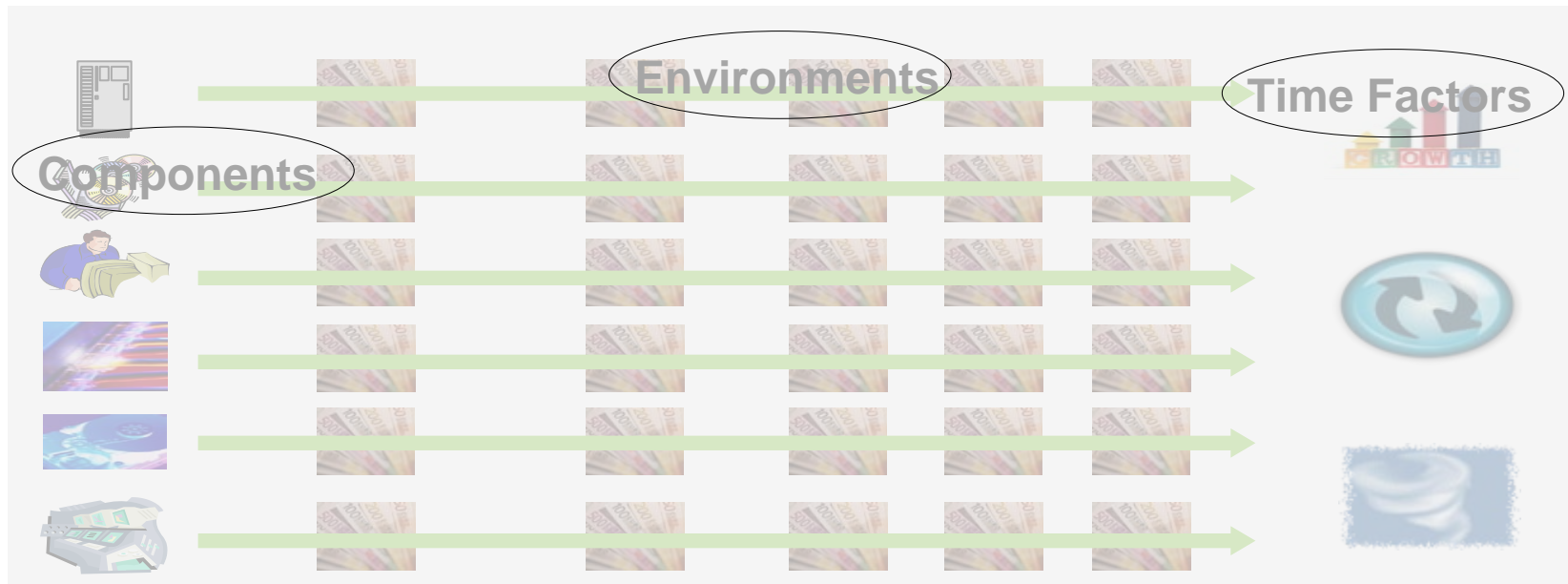


Availability …Security …Resiliency …Scalability …

**Qualities of Service, Non-Functional Requirements**

# Understand The Complete Picture

**Components**

**Environments**

**Time Factors**

**Qualities of Service, Non-Functional Requirements**

Availability …Security …Resiliency …Scalability …

And then compare the alternatives on a level playing field

# TCO Lessons Learned

- Make sure you are comparing to an optimum z environment
  - ▶ Currency reduces cost
  - ▶ Specialty processors and appliances revolutionize mainframe cost
  - ▶ Sub-capacity may produce free workloads
  - ▶ Replacing ISV software with IBM is a better deal
  - ▶ System z Linux consolidation is often a no-brainer

- Don't forget to consider these platform cost differences
  - ▶ Cost of adding incremental workloads to System z is less than linear
  - ▶ New mainframe workloads with unbeatable price points via Solution Edition
  - ▶ Distributed servers need to be replaced every 3 to 5 years
  - ▶ Changing database can have dramatic capacity impacts
  - ▶ Disaster Recovery can be more expensive than you might think without a mainframe
  - ▶ Chatty applications are not good offload candidates
  - ▶ Customers often overlook significant tool replacement cost
  - ▶ Security breaches have high costs
  - ▶ High offload costs almost never pay back in operational savings
  - ▶ Late migration projects extend dual systems cost
  - ▶ Non-production environments require fewer resources on the mainframe
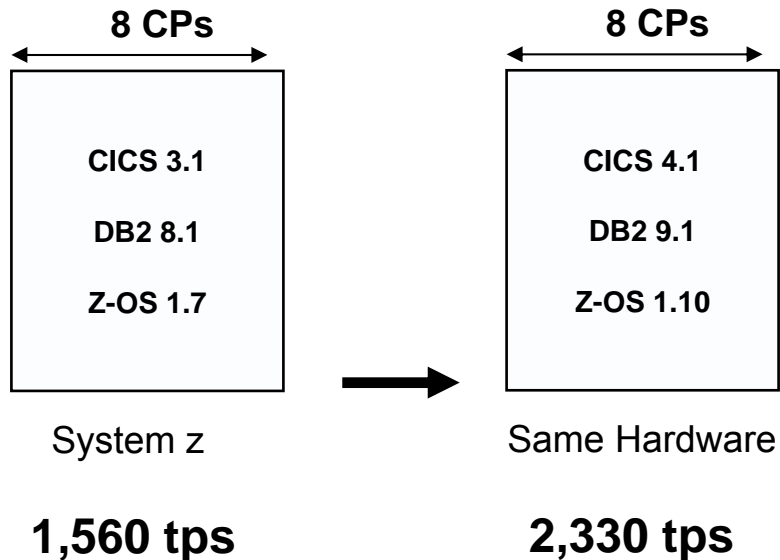  - ▶ Mainframe cost per unit of work much lower than distributed

# Currency Reduces Cost – Hardware

**Accumulated Cost Comparison**
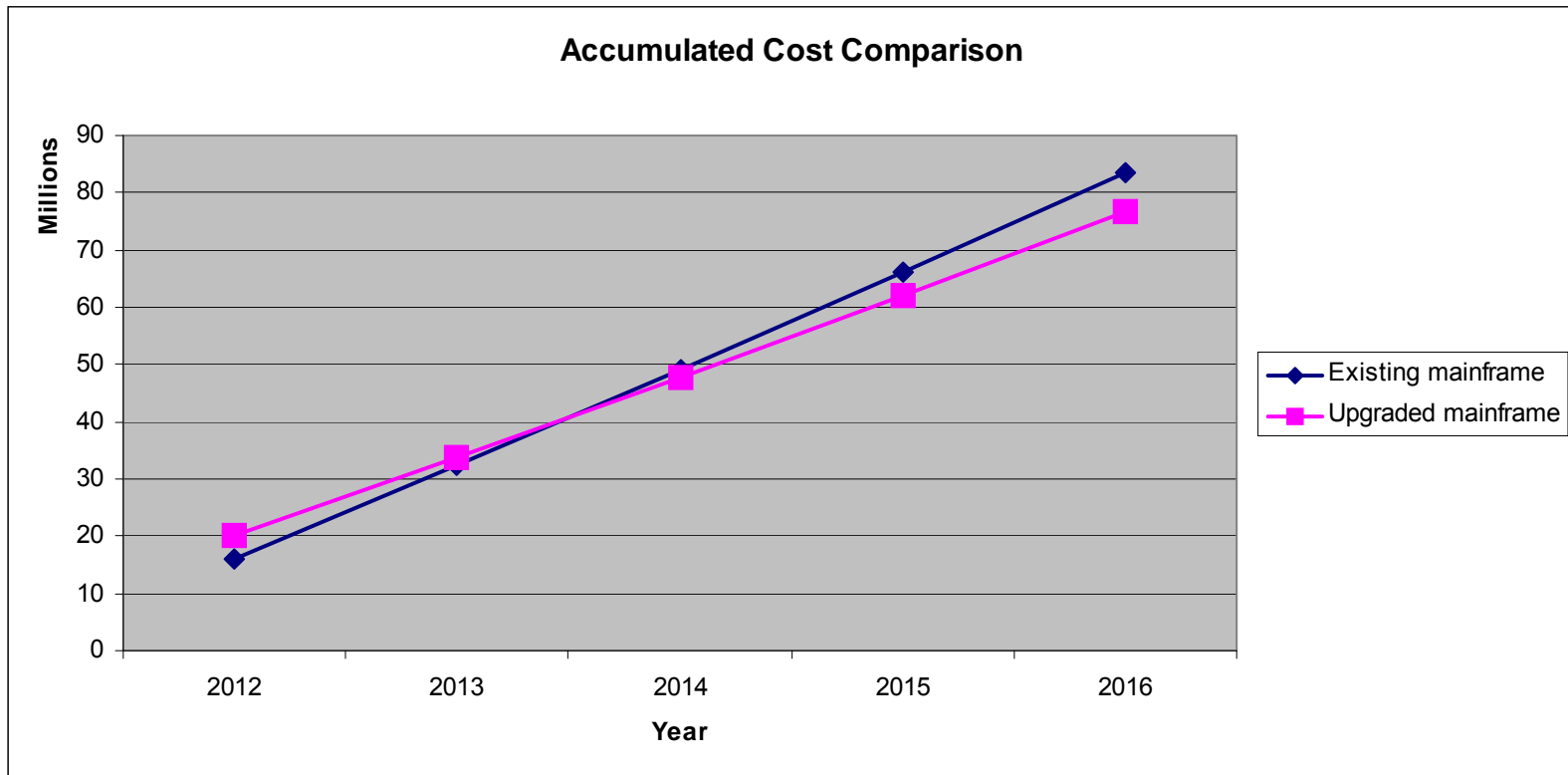


- Typical customer hardware refresh scenario
  - ▶ 2 generations, from z9 to z196
  - ▶ 2M investment pays back >1M savings every year – most cases positive in a 3 year period
  - ▶ Savings from technology dividends and specialty processor offload

- Comparing latest technology servers to old mainframes is unfair but often done

# Currency Reduces Cost – Software

**Benchmark – Only Software Changes:**

8 CPs → 8 CPs

| System z | Same Hardware |
|---|---|
| CICS 3.1 | CICS 4.1 |
| DB2 8.1 | DB2 9.1 |
| Z-OS 1.7 | Z-OS 1.10 |

**1,560 tps** → **2,330 tps**

**Comments:**
- Single LPAR configuration with CICS and DB2 collocated
- COBOL code recompiled with later COBOL compiler
- Later software (eg. DB2 v10) will be even better!

**A similar real customer example:**

- z/OS 1.6 across tens of LPARs delayed upgrade for 9 months due to "cost of upgrade effort"

- z/OS 1.8 reduced each LPAR's MIPS 5% mainly due to improvements in RMF/SMF code inside z/OS

- Monthly software cost savings paid for the upgrade effort almost immediately

## Conclusion:
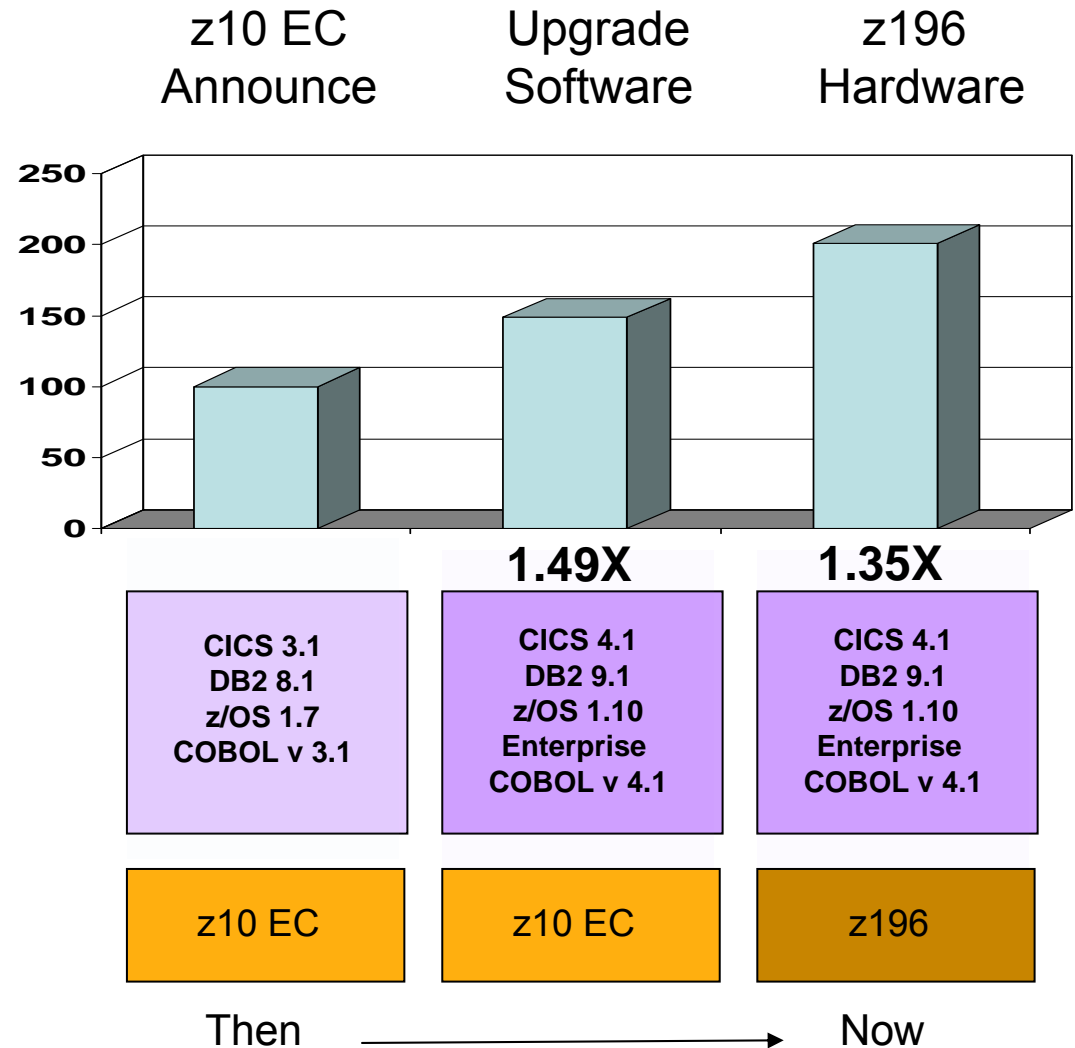## Keeping current saves money

# Currency Reduces Cost – Don't Forget About Tuning And Newer Compilers

- For core systems of record, raw throughput can affect the scalability of the whole organization

- IBM focus on tuning
  - ▶ Targets for release delta is -5% pathlength
  - ▶ Continued investment in compiler and JIT performance
  - ▶ Seeking out more hardware/software optimization and offload opportunities

- DB2 v10 is a good example
  - ▶ Between 5% and 10% CPU reduction for traditional workloads
    - − BMW Autos found 38% pathlength reduction for their heavy insert workload
  - ▶ 10x number of users by relieving memory constraints
  - ▶ IDAA appliance in zEnterprise for DB2 offloads too

# Currency Reduces Cost – CICS/DB2

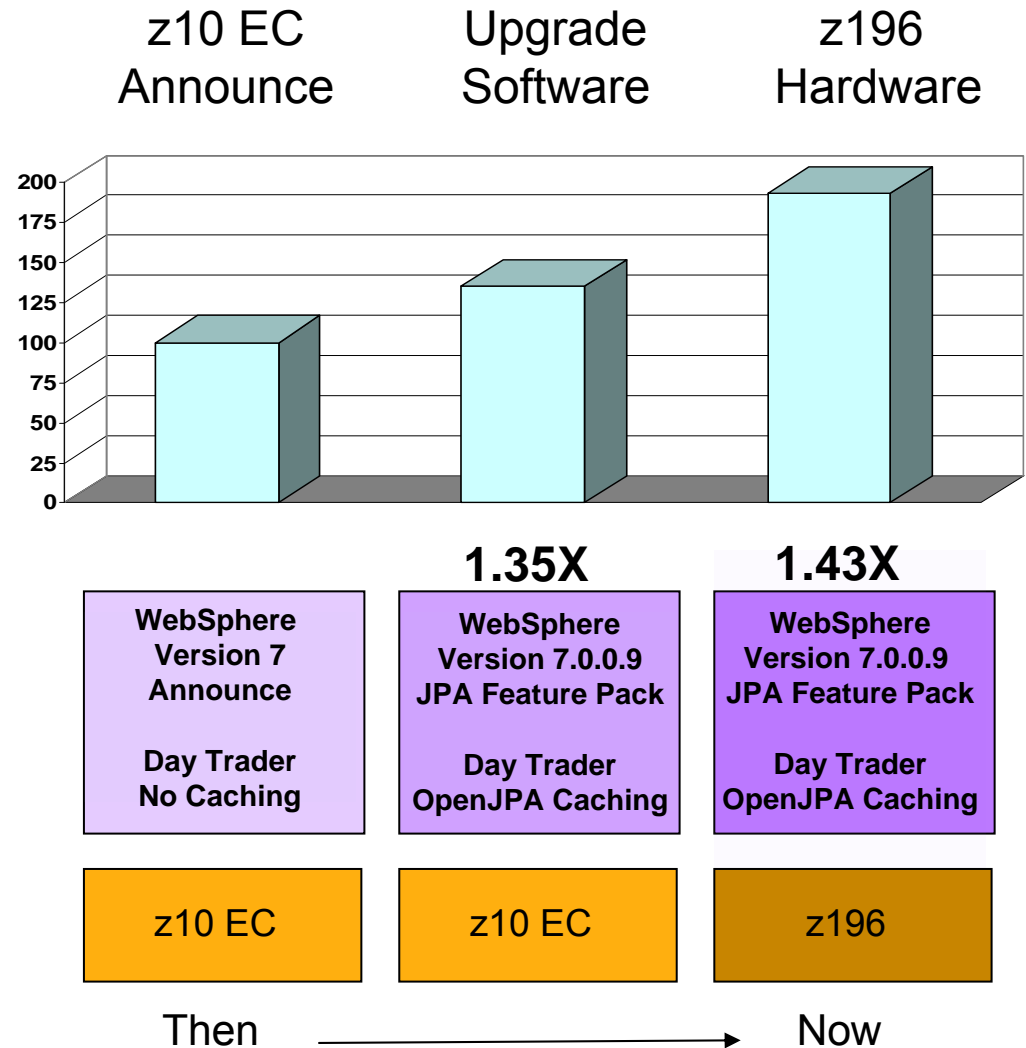**Continued investment to optimize key software for z/OS environment**

- Upgrade CICS/DB2 stack produces 1.49 times performance improvement on same z10 hardware
- Move to z196 hardware produces 1.35 times performance improvement
- From then to now – **2.01** times performance improvement



|  | z10 EC Announce | Upgrade Software | z196 Hardware |
|---|---|---|---|
|  |  | 1.49X | 1.35X |
|  | CICS 3.1 DB2 8.1 z/OS 1.7 COBOL v 3.1 | CICS 4.1 DB2 9.1 z/OS 1.10 Enterprise COBOL v 4.1 | CICS 4.1 DB2 9.1 z/OS 1.10 Enterprise COBOL v 4.1 |
|  | z10 EC | z10 EC | z196 |

Then ⟶ Now

# Currency Reduces Cost – WebSphere z/OS

**Continued investment to optimize WebSphere software for z/OS environment**

- 1.35 times performance improvement for JPA 2.0 applications that exploit the OpenJPA caching facilities available in the WebSphere Version 7 JPA Feature Pack.

- Move to z196 hardware produces 1.43 times performance improvement

- From then to now – **1.93** times performance improvement

| z10 EC Announce | Upgrade Software | z196 Hardware |
|---|---|---|



| | **1.35X** | **1.43X** |
|---|---|---|
| WebSphere Version 7 Announce<br><br>Day Trader No Caching | WebSphere Version 7.0.0.9 JPA Feature Pack<br><br>Day Trader OpenJPA Caching | WebSphere Version 7.0.0.9 JPA Feature Pack<br><br>Day Trader OpenJPA Caching |
| z10 EC | z10 EC | z196 |

Then ⟶ Now

# Currency Reduces Cost – WebSphere On System z Linux

Similar results are achieved for WebSphere software in a Linux for System z environment
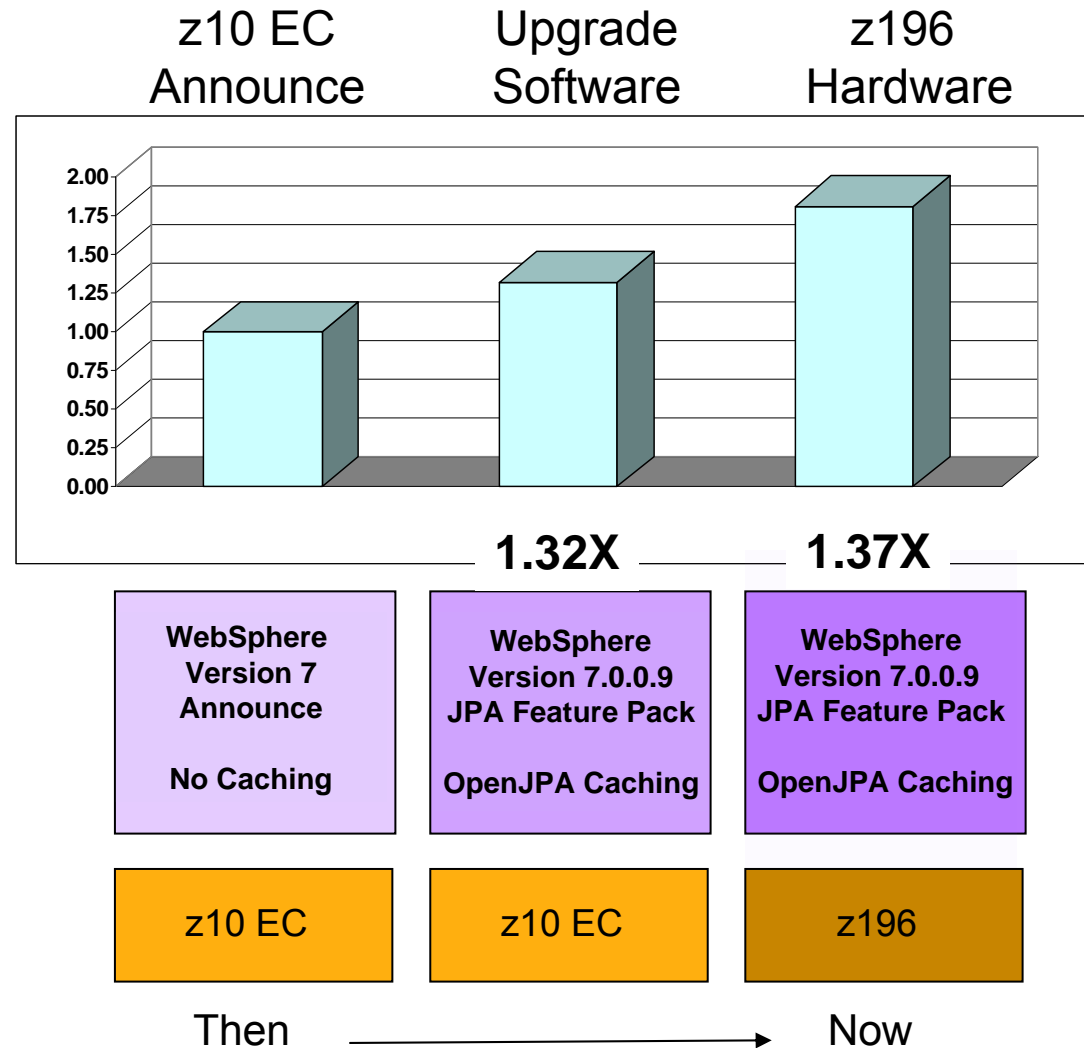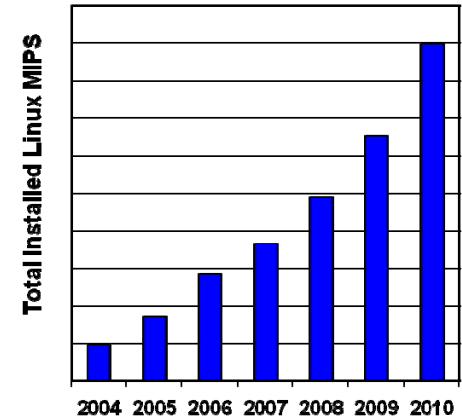
- 1.32 times performance improvement for JPA 2.0 applications that exploit the OpenJPA Caching facilities available in the WebSphere Version 7 JPA Feature Pack.
- Move to z196 hardware produces 1.37 times performance improvement
- Combined hardware and software - **1.81** times performance improvement

| z10 EC Announce | Upgrade Software | z196 Hardware |
|---|---|---|



**1.32X**          **1.37X**

| WebSphere Version 7 Announce<br><br>No Caching | WebSphere Version 7.0.0.9 JPA Feature Pack<br><br>OpenJPA Caching | WebSphere Version 7.0.0.9 JPA Feature Pack<br><br>OpenJPA Caching |
|---|---|---|
| z10 EC | z10 EC | z196 |

Then     ⟶     Now

# Specialty Processors And Appliances Revolutionize Mainframe Cost
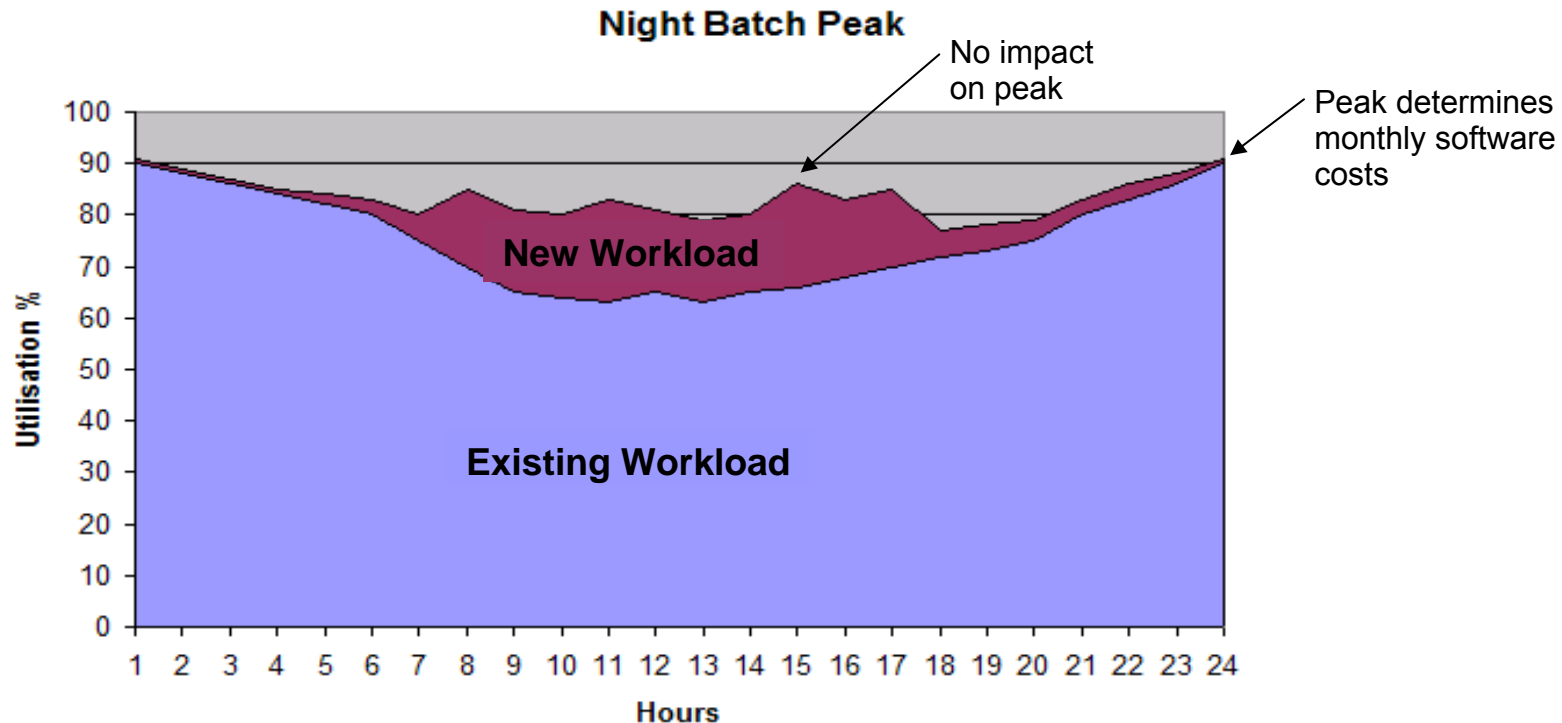
**Installed Linux MIPS**

- ■ Special assist processors for System z
  - ▶ For Java or XML workloads (zAAP)
  - ▶ For selected data, networking and security workloads (zIIP)
  - ▶ For Linux workloads (IFL)

- ■ Attractive pricing
  - ▶ Hardware is from $35K per processor one time charge
    - – Considerably less cost than a general purpose z/OS processor
  - ▶ No charge for IBM software running on zAAP/zIIP
  - ▶ IBM software on an IFL costs 120 PVU's (less than an Intel dual core)

2004 2005 2006 2007 2008 2009 2010

Linux now 18% of all installed MIPS

- ■ Cost impact of zAAP/zIIP depends on offload percentage
  - ▶ Between 40%-95% is customer experience
  - ▶ Tuned to task – for example, 3-5x pathlength expansion of Java countered with 80% offload to zAAP

- ■ IFL processors have had a big impact on Enterprise Linux
  - ▶ Over 80% of the top 100 System z clients are running Linux on the mainframe
  - ▶ This is now the lowest cost place to run most IBM software
  - ▶ Fantastic UNIX-style workload consolidation option, or a good place to co-locate UNIX-style apps with z/OS data and services

# Sub-Capacity May Produce Free Workloads

**Night Batch Peak**

No impact on peak

Peak determines monthly software costs

New Workload

Existing Workload

Utilisation %

100 90 80 70 60 50 40 30 20 10 0

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

**Hours**

- 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!

# Replacing ISV Software With IBM Is A Better Deal

- This medium-sized financial company in Europe started with a typical mixed software environment, matching our average profile shown on the left

- Migrating to IBM tooling changed their software cost profile to that shown on the right

| Average Profile (BEFORE) | | |
|---|---|---|
| Weighted MIPS | | 8,800 |
| Cost Per MIPS per Year | | Profile |
| IBM Software | 1,000.00 | 24.72% |
| | | 0.00% |
| ISV Software | 1,540.00 | 38.07% |
| TOTAL SW | 2,540.00 | |

| Actuals (AFTER) | | |
|---|---|---|
| Weighted MIPS | | 8,900 |
| Cost Per MIPS per Year | | Profile |
| IBM OTC | 376.09 | 13.66% |
| IBM MLC | 1,023.77 | 37.20% |
| ISV Software | 136.09 | 4.94% |
| TOTAL SW | 1,535.95 | |

- IBM software costs are now slightly higher than the average customer but ISV software costs are dramatically lower – saving $1000/MIPS per year from their annual software bill

# Replacing ISV Software With IBM Is Also A Better Deal Than Offloading

|  | Mainframe Offload | Move To IBM Tooling |
|---|---|---|
| Investment $, Period | $54M over 2 years | $3M over 1 year |
| Predicted Annual Cost Savings | $13M from year 3 | $6M from year 2 |
| 5 Year TCO, Breakeven | $140M, year 7 | $101M, year 2 |
| Level Of Risk | Very High | Very Low |

- **Finance perspective**
  - ▶ **Large, risky and expensive project with distant payback under unlikely assumptions, versus**
  - ▶ **A small, low risk and cheap project with instant payback**

# System z Linux Consolidation Is Often A No-Brainer

- Large financial services company with a mixture of dedicated (Oracle) and virtualized (WAS products) environments on Intel
  - ▶ Scaling out rapidly – up to 172 images on 836 cores

|  | Refresh existing x86 | Consolidate on zLinux | Observation |
|---|---|---|---|
| Software | 9.76M | 3.70M | Software costs down 62% |
| Hardware | 2.32M | 5.22M | Hardware costs up 125% |
| Labor/Facilities | 3.83M | 0.69M | Charge to department down 82% |
| Migration | 0.18M | 0.41M | |
| Cost Avoidance | - | -0.61M | |
| **Total** | **16.1M** | **9.41M** | |

- 5 Year Savings: 6.6M, with financing delivering savings in year 1 budget
- Achieve 26:1 core consolidation from virtualized x86 to zLinux
- 5 Year IBM Solution Edition pricing including two new mainframes in a pair of data centers

# 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 more than incremental cost

  ▶ e.g. $3000/MIPS/yr "rule of thumb" average vs $600 incremental
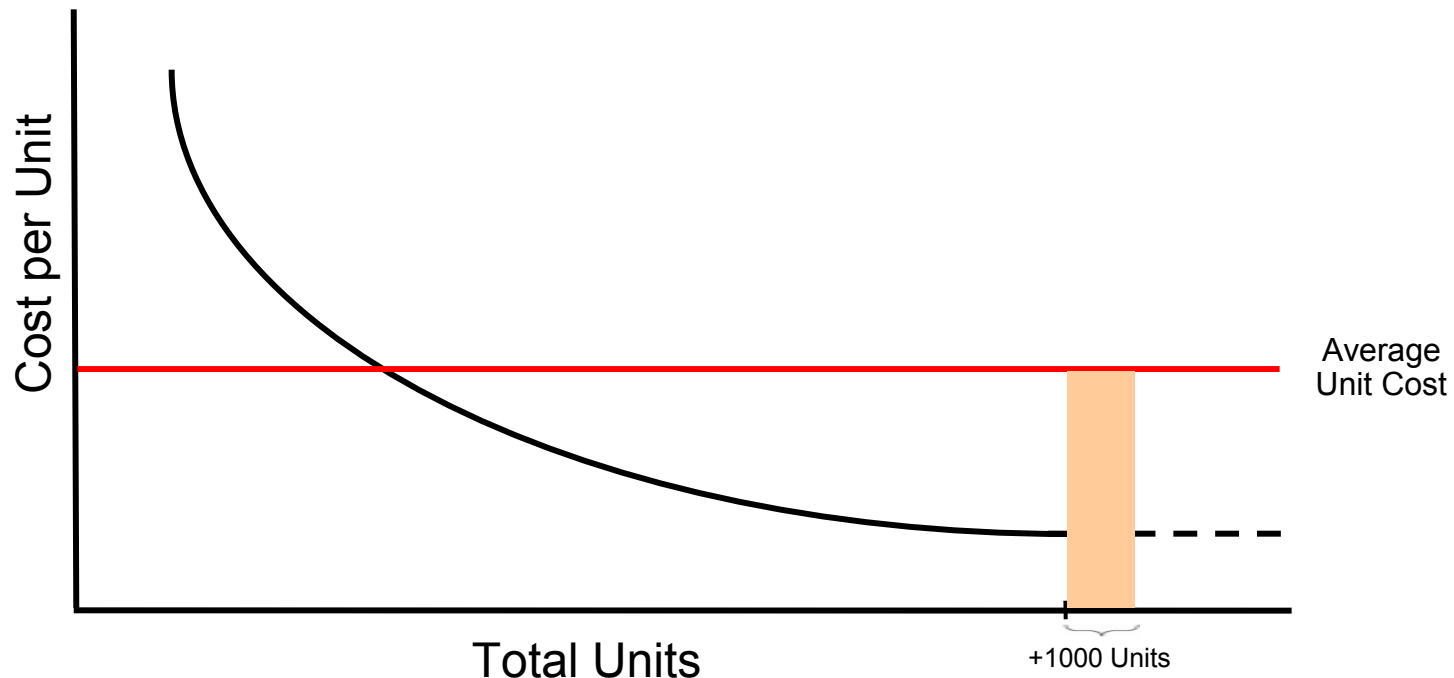
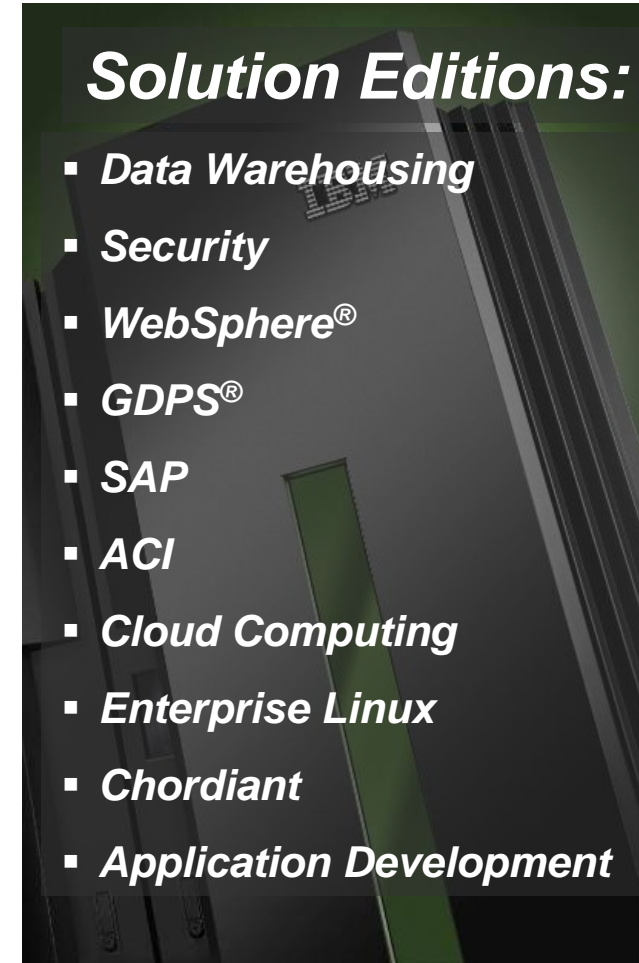# Cost Of Adding Incremental Workloads To System z Is Less Than Linear (Example)

- Customer determined current mainframe and distributed WAS application costs are similar

- Examined the incremental cost of adding one more large WAS application to each platform

- The mainframe demonstrated a clear advantage

| | Incremental Mainframe | Incremental Distributed |
|---|---|---|
| **5 Year TCO** | **1.29M**<br>**(657K OTC, 42K Y1, 147K Y2-5)** | **1.56M**<br>**(378K OTC, 192K Y1, 249K Y2-5)** |

- Although moving existing WAS applications between platforms is unlikely, future WAS deployments will therefore be targeted to the mainframe

# New Mainframe Workloads With Unbeatable Price Points Via Solution Edition
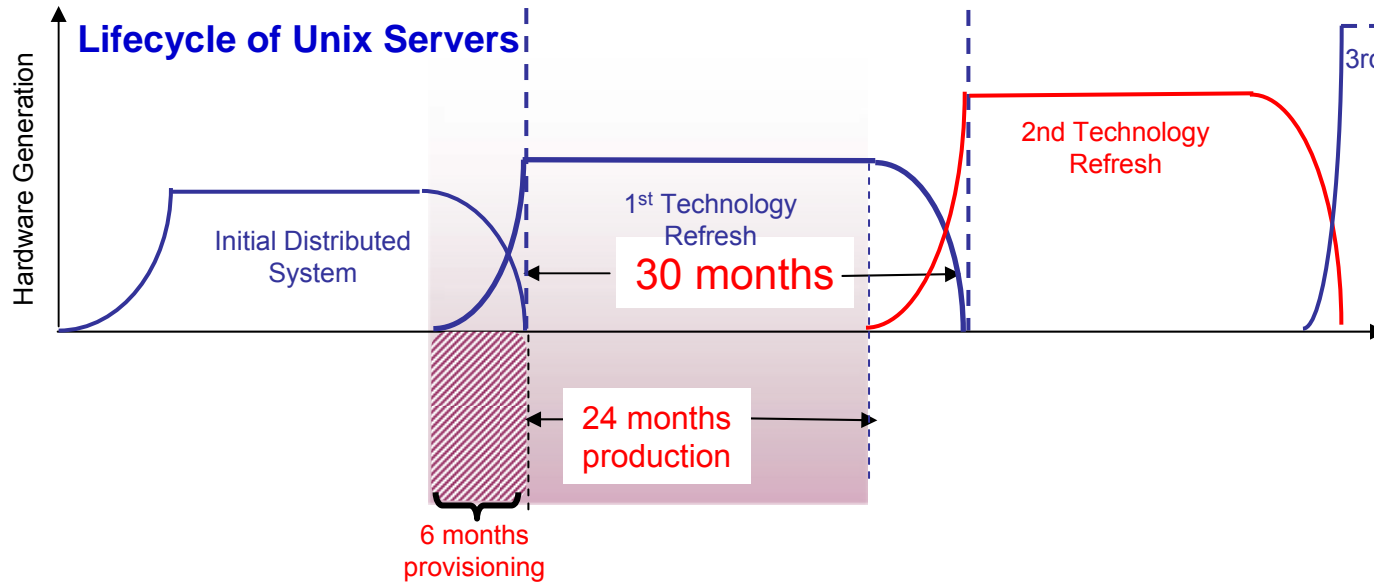
- Bundle of System z **hardware, software** and **maintenance**
  - ▶ 3 or 5 year **Best Price**

- Focus: new System z workload opportunities
  - ▶ Not for existing workloads

- Solution Editions usually include:
  - ▶ System z hardware (new footprint or incremental)
  - ▶ Prepaid hardware maintenance
  - ▶ Comprehensive middleware stack (including subscription and support)
  - ▶ Storage as an option for all Solution Editions
  - ▶ Services for some Solution Editions

*Solution Editions:*

- *Data Warehousing*
- *Security*
- *WebSphere®*
- *GDPS®*
- *SAP*
- *ACI*
- *Cloud Computing*
- *Enterprise Linux*
- *Chordiant*
- *Application Development*

# 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

# Distributed Servers Need To Be Replaced Every 3 To 5 Years (2)

**Lifecycle of Unix Servers**

Hardware Generation

Initial Distributed System

1st Technology Refresh

**30 months**

2nd Technology Refresh

3rd Technology Refresh

24 months production

6 months provisioning

**Lifecycle of Mainframe Generations**

Hardware Generation

30 months

30 months

30 months

Initial Mainframe System

1st Technology Refresh

2nd Technology Refresh

1 Weekend upgrading to new hardware and patch levels

Time

30 months production

No need to retire the server, upgrade in place.

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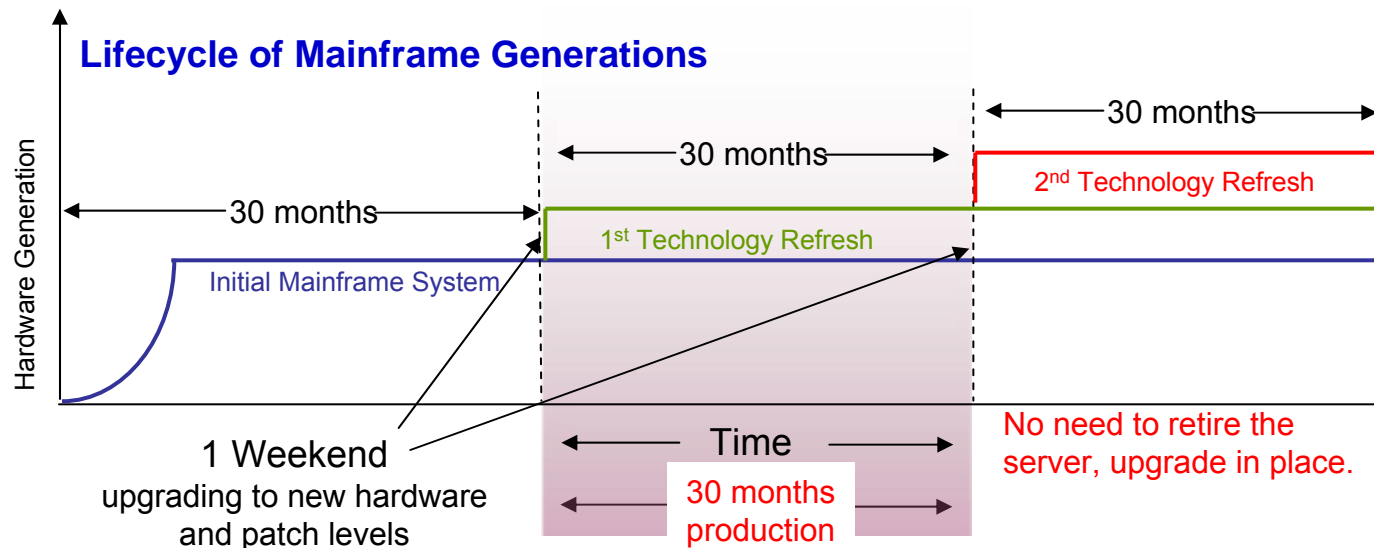
The mainframe by contrast is upgraded over a weekend and is fully productive at all times

# Changing Database Can Have Dramatic Capacity Impacts

- Asian customer migrated large IMS DB to DB2
  - Capacity (MIPS) requirement for the database doubled (2x)

- European customer reported a 3x storage requirement increase migrating IMS to Oracle RAC, as well as a huge growth in processor cores

- Small European customer rehosted a mainframe which once used IDMS but had been migrated to DB2 a few years prior to the rehost
  - This customer broke our record for the highest MIPS-to-RPE (distributed performance units from Ideas International) ratio
  - This suggests that mechanical conversion of code, which rehosting is supposed to be to maintain the "business logic", adds yet more overheads onto any existing overheads already in place – badly written applications get worse still!
  - Application and data modernization in place are faster, safer and more efficient – and normally cheaper to!

# Disaster Recovery Can Be More Expensive Than You Might Think Without A Mainframe

- Disaster Recovery costs – DR provisioning and DR testing
- DR testing example
  - ~ 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 ~ 20 minutes

- Conclusion: Mainframe both simplifies and improves DR testing

# Chatty Applications Are Not Good Offload Candidates

- "Chatty" applications - frequent reads and/or updates of one or more data sources

- Customer and offload vendor proposed a hybrid rehost
  - ▶ Move high MIPS CICS online workload to Linux on System z while retaining DB2 and some VSAM data on z/OS

- Application architecture:
  - ▶ Most CICS application programs access some VSAM and zDB2 data
  - ▶ Individual programs have a chatty data access profile
  - ▶ CICS transactions typically execute <100ms, the mainframe component of an end-to-end user interaction accounts for no more than 20% of the total

- Conclusion: "breaking apart" the CICS transaction programs is not a good idea because of induced latency impacts on SLA

# Customers Often Overlook Significant Tool Replacement Cost

- Customers often struggle to identify all the replacement tools and middleware they will need for an offload

- Straight-line extrapolation of cost from the easily identified subset is often accurate enough

- Customer example: 261 total software products on z/OS
  - ▶ 37 product replacements identified in vendor proposal and IBM identified an additional 16 for a total of 53 products of 261 (20%)

- 208 products missing – how to estimate their likely cost, especially given that not all products will end up with one-for-one replacements:
  - ▶ Applications may be re-written to not need missing products
  - ▶ New code could be written to perform the function from scratch
  - ▶ Adding operations labor to manually do the function could be an option

- We extrapolated from the known products cost and a few years later were proven to be very close to the mark

# Security Breaches Have High Costs

- Security breach was a result of theft of physical hard drives which contained customer and agent data (data, audio and video files)
  - ▶ Fifty-seven drives stolen from a data closet in a leased call center office

- Government regulations required:
  - ▶ Letters should be sent to customers and agents
  - ▶ Various local and national authorities needed to be notified
  - ▶ Fines can be levied (but weren't in this case)

- Security breach was conservatively estimated to cost $10M, including:
  - ▶ Diagnosing exposure
    - − Approximately 700 people worked on identifying "what and who" was breached
  - ▶ Restoring lost files from backups
  - ▶ Phone calls from customers and documenting issues
  - ▶ Sending letters to customers and agents (~$2M estimated, >1M letters)
  - ▶ Replacement hardware (~$6M, switched to encrypted drives in all machines / SAN, requiring other hardware and software upgrades across the organization)
  - ▶ Credit monitoring (volunteered by the organization; not required by law)

# High Offload Costs Almost Never Pay Back In Operational Savings



**Accumulated Cost Comparison**
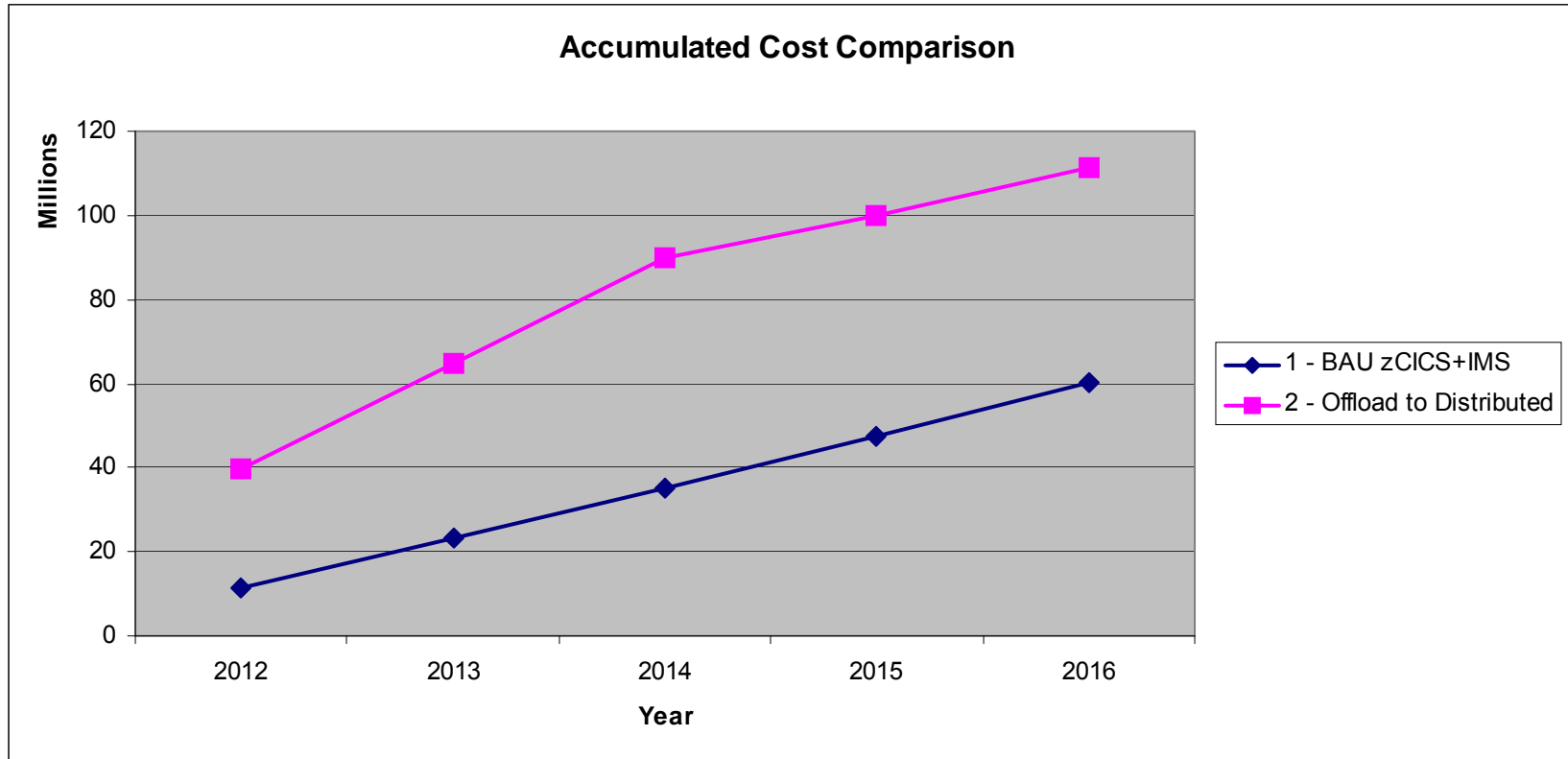
- Typical mainframe offload cost profile with huge investment cost
- Once migration completes the operating cost is similar to the current mainframe – NO ROI
- Example is existing old mainframe to new distributed

# Late Migration Projects Extend Dual Systems Cost

- Customer objective: Offload 3,500 MIPS with Micro Focus $10M budget and 1 year schedule

- 18 months later they had spent $25M and moved only a 10% of their MIPS

- Additional costs came from
  - Internal staff to cover the overrun
  - Substantial manual steps replaced mainframe automation
  - Needed many additional software products
  - Ended up acquiring additional distributed capacity over initial prediction (just to support the 10% they actually offloaded)
  - Extending the dual-running period of the rehost

- Executive sponsor lost their job

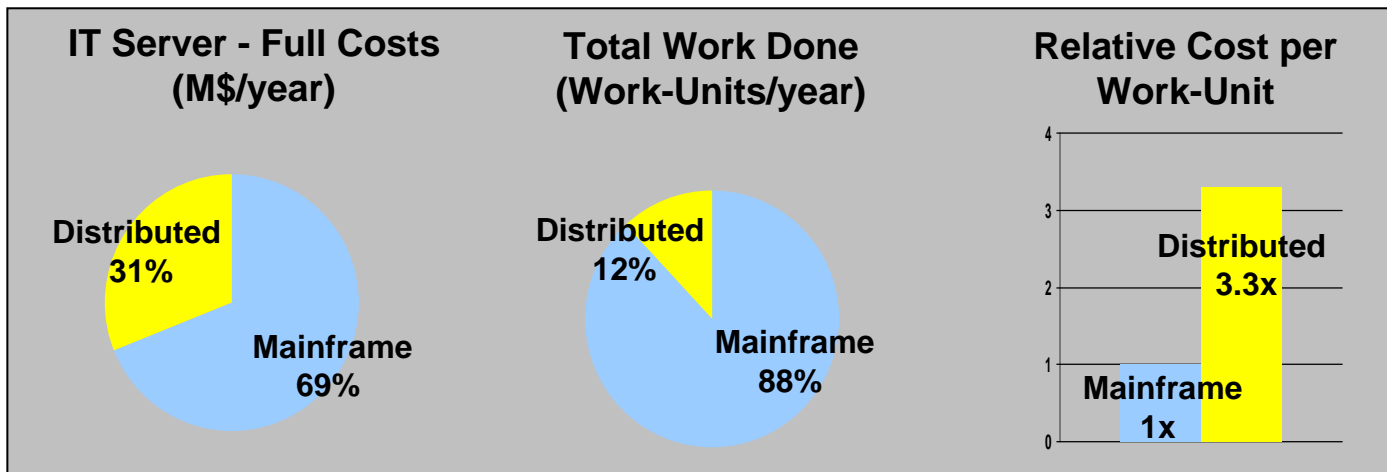# Non-Production Environments Require Fewer Resources On The Mainframe

- **Development and Test capacity**
  - ▶ Centralized – Prod+25%
  - ▶ Distributed – a range, but around Prod+150%

- **High Availability mechanisms for Production**
  - ▶ Dedicated failover (Prod x 2.5)
  - ▶ N+1 clustered (Prod x 2 worst case)
  - ▶ Mainframe (usually Prod x 1, sometimes less!)

# Mainframe Cost Per Unit Of Work Much Lower Than Distributed
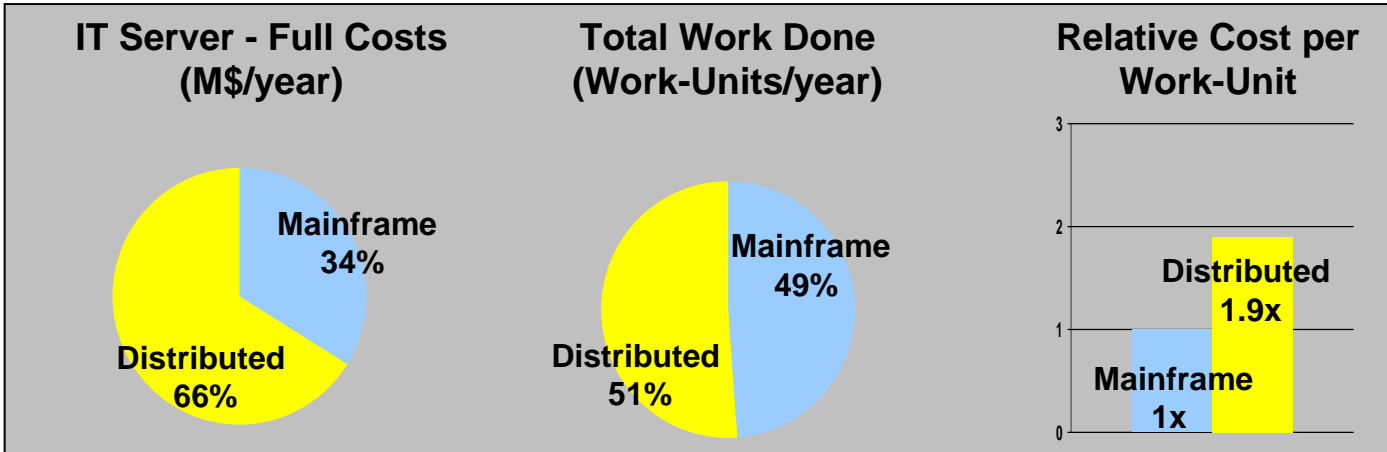
**Case 1:**
Very large retail bank
total of service delivery plus
application development

Mainly UNIX distributed
(>5000 servers)

### IT Server - Full Costs (M$/year)

Mainframe 34%
Distributed 66%

### Total Work Done (Work-Units/year)

Mainframe 49%
Distributed 51%

### Relative Cost per Work-Unit
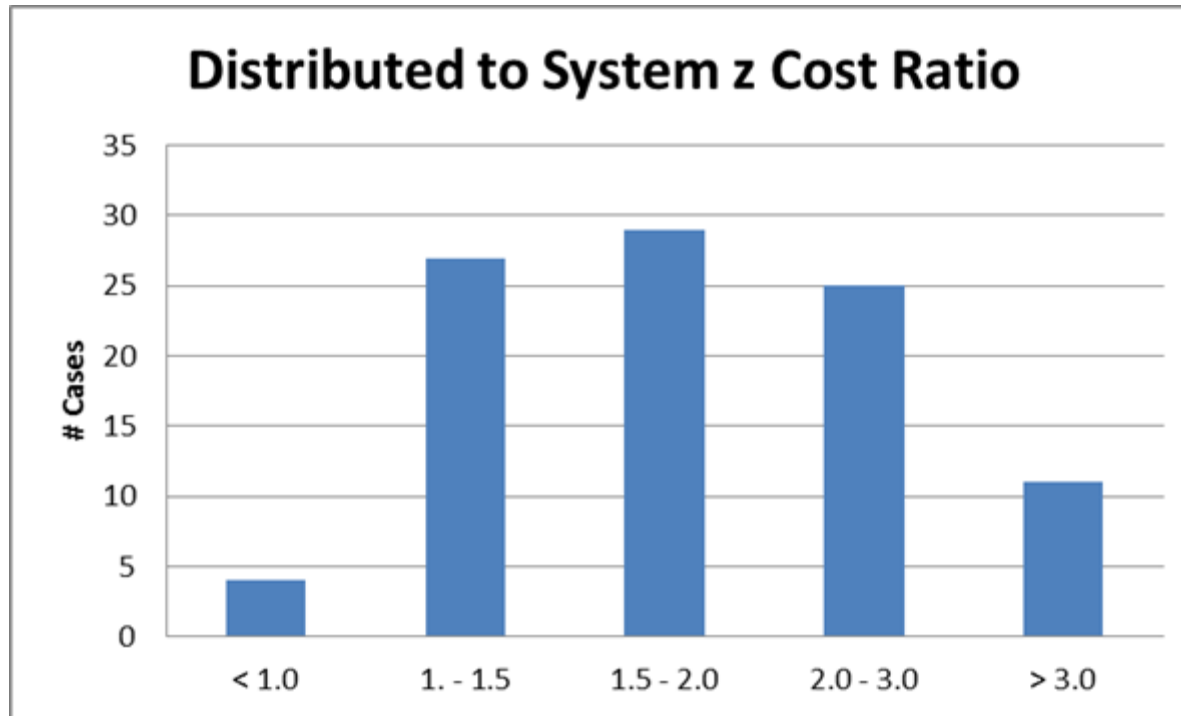
Distributed 1.9x
Mainframe 1x

**Case 2**
Medium retail bank
service delivery only
Mainly mainframe
2500 mips, 13 M txns/day
Limited distributed servers
Windows + some UNIX
(~350 servers, 12% util'n)

### IT Server - Full Costs (M$/year)

Distributed 31%
Mainframe 69%

### Total Work Done (Work-Units/year)

Distributed 12%
Mainframe 88%

### Relative Cost per Work-Unit

Distributed 3.3x
Mainframe 1x

Data from 3Q06 Scorpion studies

**The distributed IT Total Cost/Work-Unit is approx. 2-3x Mainframe Cost/Work-Unit**
**The Mainframe typically does more work, Distributed has a lot of supporting infrastructure**

# Numerous TCO Studies Prove These Learned Lessons



Distributed to System z Cost Ratio

- 97 total customer studies

- Average cost of distributed alternative is 2.2 times greater than System z

- Only 4 out of 97 studies showed lower costs on distributed