Architecting & Tuning IIB / eXtreme Scale for Maximum Performance and Reliability

Suganya Rane
Solution Architect

Prolifics

Agenda

Introduction

Challenge: The need for Speed & Scalability - WXS

- Extreme Scale Basics
- Use Cases

Performance Tuning - IIB

- Measurement - Bottlenecks
- Infrastructure

Maximizing Reliability and Reducing Downtime

References & What’s Next

Extreme Scale Caching & The Need for Speed

"As the world becomes more instrumented, interconnected and intelligent, Internet-based activities, online transactions and data volumes increase. Further, these increasing amounts of data, along with rising consumer expectations and the need to maintain a competitive edge require fast and reliable performance…. Elastic caching is the answer."

Challenge: The Need For Speed & Scalability

How scalable is your System?

- When the peak load doubles, so does response time:
- A) stay the same, B) double C) quadruple D) fail

Have you tested your system to destruction?

- How high can the load go before failure
- Where are the bottlenecks and single points of failure

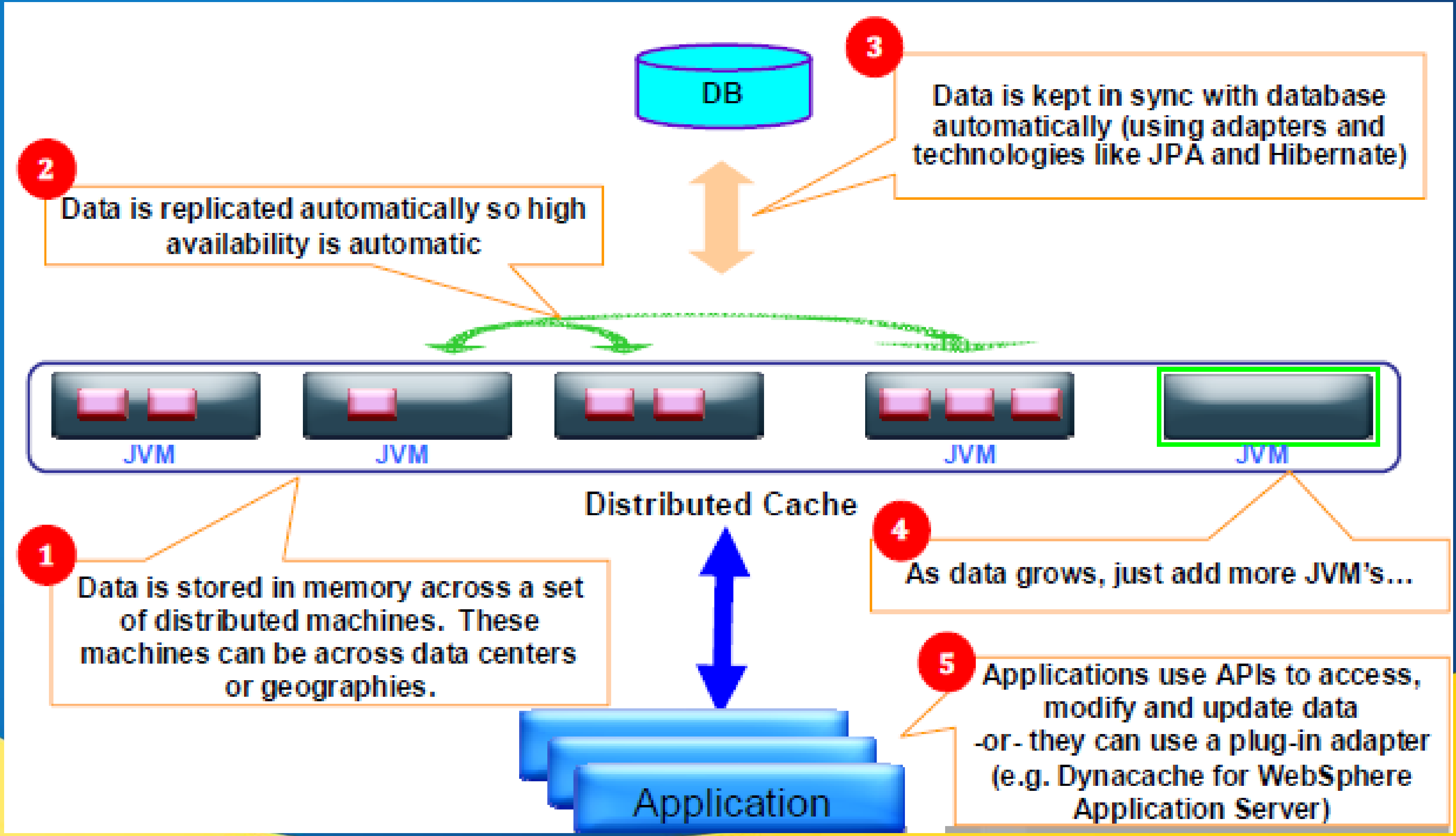
Focus on Extreme Scale

- Benefit - the approached taken for measurement, monitoring, bottleneck identification will work on many systems.

What is Extreme Scale?

- IBM WebSphere eXtreme Scale is an elastic, scalable, in-memory data grid (IMDG).
  - It is designed to help you handle exponential growth of transactions. With WebSphere eXtreme Scale, you can expect an enhanced quality of service in high-performance computing environments.
- WebSphere eXtreme Scale provides a full set of IMDG functions.
  - This approach provides a solution to scalability issues through caching and grid technology. The data grid dynamically caches, partitions, replicates and manages application data and business logic throughout multiple servers.
- WebSphere eXtreme Scale can perform massive volumes of transaction processing with high efficiency and linear scalability.
- WebSphere eXtreme Scale integrates with other middleware products such as IBM WebSphere Application Server, IBM WebSphere Commerce, IBM WebSphere Portal Server, and supports big data, mobile and cloud computing.

Extreme Scale Basics: How it Works



1 Data is stored in memory across a set of distributed machines. These machines can be across data centers or geographies.

2 Data is replicated automatically so high availability is automatic

3 Data is kept in sync with database automatically (using adapters and technologies like JPA and Hibernate)

4 As data grows, just add more JVM's…

5 Applications use APIs to access, modify and update data -or- they can use a plug-in adapter (e.g. Dynacache for WebSphere Application Server)

The Benefits

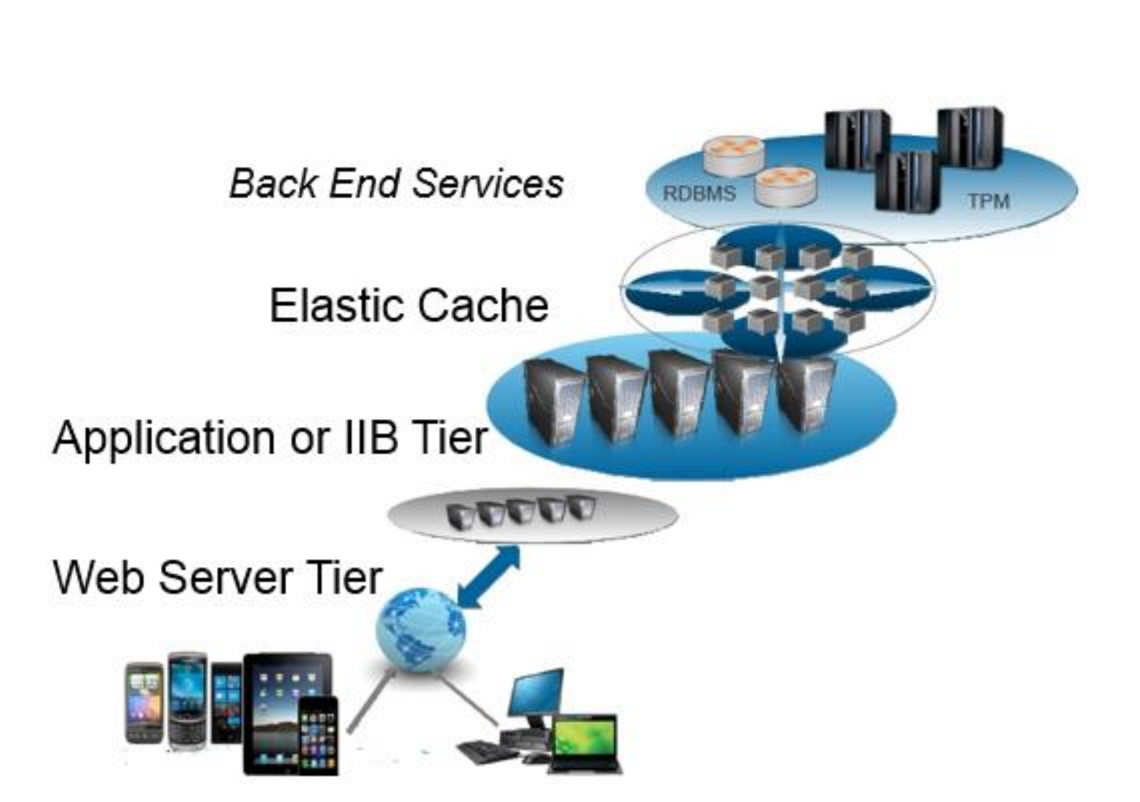
- Processes massive volumes of transactions with extreme efficiency and linear scalability.

- Rapidly builds a seamless, flexible, highly available elastic grid that scales out as applications scale, removing the performance limits of the database.

- Provides high availability and security with redundant copies of cache data, and authentication schemas that help ensure system security.

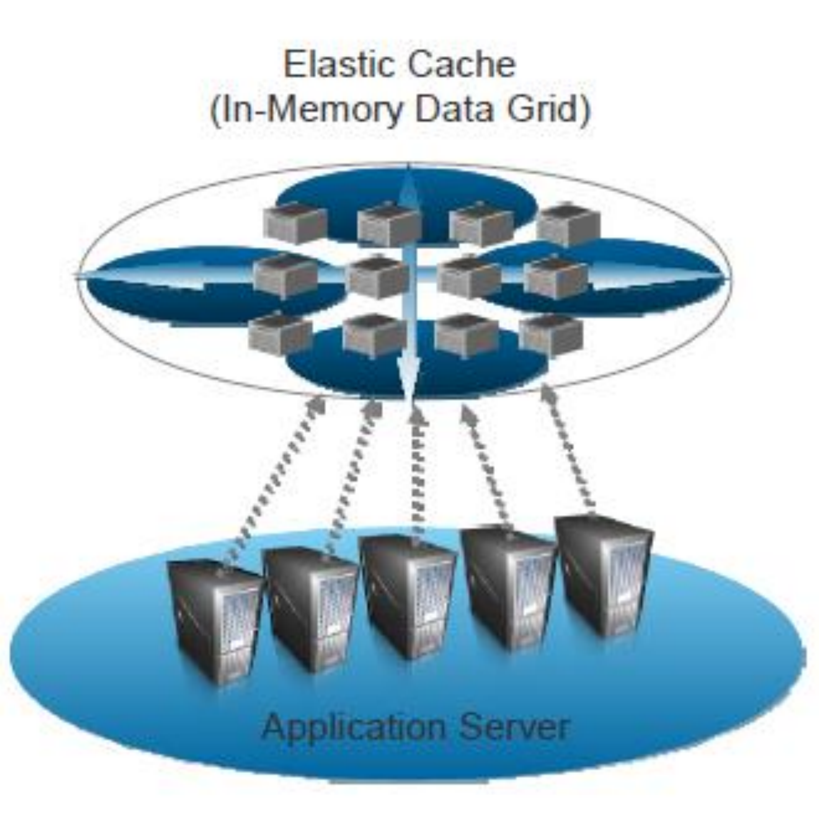
- Reduces total cost of ownership by enabling your existing backend systems to support significantly more applications.

Extreme Scale Basics: Architecture



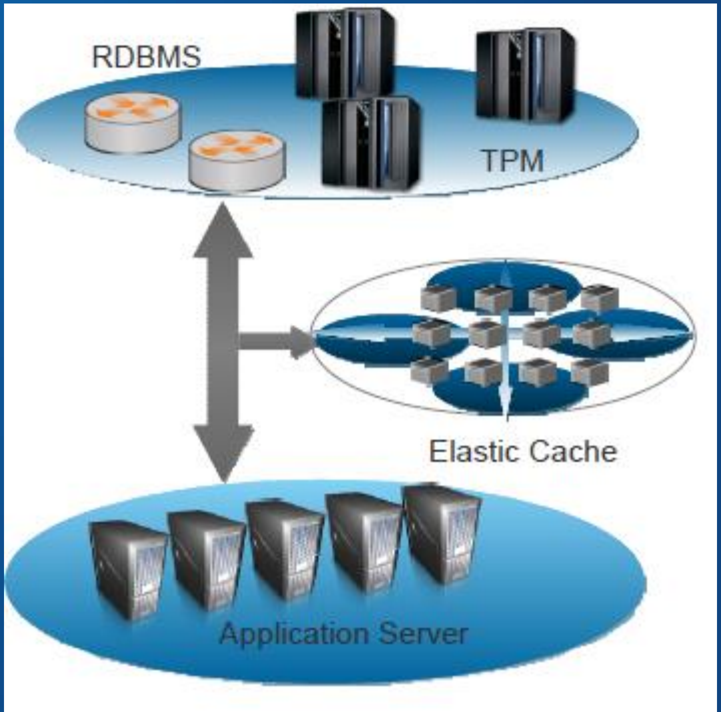
Use Case 1: Universal Application State

- Single replacement for multiple local caches
- Consistent response times
- Reduces Application Server JVM heap size
- Improved memory utilization - more memory for applications
- Faster Application Server start-up
- Removes invalidation chatter of local caches
- Applications move application state to grid
- Stateless applications scale elastically
- Application state can be shared across data centers for high availability



Use Case 2: Side Cache Pattern

- Client first checks the grid before using the data access layer to connect to a backend data store
- If an object is not returned from the grid (a cache “miss”), the client uses the data access layer as usual to retrieve the data
- The result is put into the grid to enable faster access the next time
- The back end remains the system of record, and usually only a small amount of the data is cached in the grid
- An object is stored only once in the cache, even if multiple clients use it
- Thus, more memory is available for caching, more data can be cached, which increases the cache hit rate
- Improve performance and offload unnecessary workload on backend systems



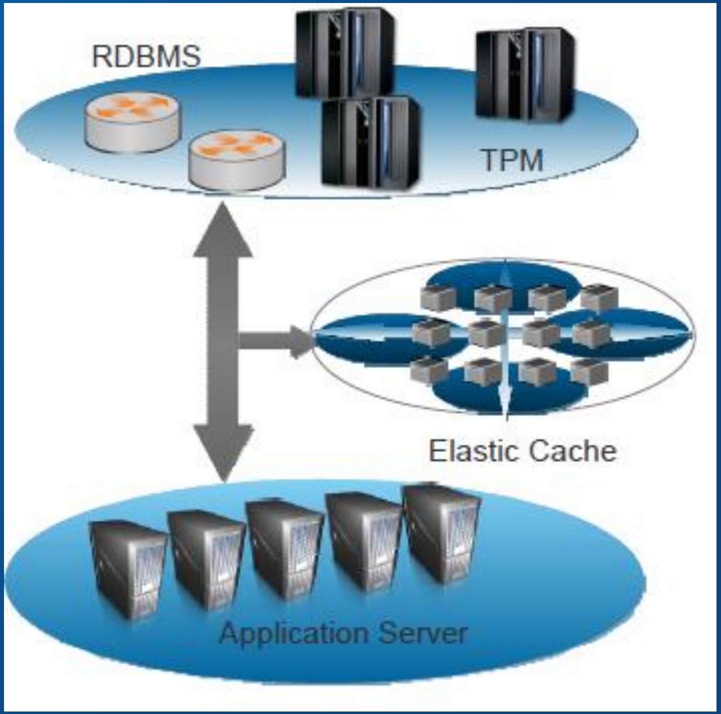
Use Case 2: ESB - Side Cache Pattern

Easily integrates into the existing business process

- No code changes to the client application or backend application
- Simply add the side cache mediation at the ESB layer

Significantly reduces the load on the back-end system by eliminating redundant requests

- Eliminates costly MIPS by eliminating redundant request
- Allows for more “REAL” work to be performed
- Improves overall response time
- Minimizes the need to scale hardware to increase processing capacity since the back-end system no longer has to handle redundant requests



Response time from elastic cache is in milliseconds

Performance Tuning & Measurement

Several useful tools include:

- For MQ performance - the MS0P Support Pac - allows us to see which Queues have buildups
- For IIB WebAdmin - interfaces with the Flow Statistic to identify nodes in a flow identifying slow (high elapse time), or inefficiency from a CPU usage point of view
- Captured CPU time for all servers, the TPS rate (end to end as well as per product) and by performing multiple runs, using the IBM Performance harness

Message Flow Tuning & Performance

- Ensure that message flows are an appropriate length.
- Use the minimum number of nodes required.
- Avoid having consecutive ESQL Compute nodes in a message flow with no other nodes between them.
  - Combine the logic into a single Compute node instead. Also avoid having consecutive JavaCompute nodes in the message flow.

Message Flow Tuning & Performance

- Minimize the volume of message parsing
- Choose wisely between Asynchronous and Synchronous calls.
  - If a message flow makes a synchronous call to an external application or service that can be slow or unpredictable in its response, it is more efficient to write the message flow using an asynchronous model.

Optimizing message flow throughput

- Multiple threads processing messages in a single message flow
- Additional Instances property of the deployed message flow in the BAR file
- Influence the order in which messages are processed by setting the Order Mode property of the MQInput node

Compare the performance of the message flows

- The Flow comparison view
  - Web User Interface to display snapshots accounting and statistics information

- The Flow Analysis view

Optimizing message flow throughput

- Multiple copies of the message flow in a broker
  - You can also deploy several copies of the same message flow to different integration servers in the same broker. This option has similar effects to increasing the number of processing threads in a single message flow.
  - Will only work for “Out of Order” processing.

- Copies of the message flow in multiple brokers
  - You can deploy several copies of the same message flow to different brokers. This option requires changes to your configuration, because you must ensure that applications that supply messages to the message flow can put their messages to the right input queue or port.

- The frequency of commits
  - Commit Count. This property represents the number of messages processed from the input queue per message flow thread, before an MQCMIT is issue
  - Commit Interval. This property represents the time interval that elapses before an MQCMIT is started

SupportPac MS0P: MQ Explorer plug-ins

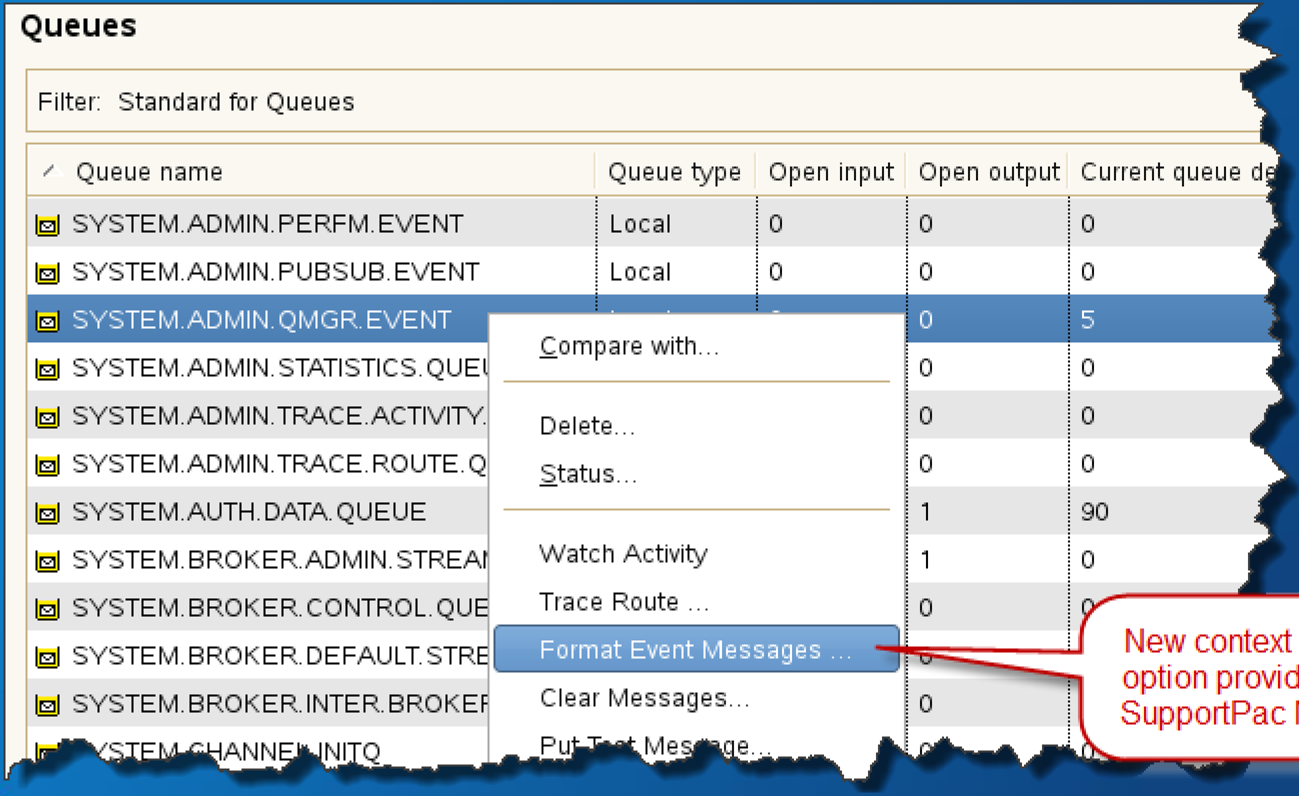
A variety of WMQ Explorer plug-ins are provided

- Format event messages into human-readable text
- Trace route function to determine the path messages take
- Export WMQ Explorer display panels to CSV
- Remote management of Windows, UNIX and Linux systems

Utilities

- Qtune
- MQIDecode turns codes into symbolic names
- OAM logging utility records PCF commands sent to CMDSVR

SupportPac MS0P: Event Message Context Menu



Queues

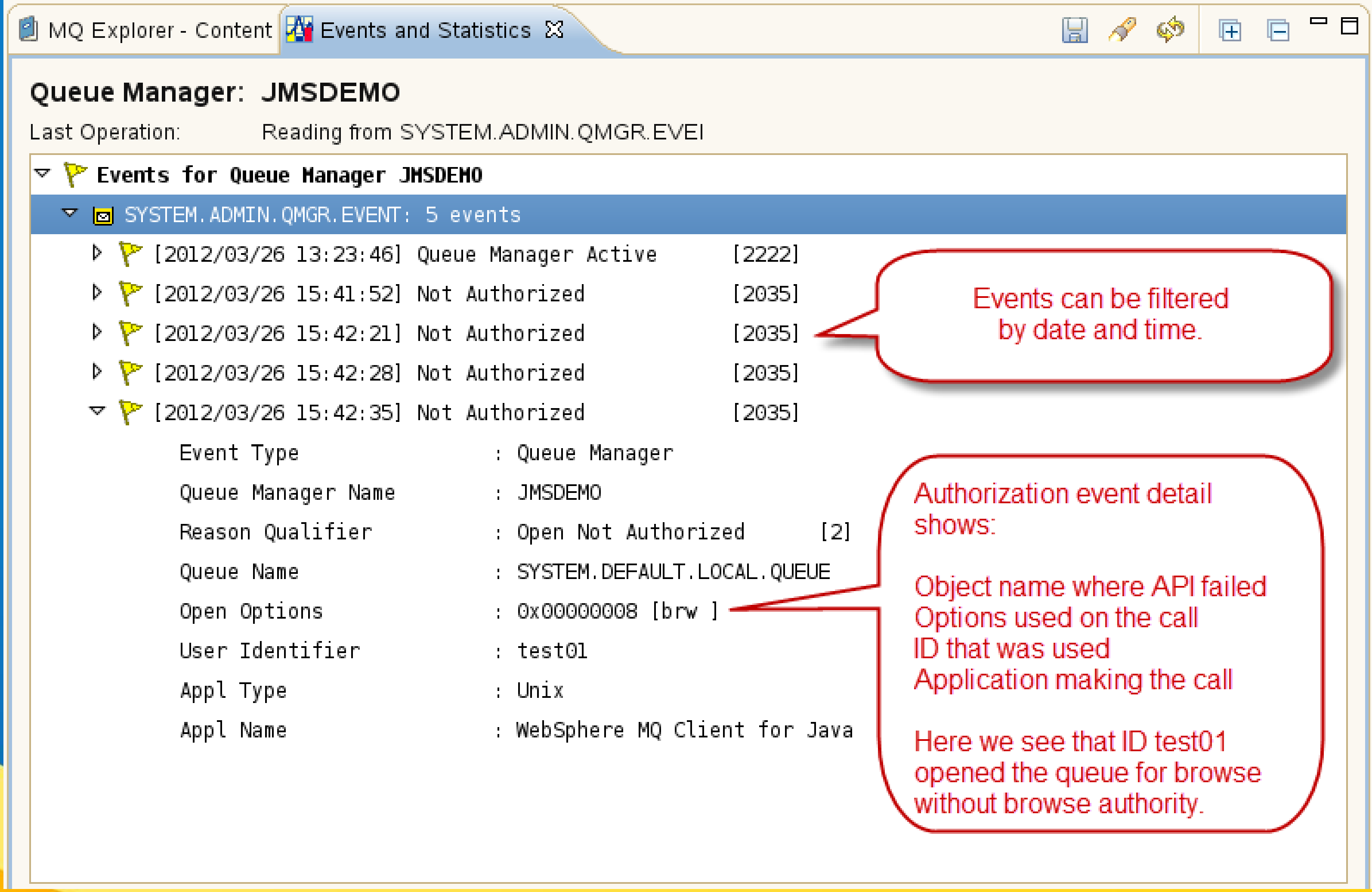
Filter: Standard for Queues

Queue name	Queue type	Open input	Open output	Current queue de
SYSTEM.ADMIN.PERFM.EVENT	Local	0	0	0
SYSTEM.ADMIN.PUBSUB.EVENT	Local	0	0	0
SYSTEM.ADMIN.QMGR.EVENT			0	5
SYSTEM.ADMIN.STATISTICS.QUEU			0	0
SYSTEM.ADMIN.TRACE.ACTIVITY.			0	0
SYSTEM.ADMIN.TRACE.ROUTE.Q			0	0
SYSTEM.AUTH.DATA.QUEUE			1	90
SYSTEM.BROKER.ADMIN.STREAI			1	0
SYSTEM.BROKER.CONTROL.QUE			0	0
SYSTEM.BROKER.DEFAULT.STRE			0	
SYSTEM.BROKER.INTER.BROKEF			0	
SYSTEM.CHANNEL.INITQ			0	0

New context menu option provided by SupportPac MS0P

After installing SupportPac MS0P, new context menu items appear in WebSphere MQ Explorer. The "Format Event Message" option shows up when right-clicking on an event queue.

SupportPac MS0P: GUI Display



Queue Manager: JMSDEMO

Last Operation: Reading from SYSTEM.ADMIN.QMGR.EVEI

```
Events for Queue Manager JMSDEMO
  SYSTEM.ADMIN.QMGR.EVENT: 5 events
    [2012/03/26 13:23:46] Queue Manager Active      [2222]
    [2012/03/26 15:41:52] Not Authorized            [2035]
    [2012/03/26 15:42:21] Not Authorized            [2035]
    [2012/03/26 15:42:28] Not Authorized            [2035]
    [2012/03/26 15:42:35] Not Authorized            [2035]
        Event Type                  : Queue Manager
        Queue Manager Name          : JMSDEMO
        Reason Qualifier            : Open Not Authorized       [2]
        Queue Name                  : SYSTEM.DEFAULT.LOCAL.QUEUE
        Open Options                : 0x00000008 [brw ]
        User Identifier             : test01
        Appl Type                   : Unix
        Appl Name                   : WebSphere MQ Client for Java
```

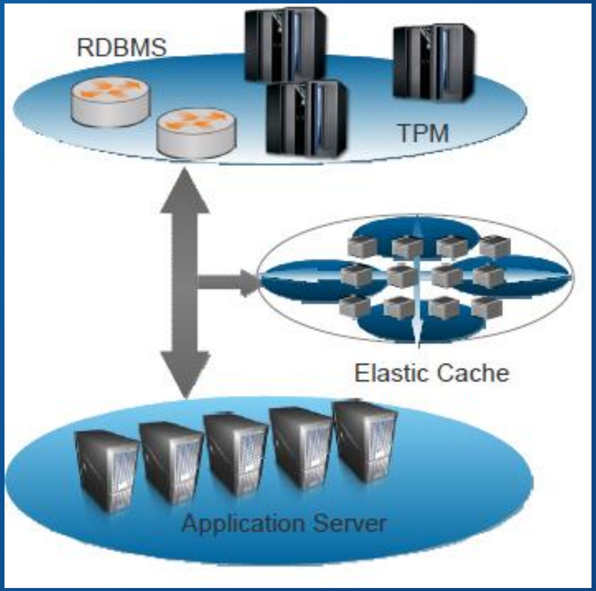
Events can be filtered by date and time.

Authorization event detail shows:

Object name where API failed
Options used on the call
ID that was used
Application making the call

Here we see that ID test01 opened the queue for browse without browse authority.

Infrastructure: Avoid Single Points of Failure



- Fast network: 10 gigabit +
- Multiple Nics
- Nearby Catalogs
- Fast Disk Drives: The new slid state drives (SSDs) are incredible.
- Trace route: On multiple occasions we have found very unexpected network delays. Measure early and often

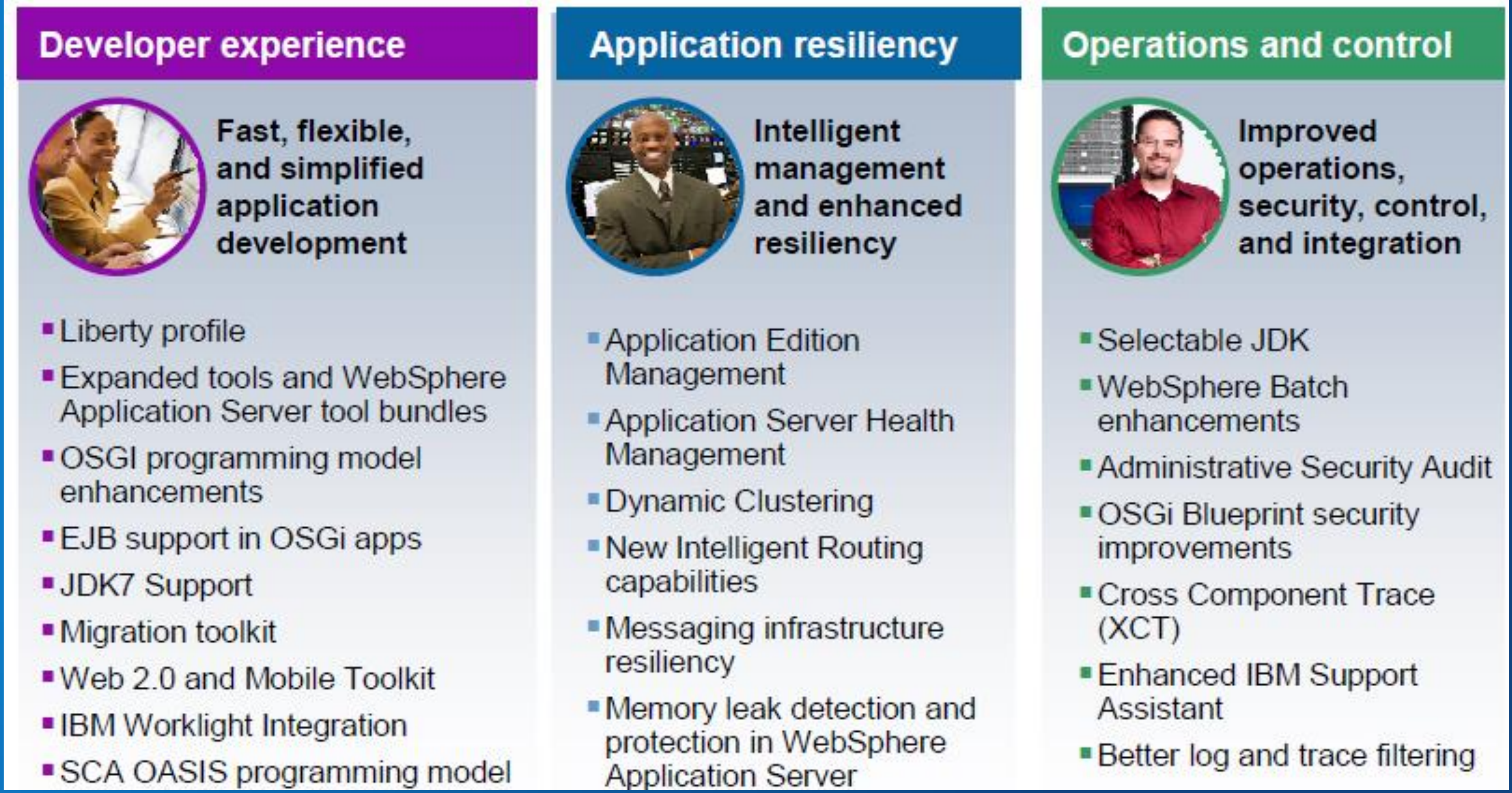
Prevention is Better than Cure

- WXS 8.6 makes it easier to recover from crashes.
- Log analysis was done exploring the cause of those crashes

The next few slides show how WAS 8.5 can

prevent some of those crashes.

WebSphere Application Server V8.5



Developer experience

Fast, flexible, and simplified application development

- Liberty profile
- Expanded tools and WebSphere Application Server tool bundles
- OSGI programming model enhancements
- EJB support in OSGi apps
- JDK7 Support
- Migration toolkit
- Web 2.0 and Mobile Toolkit
- IBM Worklight Integration
- SCA OASIS programming model

Application resiliency

Intelligent management and enhanced resiliency

- Application Edition Management
- Application Server Health Management
- Dynamic Clustering
- New Intelligent Routing capabilities
- Messaging infrastructure resiliency
- Memory leak detection and protection in WebSphere Application Server

Operations and control

Improved operations, security, control, and integration

- Selectable JDK
- WebSphere Batch enhancements
- Administrative Security Audit
- OSGi Blueprint security improvements
- Cross Component Trace (XCT)
- Enhanced IBM Support Assistant
- Better log and trace filtering

Three primary goals:

- **Provide intelligent management and enhanced resiliency for your application server environment**
- **Improve operation, security, control**
- **Integration of the application server -** providing a fast, flexible, and simplified application development environment that allows you to deliver rich user experience faster

Benefits of Extreme Scale running inside WAS 8.5

- Catalog and container servers as managed application servers
- Catalog service domain configuration
- Simplified connection to the grid
- Simplified grid container management
- Simplified administration and management
- Use WAS web console for help with monitoring

Monitoring helps find problems

Monitor for operational exceptions and to maintain health of the system:

- Expand the default size of error logs
- Check error logs for exceptions
- Archive error logs for forensic analysis
- Alert on messages in DLQ or error queue
- Distinguish between operational vs. Application alerts

FFSTSummary

Why learn to use the tools that summarize FFST and error files?

- “Because that’s where the errors are.”

FFST files contain useful information on serious problems

- http://hursleyonwmq.wordpress.com/2007/05/04/introduction-to-ffsts/

What is FFST?

- FFST stands for First Failure Support Technology, and is “designed to create detailed reports for IBM Service with information about the current state of a part of a queue manager together with historical data”

- FFSTSummary is a tool that produces one line summaries of these serious problems, arranged in chronological order

Lessons Learned

Install and Use the best tools for measurement

- Identify bottlenecks (flows, queries)

Use the best available hardware:

- Solid State Drives
- Fast network: 10 gigabit +
- Multiple Nics

Tune one thing at a time

It is a capital crime to theorize in advance of your data

Questions & Answers