Multi-Platform-Inclusive IT Optimization Assessment Methodology

Monte Bauman
IBM Columbus
mbauman@us.ibm.com

Friday, March 16, 2012: 09:30AM - 10:30AM, OMNI Hazelnut
Session Number 10491
Topics

1. Abstract
2. Multi-Platform-Inclusive IT Optimization Assessment Methodology Overview
3. Architectural Analysis
4. Requirements Analysis
5. Cost Analysis
6. Case Study
7. Case Study - Architectural Analysis
8. Case Study - Requirements Analysis
9. Case Study - Cost Analysis
10. Conclusion
Abstract
Abstract

• It is great having choices. Choices are the harvest of competition, driving innovation and price/performance (and we all want that!).
• But making choices in the vast - complex - fast moving technology and business space that is "IT" is difficult and inefficient and all too often ineffective. The results of ill-made choices can have long lasting dire effects including failed projects, blown budgets, delayed timelines, and worse (i.e. careers veering off course).
• In this presentation, our speaker will suggest an approach for making IT decisions that is architectural-based, requirements-oriented, and platform-inclusive. Our speaker's suggested approach has three parts:
  (1) A suggested taxonomy for depicting (i.e. drawing) an application architecture including its code (and data components), the containers (where the code runs), the connectors (communications between containers), and platforms (hardware and hypervisors) ... cognizant of application tiers AND application development lifecycle stages
  (2) A "local-factors"-based requirements analysis of leading architectural options (derived in step (1)) resulting in a requirements scorecard providing a relativistic rating of the architected options. Requirements are solicited from the varied and multiple viewpoints of a project (architects, developers, engineers, operations, etc. ... these "local people" provide the facts and information that are the "local factors").
  (3) A "TCO"-based analysis, creating a technically-sound and requirements-equitable cost of ownership scorecard for the top-rated architecture options (derived from step (2)).
• The result of the approach is the derivation of decision-making artifacts including understandable architectural diagrams, a requirements scorecard, and a cost scorecard, wherein the information required to assess the "best fit" (the best "choice") is brought together in an "inclusive" manner (inclusive of all viable architectural elements, AND inclusive of all local-factor-based requirements).
End of Section
Multi-Platform-Inclusive IT Optimization Assessment Methodology Overview
Platform Placement
IT Optimization Challenges

• (Platform Placement) Choices are great to have…
• Robust server marketplace
  • x86
  • System z
  • POWER
  • SPARC
  • etc.
• Emerging “Cloud” marketplace
  • Private (see above)
  • Public
• Middleware everywhere (portable containers)
• Choices are (u-pick) … hard fun time-consuming painful!

Platform Placement
Mistakes are Expensive
New Server Form Factors

- Pre-Integrated Servers
- Heterogeneous Servers
- Enclosed Clusters
- Single System Images
- Robust Hypervisors
- Appliances

Examples:
- zEnterprise
- Cisco UCS
- More coming
Building a System
Then...

Network

Server
- Application
- Middleware
- Operating System
- Hardware

Storage

Systems Management

Topics
Building a System
Now…

Singe System Image (Systems Management)

Enclosed Cluster (Network)

- Application
- Middleware
- Operating System
- Hypervisor
- Hardware

Systems Management

Topics
Evaluating a System

Build the System

Operate the System

Maintain and Enhance the System

Retire the System

Time (the project plan)

Risk (acceptable risk plan)

Capability (functionality)

Cost (the project budget)

Topics

Atlanta 2012
IT Optimization Assessment Methodology Overview

- Requirements Analysis
- Architecture Analysis
- TCO Analysis
IT Optimization Assessment Methodology

Factors

Topics

Architecture Analysis

Requirements Analysis

Options

Local Factors

Cost Factors

Cost Analysis
IT Optimization Assessment Methodology
IBM Offerings / Workshops

Topics

- Architecture Analysis
- Requirements Analysis
- Options
- Cost Analysis
- Right-Fit
- Local Factors
- Cost Factors
- IAW
- Scorpion
- RACE
- Eagle

Fit for Purpose

2012 Atlanta
IT Optimization Assessment Methodology Example

- Project
  - Architectural Analysis
  - Candidate Solutions
    - Requirements Analysis
      - Requirements Sorted by Requirements-Fit
        - TCO Analysis
          - Optimized Project

Topics

- Code/Data
- Containers
- Connectors
- Platforms
- Requirements
- Local Factors
- Products
- Technology
- Capacity Plan
- Configurations
- Costs
End of Section
Architectural Analysis
Architectural Analysis

It’s About the Options!
Being “Platform Inclusive”
IT Solution Composition Components

- Components
  - Code
  - Data
  - Rules/Scripts/Config Files/etc.

  - What languages “can” be used?
  - What languages “must” be used?
  - What data structures “can” be used?
  - What data structures “must” be used?
IT Solution Composition Containers

- Components
  - Code
  - Data
  - Rules/Scripts/etc.

- Containers
  - Operating Systems
  - Middleware
  - Processes/Address Spaces
  - The “API” Provider
  - etc.

What containers “can” be used?

What containers “must” be used?
IT Solution Composition
Connections

• Components
  • Code
  • Data
  • Rules/Scripts/etc.

• Containers
  • Operating System
  • Middleware
  • Processes/Address Spaces
  • The “API” Provider
  • etc.

• Connections
  • TCP/IP RPC
  • TCP/IP DRDA
  • TCP/IP MQ
  • TCP/IP IEDN RPC
  • etc.

What connectors “can” be used?

What connectors “must” be used?
IT Services Composition Platform

- **Components**
  - Code
  - Data
  - Files/etc.

- **Connections**
  - TCP/IP RPC
  - TCP/IP DRDA
  - TCP/IP MQ
  - etc.

- **Platform**
  - Discrete Server
    - and Operating System
  - Virtual Machine
    - and Operating System
    - and Hosting Server / Hypervisor

What platforms “can” be used?

What platforms “must” be used?
Architectural Options

- **Components**
  - Code
  - Data
  - Rules/Scripts/Config Files/etc.

- **Containers**
  - Operating Systems
  - Middleware
  - Processes/Address Spaces
  - The “API” Provider
  - etc.

- **Connections**
  - TCP/IP RPC
  - TCP/IP DRDA
  - TCP/IP MQ
  - etc.

- **Platform**
  - Discrete Server
    - and Operating System
  - Virtual Machine
    - and Operating System
    - and Hosting Server / Hypervisor
## Topology Variable Overhead

<table>
<thead>
<tr>
<th>Server/Image 1</th>
<th>Server/Image 2</th>
<th>Server/Image 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Presentation Layer</strong>&lt;br&gt;Server Utilization</td>
<td><strong>Application Layer</strong>&lt;br&gt;Server Utilization</td>
<td><strong>Database Layer</strong>&lt;br&gt;Server Utilization</td>
</tr>
<tr>
<td>Application Processing Utilization</td>
<td>Application Processing Utilization</td>
<td>Application Processing Utilization</td>
</tr>
</tbody>
</table>

**Connectors Matter!**
All this processing is added by the topology.

And then there's the response time budget.

Added capacity adds cost.

Topics
A topology using co-location reduces processing.

And Then There’s the Response Time Budget.
Architecture Analysis - Options

• Define the option:
  • Components
  • Containers
  • Connectors
  • Platforms
  • Lifecycle Support (dev/test/qa/prod environments)

• **Limit** the options analysis to those that are truly viable
  • Based upon your “patterns”
  • Based upon your “edge” (leading / bleeding / trailing)
  • Boiling the ocean is not very productive

• **Document the Architectural Options (Architectural Artifacts)**
End of Section
Requirements Analysis
Requirements Analysis

Candidate Solutions

Requirements Analysis

Solutions Sorted by Requirements-Fit

- Requirements
- Local Factors
- Products
- Technology

Topics
Requirements (examples)

- Perform a function (capability)
- Deliver the function on time
- Perform that function where needed
- Perform that function when needed
  - planned up time (and planned down time)
  - unplanned down time (availability)
- Perform that function how needed
  - in a manner the user finds productive (ease of use)
  - in a manner the user finds productive (response time)
  - For all the users using the system (throughput)
- Make changes to the function over time
- Protect the function from illicit or illegal access or use (security)
- Provide ROI (deliver on budget)
Solution Viewpoints
1. Build Components

Development

- Code
- Code
- Data
Solution Viewpoints
2. Build Containers and Connectors

- Business Application Owner
- Business Analyst (Modeler)
- Developer
- Application Architect
- System Architect

Development

- Code Container
  - Connector
  - Code Container
    - Connector
    - Data Container
Solution Viewpoints
3. Build Platforms

- Business Application Owner
- Business Analyst (Modeler)
- Developer
- Application Architect
- System Architect
- Platform Engineer
- Platform
- Code Container
- Connector
- Platform
- Code Container
- Connector
- Platform
- Data Container
Solution Viewpoints
4. Build Test Environment
Solution Viewpoints
5. Build Production Environment

Topics

Business Application Owner
Business Analyst (Modeler)
Developer
Application Architect
System Architect
Platform Engineer
Development Manager
Tester
Operator
Users
Production Control
Call Center
Audit
Security

Development
Platform
Connector
Code
Container
Platform
Data
Container
Production
Platform
Connector
Code
Container
Platform
Data
Container
Test
Platform
Connector
Code
Container
Platform
Data
Container

Atlanta 2012
Solution Viewpoints

• Users - The Business
  • Business Process Owner / Application Owner / End User
  • Finance

• Architects
  • Patterns / Standards / Connectors

• Developers
  • Components / Languages / Data / Data Structures

• Engineers - Platform and Infrastructure Engineers
  • Container Providers
    • Middleware and Operating Systems
  • Platform Providers
    • (Virtual) Servers and (Virtual) Storage
  • Connector Providers
    • (Virtual) Networks and Middleware

• Testers
  • Functional Test
  • Usability Test
  • Acceptance Test
  • Load Test
  • Quality Assurance Test

• Operators - Operations
  • Operators / Help Desk / Automation
  • Production Control
  • Problem / Performance Management
  • Change / Configuration Management

• Auditors
  • Logs
  • Security
  • Data and privacy protection

Each “viewpoint” defines “OPTIMAL” in their own terms
So What is Optimal?

- Balance ALL viewpoints:
  - Requirements-based decision-making
  - Consensus building “argument”
  - Transparent requirements-based tradeoffs

- And decide:
  - How to build **components**
  - What **containers** to use
  - What **connectors** to use
  - **What platform to use**
    - For each container
    - For each lifecycle stage
Product/Technology Matching Requirements

Product processing characteristics

- Virtualization Efficiency
- Platform Tuned Software
- Threading and/or Superscalar
- Cache, Interconnect & Memory Model
- Deployment Model
- Processor Off-load
- High Memory Bandwidth
- Out of Order Execution

Workload processing characteristics

- Virtualization Efficiency
- Platform Tuned Software
- Threading and/or Superscalar
- Cache, Interconnect & Memory Model
- Deployment Model
- Processor Off-load
- High Memory Bandwidth
- Out of Order Execution

Match the “job” with the “tool”
Requirements Analysis

- IBM Design Centers
  - Proven Track Record

- Structured Requirements Analysis
  - aka Structured Argument

1. Define Requirements
   - coming from all viewpoints
2. Prioritize Requirements
3. Assess Each Solution Option’s Ability to Meet Requirements

**Create a Requirements Analysis Scorecard**
End of Section
Cost Analysis
Cost Analysis

Solutions Sorted by Requirements-Fit

TCO Analysis

Optimized Project

Capacity Plan
Configurations
Costs

Topics
Cost Analysis
“Total Cost of Ownership”

Categories
- Servers
- Software
- Floorspace
- Energy
- Connectivity
- Storage
- Facilities
- Sys.Admin
- Net.Admin
- Migration
- Downtime

Stages
- Build
- Operate
- Maintain
- Retire

Lifecycle

Tiers
- Data
- Application
- Integration
- Presentation

Topics
- DR
- HA
- Prod
- QA
- Test
- Dev

TCO

Presentation
Integration
Application
Data

Servers
Software
Floorspace
Energy
Connectivity
Storage
Facilities
Sys.Admin
Net.Admin
Migration
Downtime

Atlanta 2012
Costs

- **Building Costs**
  - The cost of building (or buying) & implementing *components*
  - The cost of building (or buying) & implementing *containers*
  - The cost of building (or buying) & implementing *connectors*
  - The cost of building (or buying) & implementing *platforms*

- **Operating Costs**
  - The cost of operating components, containers, connectors, and platforms
  - Including managing, monitoring, energizing, cleaning, and replenishing consumables

- **Maintaining Costs**
  - The cost of changing components, containers, connectors, and platforms
  - Including building changes, testing changes, and implementing changes
Cost Analysis - Prerequisites

- Prereq 1 – Technical Equity
  - Each option being analyzed must be sized and configured to meet the same set of requirements

- Prereq 2 – Solution Lifecycle Equity
  - Each option being analyzed must similarly meet the solution’s lifecycle stages and timetable

- Prereq 3 – Financial Equity
  - Each option being analyzed must receive similar financial analysis treatment
Cost Analysis – Tools

- IBM TCO Modeling Tools
  - RACEv for distributed server (including Linux) costing
  - RACEzOS for z/OS hosted workloads

- Use RACEv and RACEzOS together to model “hybrid solutions”
  - aka Multi-Tier Multi-Platform configurations
    - e.g. zEnterprise Heterogeneous Computing

- **Create a TCO Scorecard**
IT Optimization Assessment Methodology
Putting it All Together

Topics

- Atlanta 2012

**Architectural Analysis**
- Project Plans
- Architectural Options / Artifacts
- Requirements Analysis
- Requirements Scorecard
- TCO Analysis
- TCO Scorecard

**Requirements**
- Local Factors
- Products
- Technology

**Costs**
- Code/Data
- Containers
- Connectors
- Platforms
- Requirements
- Local Factors
- Products
- Technology
- Capacity Plan
- Configurations
- Costs
Case Study
The New Business Service (New Smart Meter Customer Application)

• Industry: Electric Utility
  • Smart Grid adopter (Smart Meter provider)
  • Collecting customer meter data on 15 min intervals

• New customer service
  • Home energy usage alerts
  • When energy usage “over budget”, owner gets notified
  • When energy usage “off pattern”, owner gets notified
  • Security use case: “Away From Home”
    • If “away”, and if energy usage pattern is amiss
    • Then owner (and optionally additional parties) get notified

• Web 2.0 UI (lite-browser)
Business Requirements

• Function: Home and Away-from-home energy-usage alerts
• Time Line: Production offering coincident with stockholders meeting
• Available to all home owners equipped with smart meters
  • From any browser and/or from downloaded mobile app
  • Large (growing) number of users – solution scalability required
• 7x24x365 Uptime
  • High availability runtime with DR (RTO=30min RPO=5min)
• Homeowner userid / pin (customer profile) access control
  • Assured data privacy high priority
• Integration with existing customer management system
  • z/OS – Sysplex – CICSpix – CICS TS - Web Services (or EXCl)
• Integration with existing smart meter readings database
  • z/OS – Sysplex – DB2 for z/OS – Remote SQL (jdbc or similar)
End of Section
Case Study – Architectural Analysis
Application Architecture

New Systems
- Platform
  - HTML
  - Apache
- Connector
- Platform
  - Java & JSPs
  - WAS-ND
- Connector
- Platform
  - Pattern Tables
  - DB2

Existing Systems
- Platform
  - Customer Sys
  - COBOL
  - CICS
- Platform
  - Readings Tables
  - DB2

Topics

DEV
TEST
PROD
Option 1 (x86)

New Systems
- Platform: x86, VMware, Virt. Mach., Windows
  - HTML
  - Apache
  - Network Connector

Existing Systems
- Platform: z/OS
  - Customer Sys
  - COBOL
  - CICS
  - Readings Tables
  - DB2
  - Network Connector

- Platform: x86, VMware, Virt. Mach., Windows
  - Java & JSPs
  - WAS-ND
  - Network Connector

- Platform: x86, VMware, Virt. Mach., Windows
  - Pattern Tables
  - DB2
  - Network Connector

Topics
Option 2 (z)

New Systems
- Platform: P7 PowerVM LPAR AIX
- HTML Apache
- IEDN Connector
- Java& JSPs WAS-ND
- PC Connector
- Pattern Mart
- Pattern Tables DB2
- IDAA Pattern Mart
- Pattern Tables

Existing Systems
- Customer Sys COBOL CICS
- Readings Tables
- Platform z/OS
- PC Connector
- DEV TEST PROD
Options 3 4 5 6 7 8 and …

- There are certainly many more options
  - z/VM zLinux
  - POWER PowerVM AIX
  - zBX POWER AIX
  - zEnterprise z/VM zLinux
  - etc.

- In a “real” Fit for Purpose effort, all viable options would be examined … but care need be taken not to “boil the ocean”

- But for this talk (for this case study) we will keep things simple
  - And just look at Options 1 and 2
End of Section
Case Study – Requirements Analysis
Requirements Analysis – Step 1

• Focus on requirements
• Respect for viewpoints
• Seek consensus
  • When consensus not reached
    • Understanding differences and distances
    • Sensitivity analysis (do differences matter?)

• Step 1 - List the requirements
# Requirements List

<table>
<thead>
<tr>
<th>Requirements List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data privacy</td>
</tr>
<tr>
<td>Scalability</td>
</tr>
<tr>
<td>Integration with existing customer management system</td>
</tr>
<tr>
<td>Development lifecycle support</td>
</tr>
<tr>
<td>Availability</td>
</tr>
<tr>
<td>Performance</td>
</tr>
<tr>
<td>Manageability</td>
</tr>
<tr>
<td>Integration with existing smart meter readings database</td>
</tr>
</tbody>
</table>
Sort and Weight the Requirements

- Step 2
  - Prioritize (i.e. sort) the requirements in the list
  - Assign a weight to each requirement
    - To indicate relative importance of each requirement

<table>
<thead>
<tr>
<th>Rank</th>
<th>Requirements List</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Performance</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Scalability</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Integration with existing customer management system</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>Integration with existing smart meter readings database</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Availability</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>Data privacy</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>Manageability</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>Development lifecycle support</td>
<td>3</td>
</tr>
</tbody>
</table>
Score the Options

• Step 3 – for each requirement, score the options
  • Assess each options ability to meet the requirement

<table>
<thead>
<tr>
<th>Qualifier</th>
<th>Label</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demonstrated Can Exceed</td>
<td>D.Exceed</td>
<td>7</td>
</tr>
<tr>
<td>Likely to Exceed</td>
<td>L.Exceed</td>
<td>6</td>
</tr>
<tr>
<td>Demonstrated Can Meet</td>
<td>D.Meet</td>
<td>5</td>
</tr>
<tr>
<td>Likely to Meet</td>
<td>L.Meet</td>
<td>4</td>
</tr>
<tr>
<td>Manageability</td>
<td>Marginal</td>
<td>2</td>
</tr>
<tr>
<td>Does Not Meet</td>
<td>No.Meet</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rank</th>
<th>Requirements List</th>
<th>Weight</th>
<th>Option 1 (x86) Score</th>
<th>Option 2 (z) Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Performance</td>
<td>6</td>
<td>L.Meet</td>
<td>L.Exceed</td>
</tr>
<tr>
<td>2</td>
<td>Scalability</td>
<td>5</td>
<td>L.Meet</td>
<td>L.Exceed</td>
</tr>
<tr>
<td>3</td>
<td>Integration with existing customer management system</td>
<td>4</td>
<td>L.Meet</td>
<td>D.Meet</td>
</tr>
<tr>
<td>4</td>
<td>Integration with existing smart meter readings database</td>
<td>4</td>
<td>L.Meet</td>
<td>D.Meet</td>
</tr>
<tr>
<td>5</td>
<td>Availability</td>
<td>4</td>
<td>Marginal</td>
<td>D.Meet</td>
</tr>
<tr>
<td>6</td>
<td>Data privacy</td>
<td>3</td>
<td>L.Meet</td>
<td>D.Meet</td>
</tr>
<tr>
<td>7</td>
<td>Manageability</td>
<td>3</td>
<td>L.Meet</td>
<td>D.Meet</td>
</tr>
<tr>
<td>8</td>
<td>Development lifecycle support</td>
<td>3</td>
<td>D.Meet</td>
<td>L.Meet</td>
</tr>
</tbody>
</table>
Rate the Options

- Step 4
  - Multiply the requirement’s weight by the option’s score
  - Add up the results

<table>
<thead>
<tr>
<th>Rank</th>
<th>Requirements List</th>
<th>Weight</th>
<th>Option 1 (x86)</th>
<th>Option 2 (z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Performance</td>
<td>6</td>
<td>L.Meet 24</td>
<td>L.Exceed 30</td>
</tr>
<tr>
<td>2</td>
<td>Scalability</td>
<td>5</td>
<td>L.Meet 20</td>
<td>L.Exceed 25</td>
</tr>
<tr>
<td>3</td>
<td>Integration with existing customer management system</td>
<td>4</td>
<td>L.Meet 16</td>
<td>D.Meet 20</td>
</tr>
<tr>
<td>4</td>
<td>Integration with existing smart meter readings database</td>
<td>4</td>
<td>L.Meet 16</td>
<td>D.Meet 20</td>
</tr>
<tr>
<td>5</td>
<td>Availability</td>
<td>4</td>
<td>Marginal 8</td>
<td>D.Meet 20</td>
</tr>
<tr>
<td>6</td>
<td>Data privacy</td>
<td>3</td>
<td>L.Meet 12</td>
<td>D.Meet 15</td>
</tr>
<tr>
<td>7</td>
<td>Manageability</td>
<td>3</td>
<td>L.Meet 12</td>
<td>D.Meet 15</td>
</tr>
<tr>
<td>8</td>
<td>Development lifecycle support</td>
<td>3</td>
<td>D.Meet 15</td>
<td>L.Meet 12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Result</td>
<td>Result</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>123</td>
<td>157</td>
</tr>
</tbody>
</table>

This is a “Requirements Scorecard”
Case Study – Cost Analysis
Step 1 – RACEv Subject Servers

- The set of discrete x86 servers
  - Presentation – Prod/Test/Dev
  - Application – Prod/Test/Dev
  - Database – Prod/Test/Dev
Step 2 – RACEv x86 Server Target

- The set of x86 VMware virtual server hosting blade servers
  - Presentation – Prod/Test/Dev
  - Application – Prod/Test/Dev
  - Database – Prod/Test/Dev
Step 3 – RACEv zEnterprise Target Distributed Elements

- The set of distributed elements in the zEnterprise solution
  - Presentation – Prod/Dev/Test
    - PS701s in the zBX
  - Database – Prod/Dev/Test
    - ISAO blades in the zBX
Step 4 – RACEzOS zEnterprise Elements

• The z/OS Elements of the solution
  • Application Servers
  • WAS for z/OS
  • PROD LPAR
  • TEST/DEV LPAR

• WAS/DB2 co-location
• zAAP on zIIP engines
z196 Host = 2817-M15/700 with 6 CPs: GP=4 zAAP=1 zIIP=1
6 Active Partitions: GP=2 zAAP=2 zIIP=2
Capacity basis: 2094-751 @ 602.00 MIPS for a single partition configuration
z196 and z10 processor capacity for z/OS is represented with HiperDispatch turned ON

<table>
<thead>
<tr>
<th>Include</th>
<th>No.</th>
<th>Type</th>
<th>Name</th>
<th>SCP</th>
<th>Workload</th>
<th>Mode</th>
<th>LCPs</th>
<th>Weight</th>
<th>Weight %</th>
<th>Capping</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>GP</td>
<td>Prod</td>
<td>z/OS-1.11</td>
<td>Average</td>
<td>SHR</td>
<td>4</td>
<td>80</td>
<td>80.00%</td>
<td></td>
<td>3,657</td>
<td>4,571</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>GP</td>
<td>Dev</td>
<td>z/OS-1.11</td>
<td>Average</td>
<td>SHR</td>
<td>2</td>
<td>20</td>
<td>20.00%</td>
<td></td>
<td>900</td>
<td>2,251</td>
</tr>
<tr>
<td>*1</td>
<td>zAAP</td>
<td>Prod</td>
<td>z/OS-1.11</td>
<td>Average</td>
<td>SHR</td>
<td>1</td>
<td>80</td>
<td>80.00%</td>
<td></td>
<td></td>
<td>915</td>
<td>1,144</td>
</tr>
<tr>
<td>*2</td>
<td>zAAP</td>
<td>Dev</td>
<td>z/OS-1.11</td>
<td>Average</td>
<td>SHR</td>
<td>1</td>
<td>20</td>
<td>20.00%</td>
<td></td>
<td></td>
<td>236</td>
<td>1,178</td>
</tr>
<tr>
<td>*1</td>
<td>zIIP</td>
<td>Prod</td>
<td>z/OS-1.11</td>
<td>Average</td>
<td>SHR</td>
<td>1</td>
<td>80</td>
<td>80.00%</td>
<td></td>
<td></td>
<td>915</td>
<td>1,144</td>
</tr>
<tr>
<td>*2</td>
<td>zIIP</td>
<td>Dev</td>
<td>z/OS-1.11</td>
<td>Average</td>
<td>SHR</td>
<td>1</td>
<td>20</td>
<td>20.00%</td>
<td></td>
<td></td>
<td>236</td>
<td>1,178</td>
</tr>
</tbody>
</table>

For significant configuration changes, capacity comparisons should be considered to have a +/-5% margin-of-error.
Upgrading the processor family is considered a significant configuration change.

Input fields have white background; Single-click a "selection field" for drop-down list; Double-click a "key-in field" to open.

Topics

SHARE
Atlanta
2012
Step 6 – Complete the Models
Examine the Results

RACEzOS/RACEv Cumulative Case Comparison

- (0)x86-discrete
- (1)x86-virtual
- zEnterprise

Year 1 | Year 2 | Year 3 | Year 4 | Year 5

0 | 1,000,000 | 2,000,000 | 3,000,000 | 4,000,000 | 5,000,000 | 6,000,000 | 7,000,000
Step 7 – Examine More Results

Cost Breakdown

- (0)x86-discrete
- (1)x86-virtual
- zEnterprise

Legend:
- Downtime
- DR
- SW Maint
- HW Maint
- People
- Facilities
- Software
- Hardware

SHARE
Technology • Connections • Results

SHARE
Atlanta 2012
Step 8 - Iterate

- Refine Inputs
- Add Additional Cases and Solution Configurations
- Sensitivity Analysis

- Assumptions Analysis
  - The cost & value of zEnterprise Unified Resource Manager
  - The value of co-location
  - The cross-server sizing
  - Discounting
  - Admin ratios
  - etc.
  - etc.
  - etc.

In other words…. Have a productive argument! …
Which is what ALL of this is ALL about!!!
And Finally

- Merge the Requirements Analysis...
- And the TCO Analysis

<table>
<thead>
<tr>
<th>Rank</th>
<th>Requirements List</th>
<th>Weight</th>
<th>Option 1 (x86)</th>
<th>Option 2 (z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Performance</td>
<td>6</td>
<td>L.Meet 24</td>
<td>L.Exceed 30</td>
</tr>
<tr>
<td>2</td>
<td>Scalability</td>
<td>5</td>
<td>L.Meet 20</td>
<td>L.Exceed 25</td>
</tr>
<tr>
<td>3</td>
<td>Integration with existing customer management system</td>
<td>4</td>
<td>L.Meet 16</td>
<td>D.Meet 20</td>
</tr>
<tr>
<td>4</td>
<td>Integration with existing smart meter readings database</td>
<td>4</td>
<td>L.Meet 16</td>
<td>D.Meet 20</td>
</tr>
<tr>
<td>5</td>
<td>Availability</td>
<td>4</td>
<td>Marginal 8</td>
<td>D.Meet 20</td>
</tr>
<tr>
<td>6</td>
<td>Data privacy</td>
<td>3</td>
<td>L.Meet 12</td>
<td>D.Meet 15</td>
</tr>
<tr>
<td>7</td>
<td>Manageability</td>
<td>3</td>
<td>L.Meet 12</td>
<td>D.Meet 15</td>
</tr>
<tr>
<td>8</td>
<td>Development lifecycle support</td>
<td>3</td>
<td>D.Meet 15</td>
<td>L.Meet 12</td>
</tr>
</tbody>
</table>

And make an “optimal” decision!
End of Section
Conclusion
IBM IT Optimization Analysis

Start

Technology Exploration

Issues Exploration

Local Patterns

Other Patterns

Platform Options

Requirements

Local Costing

zEnterprise

Whiteboard Right

Fit Workshop

Integration

Architecture

Workshop

Fit for Purpose

Workshop

IT Optimization

Scorecard

Requirements

Scorecard

TCO

Scorecard

TCO Workshop

(RACE/Eagle/Scorpion)
End of Section
Questions?
Comments?
Critiques?

Requests??!??!!

Monte Bauman
IBM Columbus

mbauman@us.ibm.com

Thank You