Architectural Implementation Analysis
A Comparative Methodology

Monday, March 02, 2015, 04:30PM-05:30PM
Sheraton Seattle, Greenwood

Monte Bauman
mbauman@us.ibm.com
IBM Columbus
Enterprise Server Technical Support
Presentation Plan

• The Challenge

• An Approach – IBM IT Optimization

• Architecture Analysis with RACEa

• Conclusion – I need your help
The Challenge
Where “can” it run?

Where “could” it run?

Where “should” it run?

Fit for Purpose
Platform Positioning

Workloads

Platforms
- Converged x86
- AWS
- SoftLayer
- System z
- Power
- Pure

The Challenge
Run the right work on the right platform … optimizing cost time risk and capability

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
The Challenge
Making Everyone Happy

Architects
Developers
Engineers
Technicians
Call Center
Business Analyst
Application Owner

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
The Challenge
Making Everyone Happy…
Not Only Now … But Over Time …

Build the System
Operate the System
Maintain and Enhance the System
Retire the System

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
The Challenge
Decisions Considering All the Right Things…

- Build the System
- Operate the System
- Maintain and Enhance the System
- Retire the System
- Time (the project plan)
- Quality of Service (risk aversion)
- Capability (functionality)
- Cost (the project budget)

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
“Optimally” Executing All These Workloads is a Challenge…
No single platform can do it all (maybe that’s why we offer more than one!)

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Workloads

Some workloads thrive on thread quality
Some workloads thrive on thread quantity
Some workloads thrive on memory quality
Some workloads thrive on memory quantity
Some workloads thrive on I/O quality
Some workloads thrive on I/O quantity
Some workloads thrive on integration quality
Some workloads thrive on integration quantity

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Platforms

- Some workloads thrive on thread quality
- Some workloads thrive on thread *quantity*
- Some workloads thrive on memory quality
- Some workloads thrive on memory *quantity*
- Some workloads thrive on I/O quality
- Some workloads thrive on I/O *quantity*
- Some workloads thrive on integration quality
- Some workloads thrive on integration *quantity*

Platforms provide capabilities to workloads.
Choices Choices Choices Choices
Making the Right Choice is...
(A) Hard
(B) Necessary
(C) Time Consuming
(D) Fit for Purpose
(E) IT Optimization
(F) A function of familiarity
(G) Essential
(H) Important

Workloads make demands on hardware

Platforms provide capabilities to workloads

IDAA, System z, Enterprise Power, Power Scaleout, PureFlex, PureData, x86 Converged Systems, Public Cloud, x86 Commodity Servers

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
The Penalties of IT Un-Optimization

- **Too Many Servers**
  - Too many cores
  - Too much software
  - Too much energy/floorspace

- **Too Much Unplanned Downtime**
  - Too many hardware failures
  - Too much time spent upgrading
  - Too much time spent patching

- **Too Much Labor**
  - Too much time spent re-deploying
  - Too much time spent re-provisioning
  - Too much time spent doing post-production changes

- **Too Much Network Reliance**
  - Too many network outages
  - Too much response time spent hopping through the network

- **Too Many Security Issues**
  - Too many missed audits
  - Too many compliance failures
  - Too many security breaches

<table>
<thead>
<tr>
<th>Time</th>
<th>Capability</th>
<th>Risk</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
An Approach – IBM IT Optimization
IT Optimization Assessment Methodology

Three Stages

Architecture Analysis

Requirements Analysis

Cost Analysis

Options

Local Factors

Cost Factors

Why we need to build it this way…

What to build…

How to build it…

What it’s going to cost overall … now and forever …
IT Optimization Assessment Methodology
Three Stages

Architecture Analysis

- Functional Requirements
  - Logical Architecture
  - Code and Data
  - Containers
  - Platforms (Clusters)
  - Connections (Connectors)

Sorted list of implementation options based upon logical architecture
IT Optimization Assessment Methodology

Three Stages

**IAW**
- IBM Infrastructure Architecture Workshop
- 1 or 2 day workshop
- IBM architects + app architects + enterprise architects + engineers
- Articulate and sort options

**RACEa**
- Comparative Architecture Analysis Tool
- Technical merit scoring
- Complexity scoring
- Bill of material scoring

**Architecture Analysis**

**Requirements Analysis**
- Local Factors
- Cost Factors

**Cost Analysis**

More On RACEa Soon
IT Optimization Assessment Methodology

Three Stages

Requirements Analysis

- Non Functional Requirements
  - Quality of Service
  - Throughput and Scale
  - Resilience and DR
  - Security and Audit
  - Skills and Investments

Sorted list of implementation options based upon requirements fulfillment scoring
IT Optimization Assessment Methodology
Three Stages

**Fit for Purpose Workshop**
- 1 or 2 day workshop
- Project (or pattern) scope
  - IBM moderated
- Architects + developers + engineers + IBM SMEs
  - Structured debate
  - Tool facilitated scoring
  - Options requirements-fitness based sorting

**RACEf**
- Platform requirements analysis and filtering tool
- Once calibrated, creates customized enterprise platform positioning tool
IT Optimization Assessment Methodology
Three Stages

Architecture Analysis

Requirements Analysis

Cost Analysis

Options

Local Factors

Cost Factors

Cost Analysis

- Total cost of ownership (TCO)
- Complete Bill of Materials
- Hardware Software Storage Networks
  - Labor Facilities DR
  - Full Lifecycle
  - Build Run Manage Retire

Sorted list of implementation options based upon TCO scoring

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
IT Optimization Assessment Methodology
Three Stages

Architecture Analysis
- Options

Requirements Analysis
- Local Factors
- Cost Factors

Cost Analysis
- Eagle
- Scorpion
- RACEv

RACEv
- Right-fitting applications into consolidated environments
  - Spreadsheet-based tool
  - Technical analysis
  - Cost analysis
  - TCO Scorecard

Scorpion
- IBM Global Business Service
- Consulting Engagement
- CIO/budget & down analysis
  - Report for the CIO

Eagle
- Consulting Engagement
  - Bottom-up technical and TCO analysis
  - TCO Scorecard
IT Optimization Assessment Methodology
Three Stages

Architecture Analysis

Requirements Analysis

Cost Analysis

Fit for Purpose (F4P) Workshop

Local Factors

Cost Factors

Options

IAW

RACEa

RACEf

Eagle

Scorpion

RACEv

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Workload Placement IT Optimization Process

Platform List

Candidate Platform 1

Candidate Platform 2

Candidate Platform N

Functional Requirements

Architectural Analysis

Non-Functional Requirements Analysis

Total Cost of Ownership Analysis

Candidate Platform 1

Candidate Platform N

Non-Functional Requirements Analysis

F4P

RACEv

RACEa

RACEf

Candidate Platform 1

Candidate Platform N

Target Platform 1

Candidate Platform N

Programming Model

Compute Model

Containers

Quality of Service

IT Strategy

Investments

Time & Budget

Workload Runtime Characteristics

Capacity Plan

Configuration

Unit Costs

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Architecture Analysis with RACEa
RACEa - Step by Step

1. Understand RACEa’s architecture taxonomy
2. Document project description
3. Describe project attributes
4. Calibrate scoring attributes and tables
5. Setup architecture component relationships
6. Describe architecture implementation one
7. Describe architecture implementation two (three/four)
8. Review output reports
9. Implement the optimal architecture implementation
RACEa’s Architecture Taxonomy

Taxonomy

How RACEa flexibly and simply describes the components of an architecture implementation

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Taxonomy: A Node in the Architecture...

**Node**

- **Code** node contains code
- **Data** node contains data
- **Originator** node contains a device or sensor or other “internet of things” thing
- **Container** “holds” the code or data
- Usually middleware like WAS or DB2 or Apache

- **Platform** “holds” the container
- Usually a combo of hardware and hypervisor (optional) and operating system

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Taxonomy: Clusters and Connectors

Connections
- Container interactions
- How nodes connect
- How fast / how distant
- How much data flows and how often

Clusters
- Type of cluster
  - Code cluster
- Container cluster (like RAC)
- Platform cluster (like Sysplex)
- Type of cluster connection
  - Local Area Network
  - Coupling Links / Coupling Facility

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Taxonomy: “Production” and ...

Lifecycle Stages

- Unit Test
- Integration Test
- Stress Test
- QA Test
- Production
- DR

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Taxonomy Elements

- **Originator**
  - Mobile device, internet of things thing, browser, etc.
- **Code**
  - HTML, Java, COBOL, config-files, etc.
- **Data**
  - Rows (tables), records (files), streams, etc.
- **Container**
  - Middleware
- **Platform**
  - Server, hypervisor (optional), and operating system (typically)
- **Inbound and Outbound Connector**
  - Payload and Inter-Node Invocation Frequency
- **Inbound and Outbound Connection**
  - Type and Distance
- **Cluster Type**
  - Code cluster
  - Container cluster
  - Platform cluster
  - Inter-Cluster Communication
Project Attributes

- Project description
  - Nature of project / purpose of analysis
- Custom one-off project
  - In which case automation (provisioning/orchestration) is not important
- Pattern-based highly replicated project
  - In which case automation is essential
- Something in between
  - In which case automation is important, but not essential
- Ample opportunities exist for localizing the tool’s merit and complexity scoring system in simple weighted scoring tables
Scoring Attributes and Tables

- Element Ownership
- Element Disposition
- Element Provisioning Source
- Element Deployment Technique
- Quality of Service Confidence

<table>
<thead>
<tr>
<th>Assessment Attributes</th>
<th>Merit Score Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Project Type</td>
</tr>
<tr>
<td></td>
<td>Highly Custom One-</td>
</tr>
<tr>
<td></td>
<td>of-a-Kind Project</td>
</tr>
<tr>
<td></td>
<td>Project with both</td>
</tr>
<tr>
<td></td>
<td>pattern based and</td>
</tr>
<tr>
<td></td>
<td>one of a kind</td>
</tr>
<tr>
<td></td>
<td>componentry</td>
</tr>
<tr>
<td></td>
<td>Highly automatable</td>
</tr>
<tr>
<td></td>
<td>highly replicated</td>
</tr>
<tr>
<td></td>
<td>pattern-based</td>
</tr>
<tr>
<td></td>
<td>project</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Element Ownership</td>
<td>4</td>
</tr>
<tr>
<td>Element Disposition</td>
<td>7</td>
</tr>
<tr>
<td>Element Provisioning Source</td>
<td>5</td>
</tr>
<tr>
<td>Element Deployment Technique</td>
<td>4</td>
</tr>
<tr>
<td>Quality of Service Confidence</td>
<td>10</td>
</tr>
</tbody>
</table>
Scoring Attributes – Element Ownership

- Element Ownership
  - Corporate
  - Partner
  - Customer
  - Vendor/Supplier

<table>
<thead>
<tr>
<th>Assessment Attribute 1: Element Ownership</th>
<th>Code</th>
<th>Data</th>
<th>Conta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Partner</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Customer</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Vendor/Supplier</td>
<td>7</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
Scoring Attributes – Element Disposition

- Element Disposition
  - New
  - Extended
  - Existing / Shared

<table>
<thead>
<tr>
<th>Assessment Attribute 2: Element Disposition</th>
<th>Code</th>
<th>Data</th>
<th>Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Extended</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Existing / Shared</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>n/a</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Scoring Attributes – Element Provisioning Source

- Element Provisioning Source
  - Whitespace-Pool
  - Upgrade-Pool
  - New
  - Provisioning Not Required

<table>
<thead>
<tr>
<th>Assessment Attribute 3: Element Provisioning Source</th>
<th>Code</th>
<th>Data</th>
<th>Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whitespace-Pool</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Upgrade-Pool</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>New</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Provisioning Not Required</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Scoring Attributes – Element Deployment Technique

- Element Deployment Technique
  - Custom
  - Pattern-Based
  - Orchestrated-Pattern
  - Deployment Action Not Required

<table>
<thead>
<tr>
<th>Assessment Attribute 4: Element Deployment Technique</th>
<th>Code</th>
<th>Data</th>
<th>Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Pattern-Based</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Orchestrated-Pattern</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Deployment Action Not Required</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Scoring Attributes – QoS Confidence

- Quality of Service Confidence
  - High Confidence
  - Medium Confidence
  - Low Confidence
  - No Confidence

<table>
<thead>
<tr>
<th>Assessment Attribute 5: Quality of Service Confidence</th>
<th>Code</th>
<th>Data</th>
<th>Container</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Confidence</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Medium Confidence</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Low Confidence</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
Architecture Component Relationships

- **Customizable Configuration Tables**
  - Use Case 1 – describe what your enterprise supports
    - For application development lifecycle support
  - Use Case 2 – describe what is possible
    - For enterprise architecture development

- **Setup valid relationships between elements**
  - Populate drop-down selection lists
  - The containers that can hold code
    - e.g. “WAS-ND” can hold “Java”
  - The containers that can hold data
    - e.g. “DB2” can hold “Row” (or “Table”, if you prefer)
  - The platforms that can hold containers
    - e.g. “z/OS on zEC12” can hold “DB2”
  - The connectors that connect containers
    - e.g. “WAS-ND” supports “JCA”
  - The connections that support connectors
    - e.g. “JCA” can flow over “Local_LAN”

<table>
<thead>
<tr>
<th>Code, Data, or Originator Element</th>
<th>Candidate Containers…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>WAS-ND</td>
</tr>
<tr>
<td>COBOL</td>
<td>CICS</td>
</tr>
<tr>
<td></td>
<td>Liberty</td>
</tr>
<tr>
<td></td>
<td>CICS</td>
</tr>
<tr>
<td></td>
<td>JES</td>
</tr>
<tr>
<td></td>
<td>IMS</td>
</tr>
<tr>
<td></td>
<td>Tomcat</td>
</tr>
<tr>
<td></td>
<td>DB2-SP</td>
</tr>
</tbody>
</table>

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Implementation One …

• Map the logical architecture to a physical architecture
• For each node:
  – Choose code (or data) to use
  – Choose containers to use
  – Choose platforms to use
  – Choose connectors to use
  – Choose connections to use
  – Choose clustering and cluster connections to use
  – For each element … pick assessment attribute4s
  – Got another node? … add one! (add as many as you like!!)
• Describe the physical architecture for production
  – Describe for test, QA, etc. (by adding more and more nodes)
• Concentrate on the things that vary between options
• Build the 1st … then up to 3 more (total of 4) … one per sheet
## Implementation One …

<table>
<thead>
<tr>
<th>Tier Label</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description (Optional)</td>
<td>WAS Layer</td>
<td>CICS Layer</td>
</tr>
<tr>
<td>Application Development Lifecycle Stage</td>
<td>WAS</td>
<td>CICSPlex</td>
</tr>
<tr>
<td>Tier Quality of Service Requirement</td>
<td>Production</td>
<td>Production</td>
</tr>
<tr>
<td>Cluster Type</td>
<td>High Quality of Service</td>
<td>High Quality of Service</td>
</tr>
<tr>
<td>Number of Nodes in Cluster</td>
<td>Not a Cluster</td>
<td>Container Cluster</td>
</tr>
<tr>
<td>Intra-Cluster Communication</td>
<td>None</td>
<td>System z Coupling Facility</td>
</tr>
<tr>
<td>Tier Type (Originator/Code/Data)</td>
<td>Code</td>
<td>Code</td>
</tr>
<tr>
<td>Originator/Code/Data Selection</td>
<td>Java</td>
<td>COBOL</td>
</tr>
<tr>
<td>Container Selection</td>
<td>WAS-ND</td>
<td>CICS</td>
</tr>
<tr>
<td>Platform Selection</td>
<td>Windows-VM-x86</td>
<td>z/OS</td>
</tr>
<tr>
<td>Inbound Connector Selection</td>
<td>JCA</td>
<td>Local LAN</td>
</tr>
<tr>
<td>Inbound Connection Selection</td>
<td></td>
<td>Small</td>
</tr>
<tr>
<td>Inbound Average Payload Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outbound Connector Selection</td>
<td>JCA</td>
<td></td>
</tr>
<tr>
<td>Outbound Connection Selection</td>
<td>Local LAN</td>
<td></td>
</tr>
<tr>
<td>Outbound Average Payload Size</td>
<td>Small</td>
<td></td>
</tr>
<tr>
<td>Average Processing Load Estimate</td>
<td></td>
<td>Medium</td>
</tr>
<tr>
<td>Average Invocations of This Tier Per Transaction (Per Execution)</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
Analysis Output Report

- After two or more architecture implementations are described you can compare them on the “ScoreCard”

- ScoreCard Elements:
  - Relative Overall Merit Score
  - Relative Complexity Score
  - Relative Processing Burden Score
  - Relative Networking Burden Score
  - Relative Bill of Materials Report
Overall Merit Scoring

• Blended (weighted) score
  – Based upon each component’s:
    • Ownership selection
    • Disposition selection
    • Provisioning Source selection
    • Deployment Technique selection
    • Quality of Service Confidence selection
• Provides “relative” indicator of merit
  – Indeed … all “scores” in this tool are “relative”
    • And only of value used within the tool to compare options
## Complexity Scoring

### Assessment Attribute 2: Element Disposition

<table>
<thead>
<tr>
<th>Element Disposition</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>10</td>
</tr>
<tr>
<td>Extended</td>
<td>6</td>
</tr>
<tr>
<td>Existing / Shared</td>
<td>4</td>
</tr>
<tr>
<td>n/a</td>
<td>0</td>
</tr>
</tbody>
</table>

### Assessment Attribute 3: Element Provisioning Source

<table>
<thead>
<tr>
<th>Provisioning Source</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whit space-Pool</td>
<td>2</td>
</tr>
<tr>
<td>Upgrade-Pool</td>
<td>6</td>
</tr>
<tr>
<td>New</td>
<td>10</td>
</tr>
<tr>
<td>Provisioning Not Required</td>
<td>0</td>
</tr>
</tbody>
</table>

### Assessment Attribute 4: Element Deployment Technique

<table>
<thead>
<tr>
<th>Deployment Technique</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom</td>
<td>10</td>
</tr>
<tr>
<td>Pattern-Based</td>
<td>7</td>
</tr>
<tr>
<td>Orchestrated-Pattern</td>
<td>4</td>
</tr>
<tr>
<td>Deployment Action Not Required</td>
<td>0</td>
</tr>
</tbody>
</table>
# Processing Burden Scoring

<table>
<thead>
<tr>
<th>Processing Load Table</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown</td>
<td>0</td>
</tr>
<tr>
<td>Light</td>
<td>1</td>
</tr>
<tr>
<td>Medium-Light</td>
<td>10</td>
</tr>
<tr>
<td>Medium</td>
<td>100</td>
</tr>
<tr>
<td>Medium-Heavy</td>
<td>1000</td>
</tr>
<tr>
<td>Heavy</td>
<td>10000</td>
</tr>
</tbody>
</table>

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
# Network Burden Scoring

- Calibrating the “cost” of networking
- Remembering … the best networking is NO networking

<table>
<thead>
<tr>
<th>Connections</th>
<th>Type</th>
<th>SubType</th>
<th>Distance</th>
<th>Parameter Passing by Value</th>
<th>Short Description</th>
<th>Score</th>
<th>Network Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Memory</td>
<td>Intra-Process Call/Return</td>
<td>n/a</td>
<td>Parameter Passing by Reference</td>
<td>Call_by_Reference</td>
<td>10</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Parameter Passing by Value</td>
<td>Call_by_Value</td>
<td>9.9</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inter-Process Call/Return</td>
<td>n/a</td>
<td>Parameter Passing by Value</td>
<td>Inter-Process_Call_by_Value</td>
<td>9.5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Message Queue Put/Get</td>
<td></td>
<td>Parameter Passing by Value</td>
<td>Local_Message_Queue</td>
<td>9</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td>LAN</td>
<td>Local_IP_Stack_Reflection</td>
<td>Parameter Passing by Value</td>
<td>Local_IP_Stack_Reflection</td>
<td>8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local_Virtual_LAN</td>
<td>Parameter Passing by Value</td>
<td>Local_Virtual_LAN</td>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMC-R_Local_LAN</td>
<td>Parameter Passing by Value</td>
<td>SMC-R_Local_LAN</td>
<td>7.5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local_Shared_NIC</td>
<td>Parameter Passing by Value</td>
<td>Local_Shared_NIC</td>
<td>6</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local_LAN</td>
<td>Parameter Passing by Value</td>
<td>Local_LAN</td>
<td>5</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Metro_LAN</td>
<td>Parameter Passing by Value</td>
<td>Metro_LAN</td>
<td>4</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAN</td>
<td>Metro_WAN</td>
<td>Parameter Passing by Value</td>
<td>Metro_WAN</td>
<td>4</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Regional_WAN</td>
<td>Parameter Passing by Value</td>
<td>Regional_WAN</td>
<td>3</td>
<td>1500</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>National_WAN</td>
<td>Parameter Passing by Value</td>
<td>National_WAN</td>
<td>2</td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>International_WAN</td>
<td>Parameter Passing by Value</td>
<td>International_WAN</td>
<td>1</td>
<td>2500</td>
<td></td>
</tr>
</tbody>
</table>
Network Burden Scoring (cont.)

<table>
<thead>
<tr>
<th>Size</th>
<th>Bytes</th>
<th>Kbytes</th>
<th>Mbytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Small</td>
<td>1024</td>
<td>1</td>
<td>0.00</td>
</tr>
<tr>
<td>Small</td>
<td>4096</td>
<td>4</td>
<td>0.00</td>
</tr>
<tr>
<td>Medium</td>
<td>16384</td>
<td>16</td>
<td>0.02</td>
</tr>
<tr>
<td>Large</td>
<td>262144</td>
<td>256</td>
<td>0.25</td>
</tr>
<tr>
<td>Very Large</td>
<td>2097152</td>
<td>2048</td>
<td>2</td>
</tr>
<tr>
<td>Huge</td>
<td>16777216</td>
<td>16384</td>
<td>16</td>
</tr>
</tbody>
</table>
Bill of Materials

• List of all elements composing the architecture
• What’s new list
• What’s extended list
• What’s reused (shared) list
## ScoreCard

### Architectural Analysis

#### Name

<table>
<thead>
<tr>
<th>Topology ONE</th>
<th>Topology TWO</th>
<th>Topology THREE</th>
<th>Topology FOUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>WinTel</td>
<td>zUnix</td>
<td>AIX</td>
<td>z/OS</td>
</tr>
</tbody>
</table>

#### Description

<table>
<thead>
<tr>
<th>WAS on Windows on x86</th>
<th>WAS on zUnix</th>
<th>WAS on AIX</th>
<th>WAS on z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>303</td>
<td>303</td>
<td>303</td>
<td>303</td>
</tr>
<tr>
<td>548</td>
<td>548</td>
<td>548</td>
<td>548</td>
</tr>
<tr>
<td>534</td>
<td>561</td>
<td>534</td>
<td>561</td>
</tr>
<tr>
<td>534</td>
<td>561</td>
<td>561</td>
<td>561</td>
</tr>
<tr>
<td>190</td>
<td>190</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>190</td>
<td>190</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>257</td>
<td>281</td>
<td>257</td>
<td>309</td>
</tr>
<tr>
<td>257</td>
<td>257</td>
<td>257</td>
<td>309</td>
</tr>
</tbody>
</table>

#### Total Architectural Score

<table>
<thead>
<tr>
<th></th>
<th>Topology ONE</th>
<th>Topology TWO</th>
<th>Topology THREE</th>
<th>Topology FOUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural Rank</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Normalized Architectural Score</td>
<td>0.947</td>
<td>0.973</td>
<td>0.956</td>
<td>1.000</td>
</tr>
</tbody>
</table>

### Complexity Analysis

#### Cluster Scoring

<table>
<thead>
<tr>
<th>Topology ONE</th>
<th>Topology TWO</th>
<th>Topology THREE</th>
<th>Topology FOUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>127</td>
<td>91</td>
<td>127</td>
<td>42</td>
</tr>
<tr>
<td>127</td>
<td>127</td>
<td>127</td>
<td>42</td>
</tr>
</tbody>
</table>

#### Total Complexity Score

<table>
<thead>
<tr>
<th></th>
<th>Topology ONE</th>
<th>Topology TWO</th>
<th>Topology THREE</th>
<th>Topology FOUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity Rank</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Normalized Complexity Score</td>
<td>0.188</td>
<td>0.176</td>
<td>0.188</td>
<td>0.131</td>
</tr>
</tbody>
</table>

---

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
### ScoreCard (cont.)

<table>
<thead>
<tr>
<th>Load Analysis</th>
<th>Topology ONE</th>
<th>Topology TWO</th>
<th>Topology THREE</th>
<th>Topology FOUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Networking Load</td>
<td>1638400</td>
<td>16384</td>
<td>1638400</td>
<td>0</td>
</tr>
<tr>
<td>Processing Load</td>
<td>410</td>
<td>410</td>
<td>410</td>
<td>410</td>
</tr>
<tr>
<td>Total Complexity Score</td>
<td>1638810</td>
<td>16794</td>
<td>1638810</td>
<td>410</td>
</tr>
<tr>
<td>Complexity Rank</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Normalized Complexity Score</td>
<td>551.602</td>
<td>5.653</td>
<td>551.602</td>
<td>0.138</td>
</tr>
</tbody>
</table>

#### Bill of Materials Lists

<table>
<thead>
<tr>
<th>New</th>
<th>Topology ONE</th>
<th>Topology TWO</th>
<th>Topology THREE</th>
<th>Topology FOUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java + WAS-ND + Windows-VM-x86 + JCA</td>
<td>Java + WAS-ND + zLinux + JCA</td>
<td>Java + WAS-ND + AIX- Power-LPAR + JCA</td>
<td>Java + WAS-ND + zLinux + JCA</td>
<td></td>
</tr>
<tr>
<td>Unique New Elements Count</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Total New Elements Count</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

#### Extended

<table>
<thead>
<tr>
<th>Extended</th>
<th>Topology ONE</th>
<th>Topology TWO</th>
<th>Topology THREE</th>
<th>Topology FOUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>* + Local_LAN</td>
<td>* + Local_LAN</td>
<td>* + Local_LAN</td>
<td>* + Inter-Process_Call_by_V</td>
<td></td>
</tr>
<tr>
<td>Local_IP_Stack_Reflection</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique Extended Elements Count</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total Extended Elements Count</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

#### Existing / Shared

<table>
<thead>
<tr>
<th>Existing / Shared</th>
<th>* + COBOL + CICS + z/OS</th>
<th>* + CICS + z/OS</th>
<th>* + COBOL + CICS + z/OS</th>
<th>* + COBOL + CICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique Existing / Shared Elements Count</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Total Existing / Shared Elements Count</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Unique Number of Elements</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Total Number of Elements</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>
Conclusion
Keys to Success

- Look at the options (for projects (for patterns))
- Understand the options (what you have (what you could have))
- Pick the right option (For the right reason)
- Systematically
  - Adaptable
    - To technology as it evolves (change is constant)
    - To “local factors” as they evolve (change is constant)
  - Repeatable
  - Facilitate understanding, teaming, and learning

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Executing a RACEa Workshop

• Architecture Analysis - RACEa Workshop
  – Typically one-half to one day on-site tooling-facilitated no-charge workshop
  – With application architects and platform architects & others
    • (1) define logical system architecture
    • (2) define rendition 1 architecture implementation
    • (3) define rendition 2 (3/4) architecture implementation(s)
    • (4) calibrate scoring tables
    • (5) examine results, discuss, calibrate, loop
    • (6) finalize results and report

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Next Steps

• RACEg
  – TCO Analysis Tool
  – beta testers needed

• RACEf
  – Requirements-Based Platform Selection Tool
  – beta testers needed

• RACEa
  – Architecture Analysis Tool
  – beta testers needed
  – need calibration data / network & processing burden data

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
The Final Chart

• Any questions?
• Any suggestions?
• Any way I can be of service?

– Monte Bauman
– Enterprise Server Technical Support
– IBM Columbus
– mbaum@us.ibm.com

Thank You

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Network Latency Matters

**The Objective:** Determine the actual latency incurred when making off-platform calls

This study provides response time measurements for two simple TCP/IP configurations.

Two System z LPARs on the same zEC12 server share an OSA-Express adapter in the 1st measurement.

In the second, the same two LPARs use two different OSA cards connected to the same router (the LPARs are one network hop away)
Test Environment Comparison

Shared OSA Configuration

- z/TPF
- zLinux
- OSA

One-Hop Route Configuration

- z/TPF
- zLinux
- OSA
- IP Router

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Controlled Test Environment for Apples to Apples Comparison

Server Constants

- Same zEC12 processor was used for all tests
  - z/TPF LPAR with one dedicated CP
  - zLinux LPAR with one dedicated IFL
- Same driver was used in all tests
  - Same number of driver instances was run for each comparison test
- The only difference in a given comparison test was the network path used

Message Driver Input

- The number of driver instances to start
- The message size (This is the amount of user data in each request message and each response message)
- The delay factor (which is how long to wait after receiving a response before sending the next request message)
- Number of messages to send before the driver exits
Each Instance of the Driver Does What

• Starts a long running TCP socket

• Loops N times doing the following:
  – Save current time (T1)
  – Issue socket send() API to send request message of size X
  – Issue socket read() API to read the response message
  – Get current time (T2) and calculate round trip time (RTT) for this request/response message pair (T2-T1)
    and then adjust the average RTT

• **This is the RTT from the application perspective – this is not the network (TCP) RTT**
  – Sleep for a user specified amount of time

• Ends the socket
<table>
<thead>
<tr>
<th>Message Rate (Messages/Second)</th>
<th>Shared OSA RTT (microseconds)</th>
<th>1-Hop Route RTT (microseconds)</th>
<th>RTT Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,000</td>
<td>159</td>
<td>275</td>
<td>1.73</td>
</tr>
<tr>
<td>20,000</td>
<td>181</td>
<td>345</td>
<td>1.90</td>
</tr>
<tr>
<td>30,000</td>
<td>216</td>
<td>553</td>
<td>2.56</td>
</tr>
<tr>
<td>40,000</td>
<td>260</td>
<td>724</td>
<td>2.78</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message Rate (Messages/Second)</th>
<th>Shared OSA RTT (microseconds)</th>
<th>1-Hop Route RTT (microseconds)</th>
<th>RTT Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>5,000</td>
<td>154</td>
<td>439</td>
<td>2.85</td>
</tr>
<tr>
<td>10,000</td>
<td>161</td>
<td>443</td>
<td>2.75</td>
</tr>
<tr>
<td>15,000</td>
<td>170</td>
<td>448</td>
<td>2.63</td>
</tr>
<tr>
<td>20,000</td>
<td>181</td>
<td>455</td>
<td>2.51</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message Rate (Messages/Second)</th>
<th>Shared OSA RTT (microseconds)</th>
<th>1-Hop Route RTT (microseconds)</th>
<th>RTT Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>150</td>
<td>915</td>
<td>6.10</td>
</tr>
<tr>
<td>4000</td>
<td>153</td>
<td>920</td>
<td>6.01</td>
</tr>
<tr>
<td>6000</td>
<td>155</td>
<td>928</td>
<td>5.99</td>
</tr>
<tr>
<td>8000</td>
<td>159</td>
<td>934</td>
<td>5.87</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message Rate (Messages/Second)</th>
<th>Shared OSA RTT (microseconds)</th>
<th>1-Hop Route RTT (microseconds)</th>
<th>RTT Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>550</td>
<td>2120</td>
<td>3.85</td>
</tr>
<tr>
<td>1000</td>
<td>551</td>
<td>2120</td>
<td>3.85</td>
</tr>
<tr>
<td>1500</td>
<td>550</td>
<td>2130</td>
<td>3.87</td>
</tr>
<tr>
<td>2000</td>
<td>552</td>
<td>2394</td>
<td>4.33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message Rate (Messages/Second)</th>
<th>Shared OSA RTT (microseconds)</th>
<th>1-Hop Route RTT (microseconds)</th>
<th>RTT Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>401</td>
<td>2530</td>
<td>6.31</td>
</tr>
<tr>
<td>500</td>
<td>467</td>
<td>2547</td>
<td>5.45</td>
</tr>
<tr>
<td>750</td>
<td>390</td>
<td>2556</td>
<td>6.55</td>
</tr>
<tr>
<td>1000</td>
<td>390</td>
<td>2564</td>
<td>6.57</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message Rate (Messages/Second)</th>
<th>Shared OSA RTT (microseconds)</th>
<th>1-Hop Route RTT (microseconds)</th>
<th>RTT Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>362</td>
<td>4288</td>
<td>11.88</td>
</tr>
<tr>
<td>250</td>
<td>361</td>
<td>4344</td>
<td>12.00</td>
</tr>
<tr>
<td>375</td>
<td>367</td>
<td>4377</td>
<td>11.93</td>
</tr>
<tr>
<td>500</td>
<td>371</td>
<td>4390</td>
<td>11.83</td>
</tr>
</tbody>
</table>
## Summary (Shared OSA vs 1-Hop Route)

<table>
<thead>
<tr>
<th>Message Size</th>
<th>Average RTT Ratio</th>
<th>Average Extra Time Per Message (microseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2.2</td>
<td>270</td>
</tr>
<tr>
<td>500</td>
<td>2.7</td>
<td>280</td>
</tr>
<tr>
<td>1400</td>
<td>6.0</td>
<td>292</td>
</tr>
<tr>
<td>5000</td>
<td>4.0</td>
<td>1640</td>
</tr>
<tr>
<td>10,000</td>
<td>6.2</td>
<td>2137</td>
</tr>
<tr>
<td>20,000</td>
<td>11.9</td>
<td>3984</td>
</tr>
</tbody>
</table>