TOC

INDEX

VIEW

F19 DB2 UDB and WebSphere Application Server at Light Speed

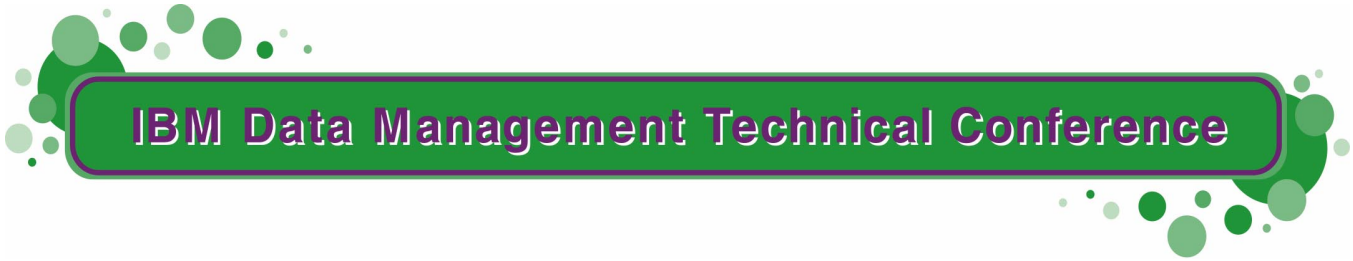
Quentin Presley, Staff Software Engineer, IBM Toronto Lab

This session will cover the latest tuning and performance techniques available to get the most out of your WebSphere Application Server and DB2 UDB system on the Windows and Unix platforms. Topics covered will include tuning the WebSphere Connection Pools, HTTP Session Persistence tuning through larger page sizes and multirow sessions, How Statement caching affects throughput to and from the Database, and Recent benchmark results. We will also cover some tuning tips for EEE systems, and some DB2 Connect tuning best practices.

F19

DB2 UDB and WebSphere Application Server at Light Speed

Quentin Presley



Anaheim, CA **Sept 9 - 13, 2002**

RETURN TO INDEX

Disclaimers

Notices, Trademarks, Service Marks and Disclaimers

This document contains proprietary information of IBM. The information contained in this publication does not include any product warranties, and any statements provided in this document should not be interpreted as such.

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries: IBM, DB2, DB2 Universal Database, WebSphere, WebSphere Application Server, OS/390, OS/400, AIX.

Windows and Windows-based trademarks and logos are trademarks or registered trademarks of Microsoft Corp.

Unix and Unix-based trademarks and logos are trademarks or registered trademarks of The Open Group.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc.

In the United States and other countries. Other company, product or service names may be the trademarks or service marks of others. The furnishing of this document does not imply giving licence to any IBM patents. References in this document to IBM products, Programs, or Services do not imply that IBM intends to make these available in all countries in which IBM operates.

Agenda

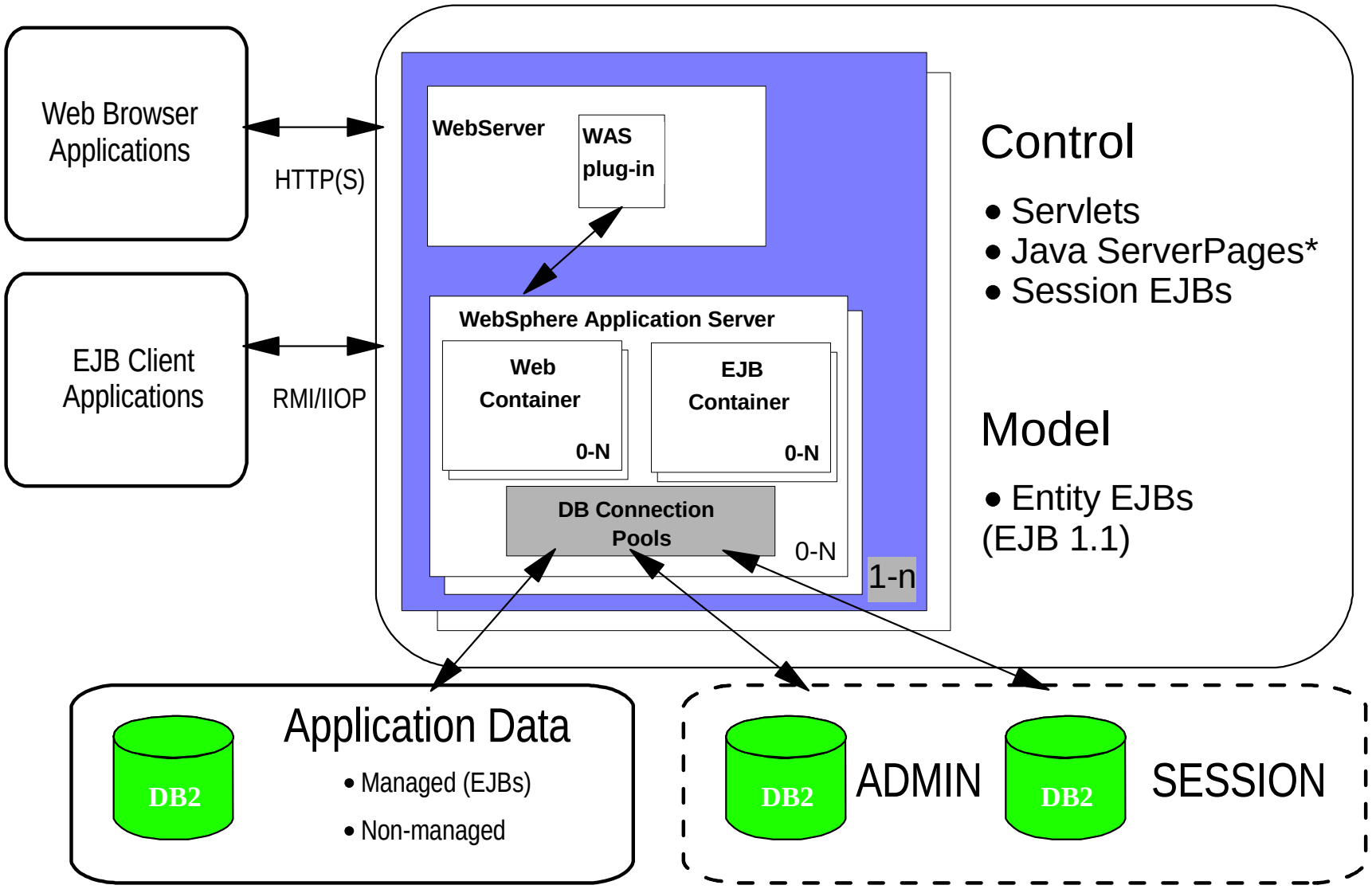
- How does WebSphere access DB2
- Configuring DB2 for performance
- Managing DB2 connections
- WebSphere PreparedStatement caching
- Persistent Session tips for DB2
- High availability considerations
- Benchmarks and Scalability
- Importance of the J2EE certification

Distributed Application Bottlenecks

- **Analyze CPU and I/O at each layer**

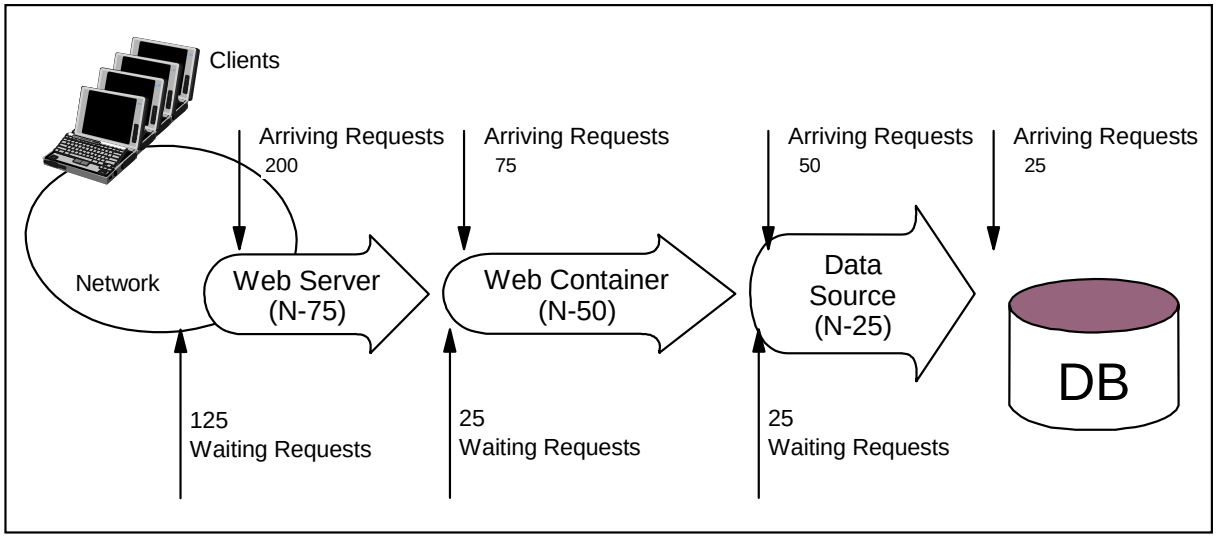
- Application Servers (WebSphere)
  - CPU
  - Garbage collection
  - Application synchronization code
  - Over allocated memory (paging, I/O)
- Database Servers (DB2)
  - Memory usage
  - Physical layout
    - Indexes
    - Clustering (Multi-dimensional clusters)
    - Tablespace layout
  - Access plan
    - Plan selections/Statement caching
    - Optimization
  - Concurrency
- Network traffic between layers (J2EE programming model)

WebSphere's Usage of DB2 UDB



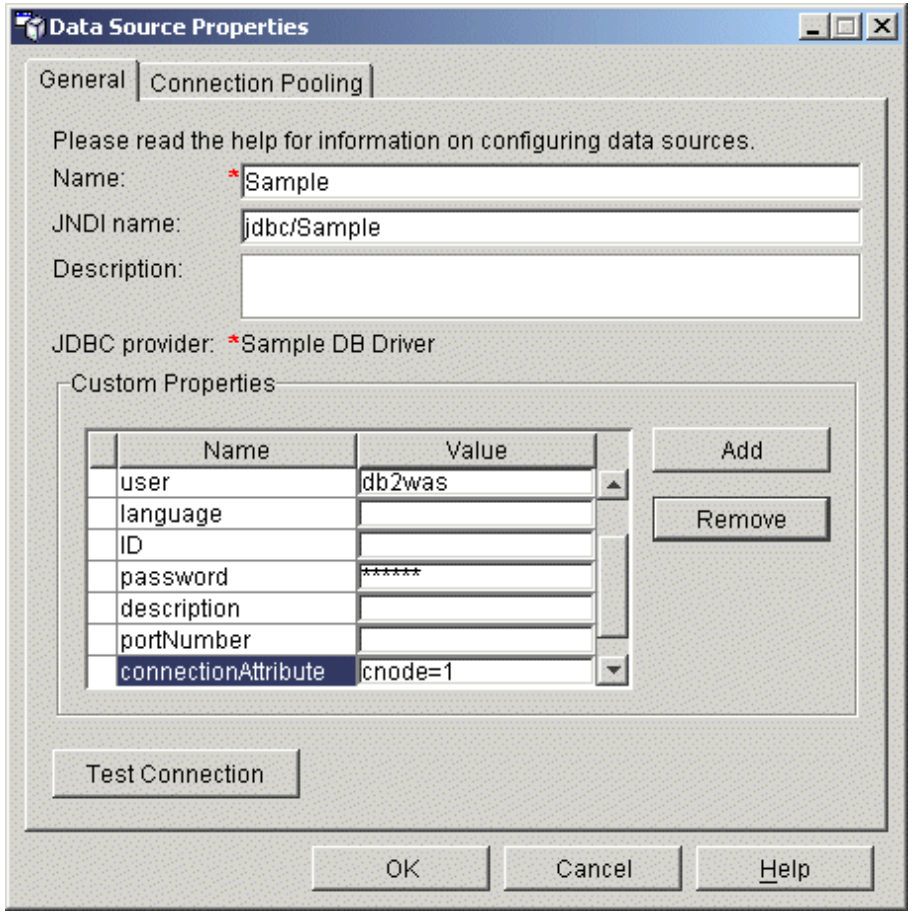
Queuing

- Queues are used in each layer of a WAS environment
- Set data sources according to the maximum queue sizes for Web containers
- DB2 does not use queues as all requests are synchronous
- Aggressive queue sizes can lead to resource saturation.



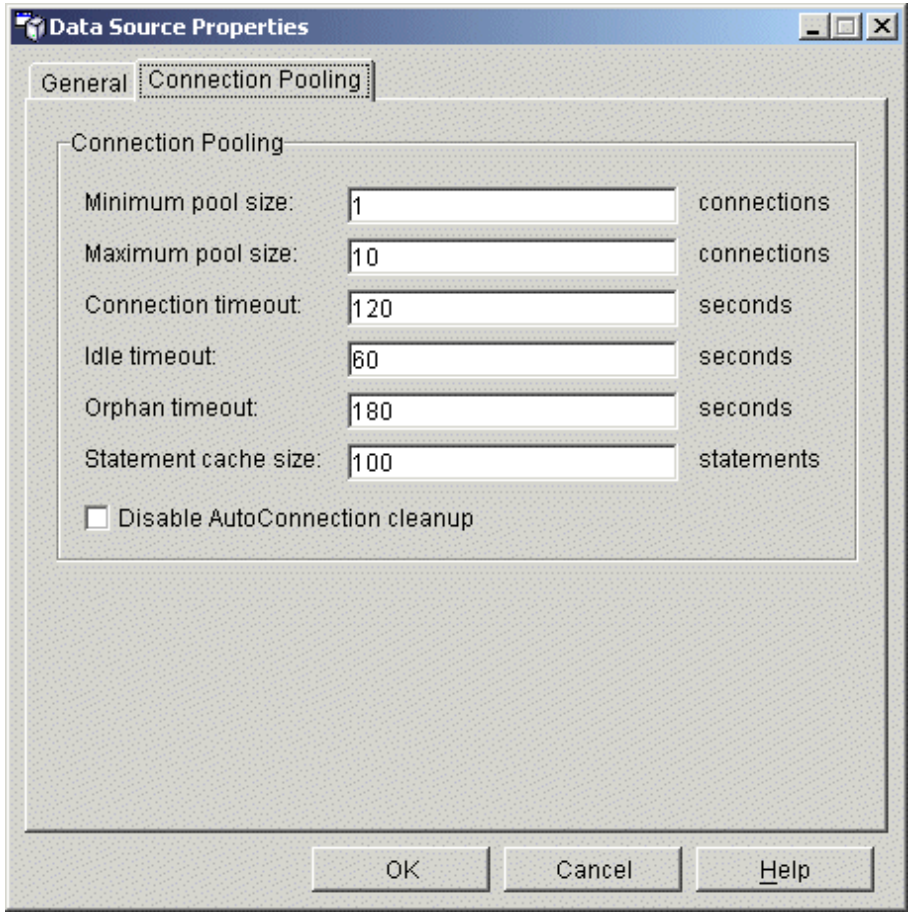
WebSphere DataSources

- WAS provides a Java Naming service JNDI implementation to ease access to DB2.
- DB2 databases must be defined within WAS.
- Port number is not required for use of Type 2 JDBC driver
- Connection attributes such as default userids and passwords can be defined.



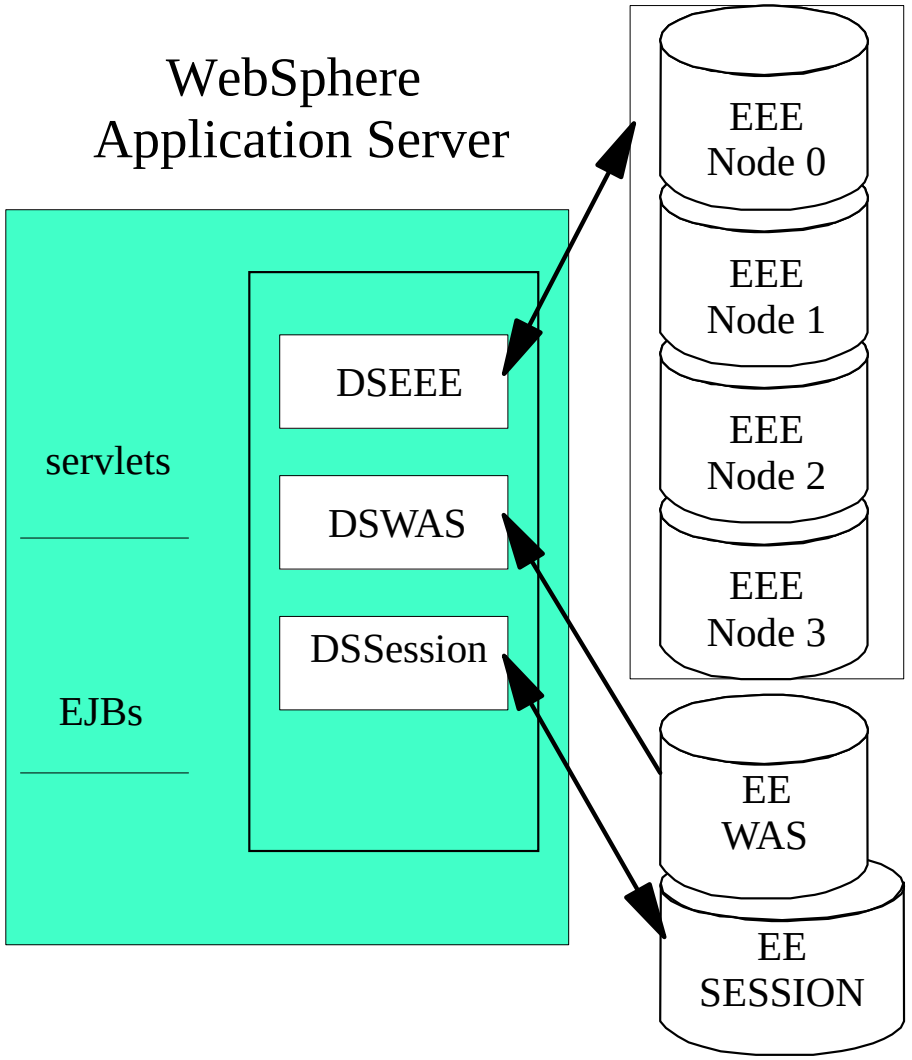
WebSphere Connection Pooling

- Pool is shared by all apps using this datasource
- Bigger is not always better
- Min/Max can be set to the same value to avoid releasing physical connections
- MAXAGENTS must be at least as big as the sum of all max pool sizes
- Note: ECPerf used maximum pool size of 24



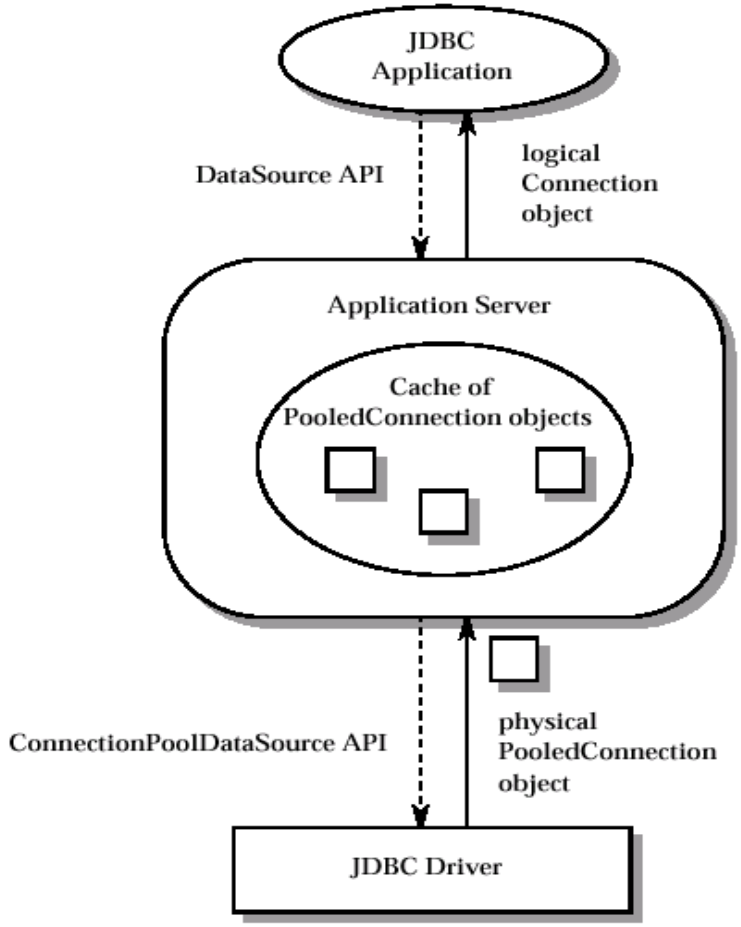
Considerations with DB2 EEE

- Accessing DB2 UDB Enteprise Extended Edition (EEE) DBs from WAS is supported.
- WAS session and administration DB should be created in a EE instance (can use EEE installation)
- WAS data sources can be directed to specific DB2 partitions using data source properties.



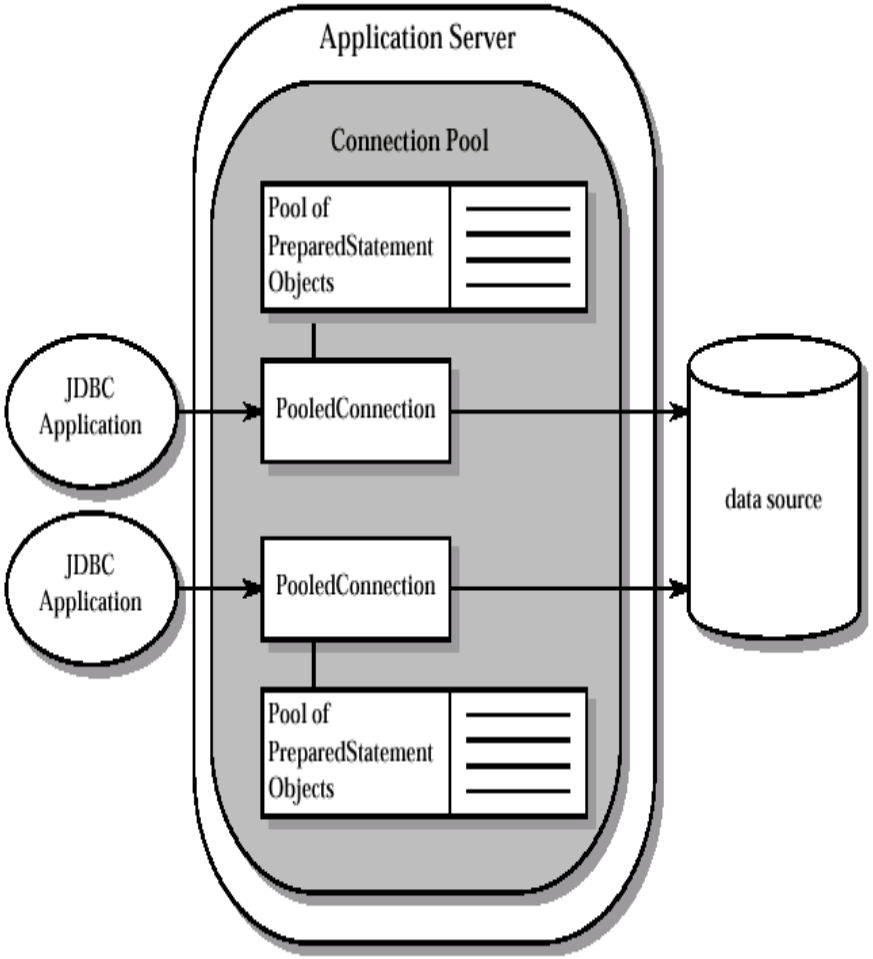
WAS tuning – Connection Pools

- WebSphere pools connections to DB2.
- A pool for each unique DataSource (userid /password / other properties)
- New physical connection only obtained if none in pool match
- Eliminates network traffic to remote databases



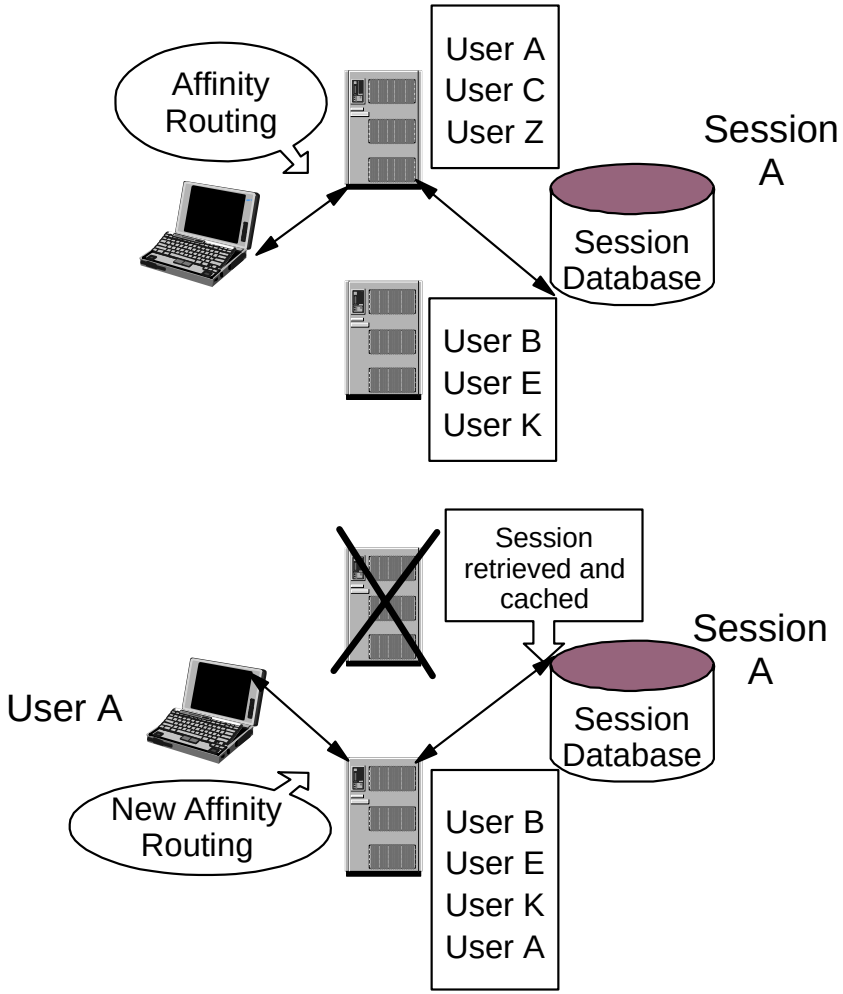
WebSphere Statement Pooling

- WAS will check its cache before flowing request to DB2
- WAS cache is only used for PreparedStatements
- DB2's PKGCACHE is used for all SQL (dynamic & static).
- Both caches should be monitored and tuned.
- Note: ECPerf had a WAS cache of 1560



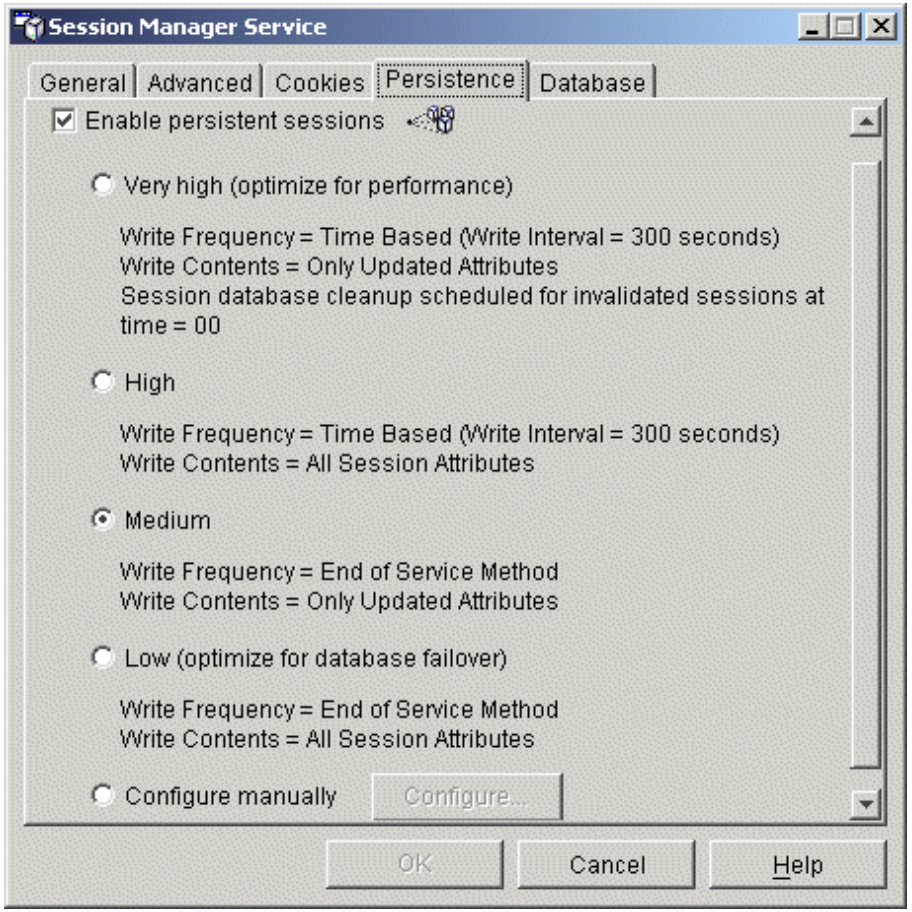
WebSphere Session Affinity

- DB2 can provide the session persistence service to allow continuous operation for applications.
- User applications are dynamically rerouted to an available WAS node and the session data is retrieved.



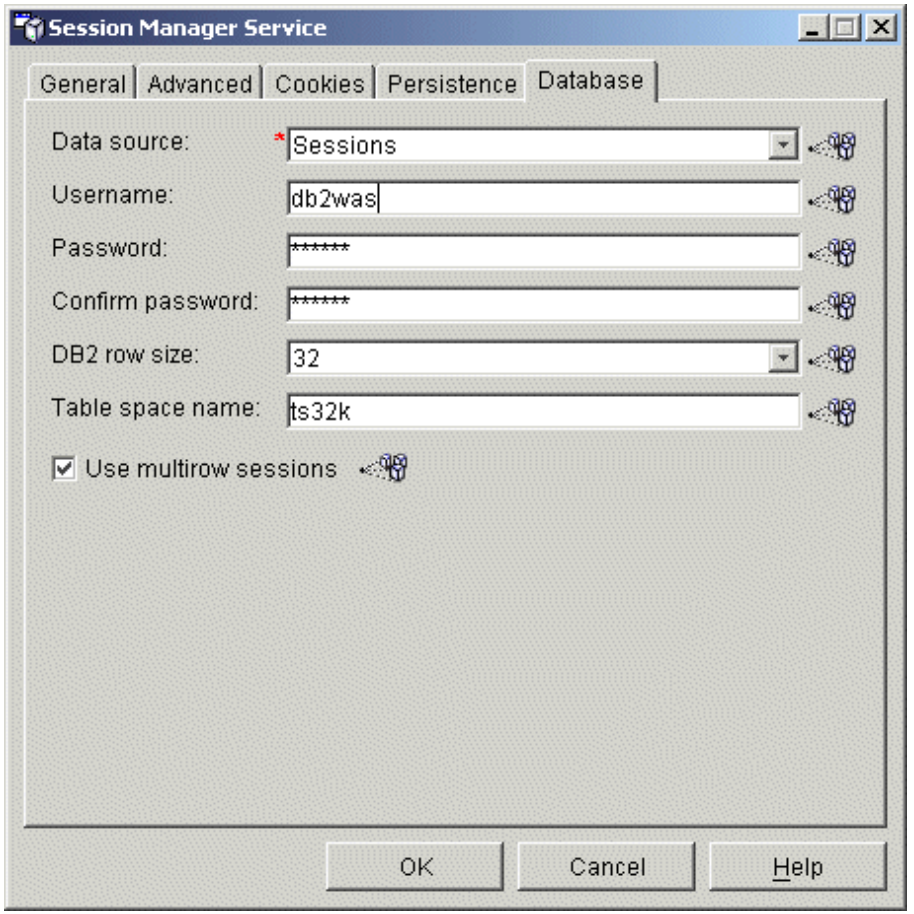
Enabling Persistent Sessions

- Purpose:
  - WAS will managed HTTP session objects on behalf of the application to enable high availability (failover)
- Single session persistence table is highly updated and volatile in size
- Use DB2 SQL snapshots to understand access patterns
- Update activity is controllable.



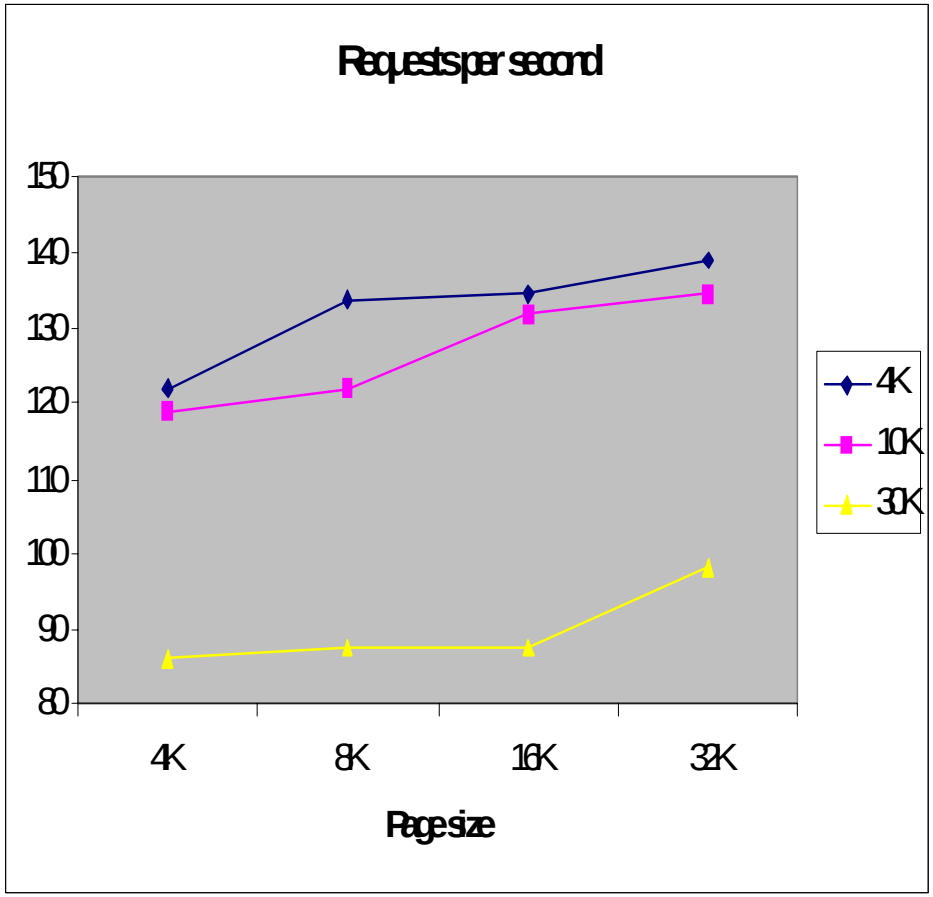
Session DataSource Properties

- Uses the same WAS Connection Pooling
- Single or multiple rows used to represent HTTPSession data
- Large tablespaces can be used for performance.



WAS exploitation of DB2's Larger Page Sizes

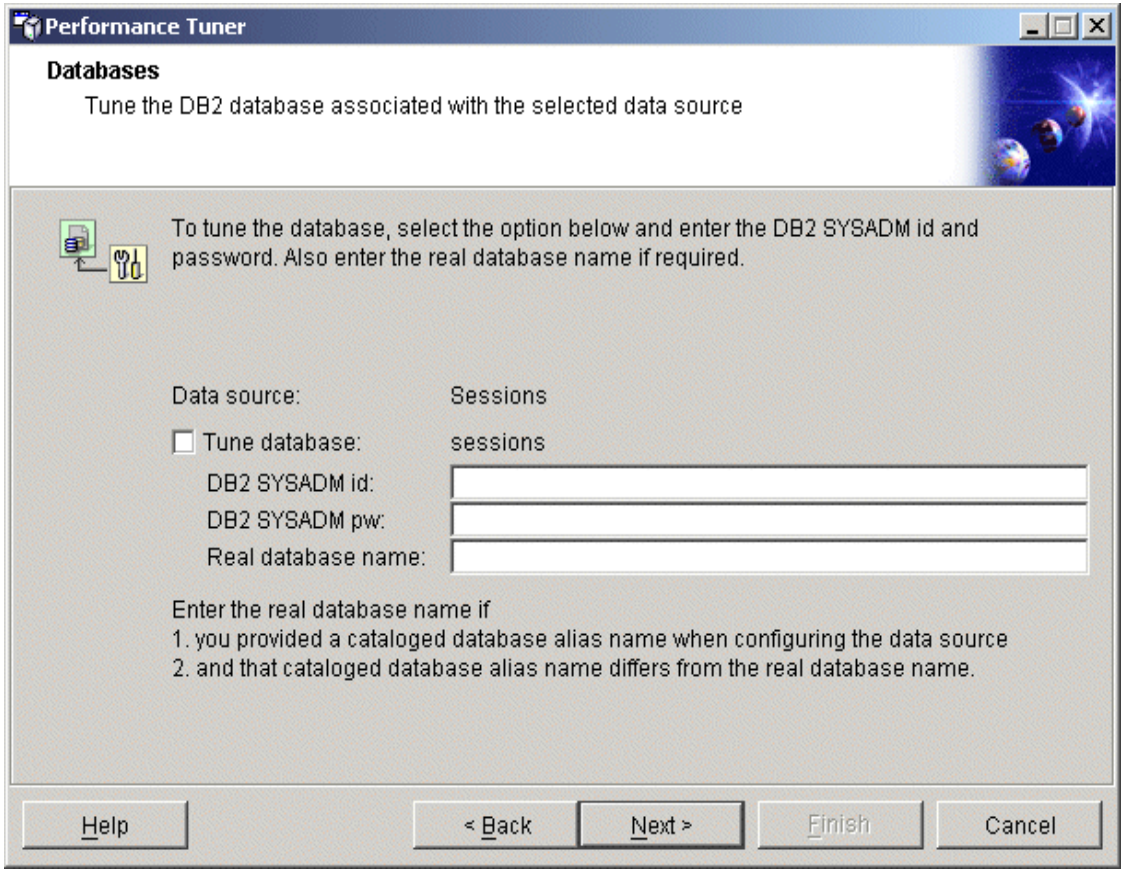
- Persisting HTTP Session data is important for failover of J2EE applications
- WAS v4 provides the ability to store session data as VARCHAR values using DB2's 16KB or 32KB pages
- Overall system performance improves approximately 15%



Source: **Tuning DB2 UDB for Session Persistence, IBM white paper (2002)**

WAS Integrated Performance Wizard - DB2 only

- DataSource is associated with an Application Server
- Can be run over a network
- Tunes database for OLTP access
- Provides easy point and click tuning of datasources



DB2 Tuning - Memory Usage

- Database Manager configuration
  - MAXAGENTS
    - The maximum number of agents that can be started
    - Must be high enough to handle maximum pools for all WAS connection pools
  - NUM_INITAGENTS
    - Number of agents started when **db2start** is issued
    - Used to avoid initial connection setup time
  - NUM_POOLAGENTS
    - Specifies maximum size of the agent pool (idle and in-use agents)
- Database configuration
  - APPLHEAPSZ
    - Holds application related information on the DB2 server
  - PKGCACHE
    - Set according to WAS statement cache + other SQL
  - LOGBUFSZ / DBHEAP / MINCOMMIT
    - The LOGBUFSZ is used along with the MINCOMMIT parameter to improve I/O wait conditions on DB2's transaction log files

DB2 Tuning - Access Plans

- Volatile tables
  - Table size varies frequently
  - Forces DB2 to perform index scans
- Optimization level
- Database Manager Configuration Parameters
  - INTRA_PARALLEL
    - For use in SMP systems
    - Specifies whether a database manager can use parallel process for processing queries
  - MAX_QUERYDEGREE
    - Specifies the maximum degree of intra-parallelism used for any SQL statement
- Database Configuration Parameters
  - AVG_APPLS
    - Specifies the average number of active applications
    - Optimizer uses this to determine the amount of bufferpool that will be available at runtime
    - Database Configuration parameter

DB2 Tuning - Concurrency

- Database Configuration Parameters
  - LOCKLIST
    - Specifies the maximum storage in pages allocated to the lock list
    - Larger LOCKLIST size avoids lock escalation
  - MAXLOCKS
    - Specifies the maximum percentage of the lock list that any single application can use
    - This parameter needs to be carefully tuned
    - A large value could cause lock escalations as any one application could use too much of the lock list
    - A small value could cause lock escalations since each application could run out of locking space quickly
- JDBC Isolation levels and DB2 Isolation Levels have different definitions

JDBC Isolation Levels	DB2 Isolation Levels
TRANSACTION_SERIALIZABLE	Repeatable Read (RR)
TRANSACTION_REPEATABLE_READ	Read Stability (RS)
TRANSACTION_READ_COMMITTED	Cursor Stability (CS)
TRANSACTION_DIRTY_UNCOMITTED	Uncommitted Read (UR)

DB2 Tuning - Physical Layout

- Indexes
  - ensure they are defined
  - Clustering indexes for improved access
  - Multi-dimensional clusters (MDC)
- Large page
  - Large page sizes improve I/O (as shown in session persistence
- Buffer pools
  - Use available memory on DB2 server
  - Proper bufferpool sizes can reduce I/O
- Tablespaces
  - Multiple containers to reduce I/O

DB2/WAS Best Practices

- Don't use DriverManager (no pooling)
  - ~~java.sql.DriverManager.getConnection (url, user, pwd)~~
- Use DataSources (WebSphere v4 syntax)
  - WebSphere implements javax.sql.DataSource based on
    - javax.sql.ConnectionPoolDataSource
    - javax.sql.XADataSource
- Create an Initial Naming Context (JNDI) for ds lookup

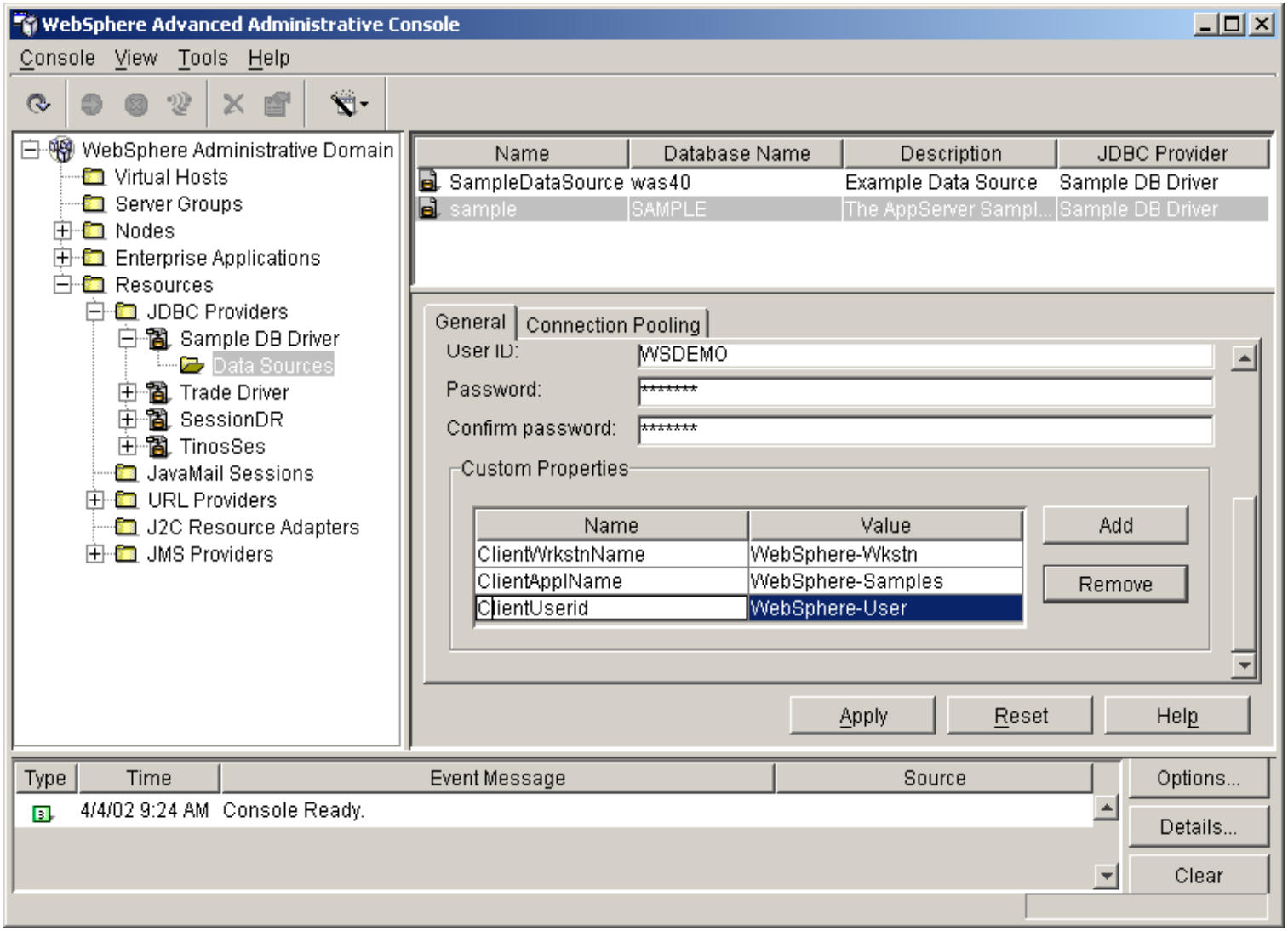
  javax.sql.DataSource ds = (javax.sql.DataSource) ctx.lookup java:comp/env/jdbc/SAMPLE");
  java.sql.Connection con = ds.getConnection();
- Programming Model - catch and release

```
catch ( java.sql.SQLException sqle)
{}
finally
{
  rs.close();
 stmt.close();
 conn.close() ;
}
```

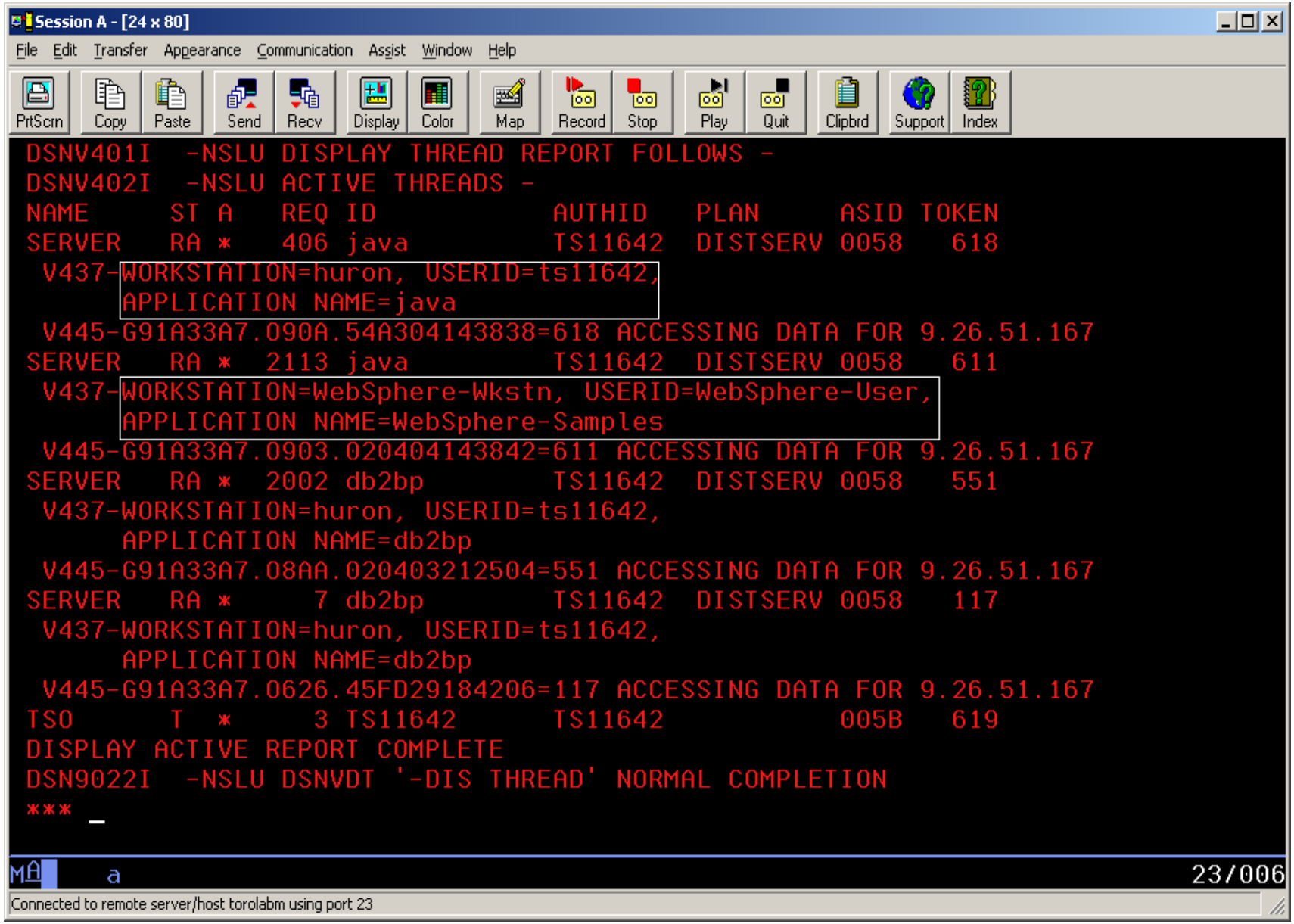
Monitoring DB2 and WebSphere

- DB2
  - Snapshot monitor – operational, connections, statements, bottlenecks (locks, logging)
  - Event monitor – deadlocks
  - Explain – poor access plans
  - JDBC trace analysis
- WAS
  - Resource analyzer can be used to monitor connection pools, statement caches, queues
- OS level facilities – netstat, iostat, vmstat, etc….

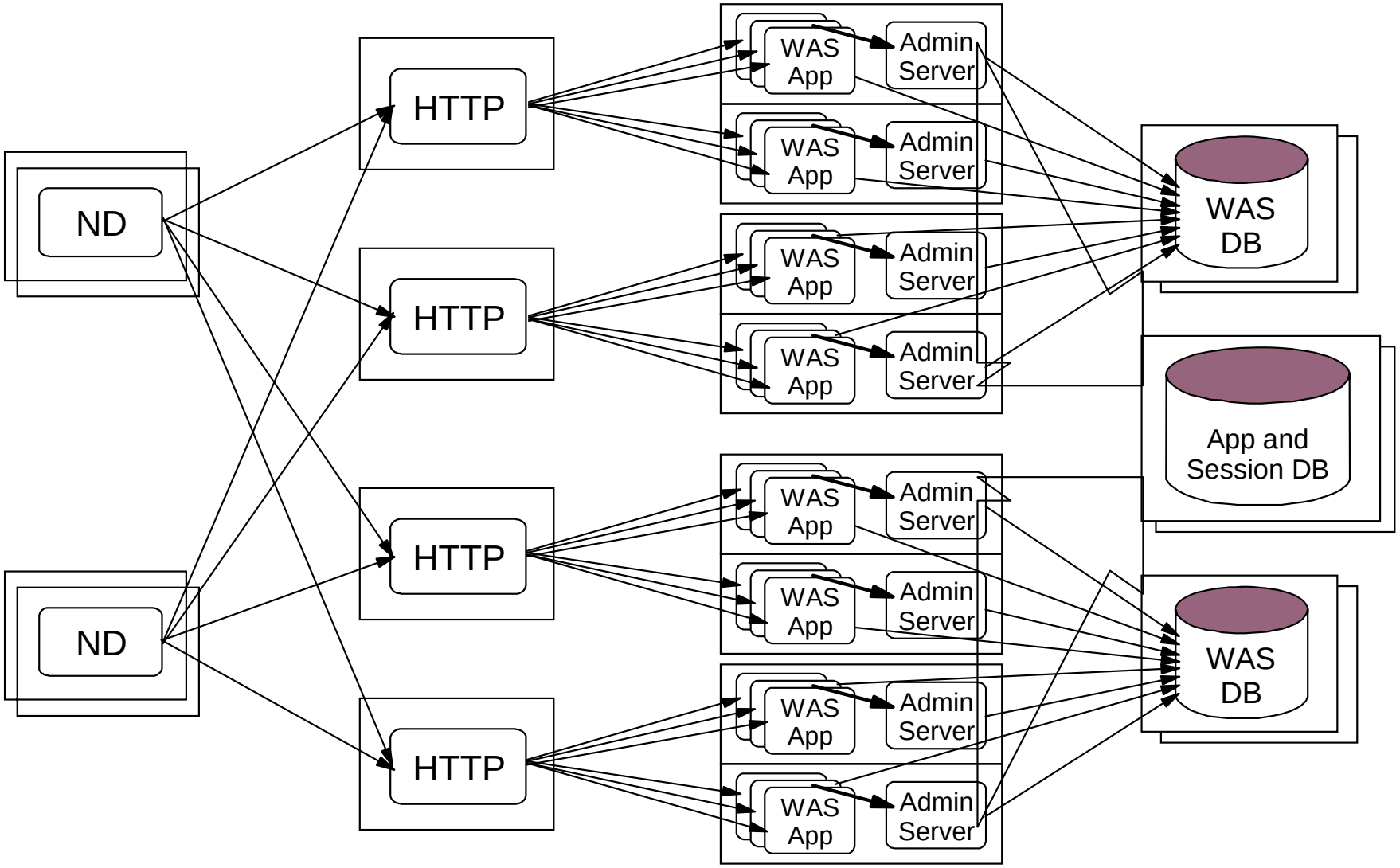
Monitoring WebSphere Access to DB2 for OS/390



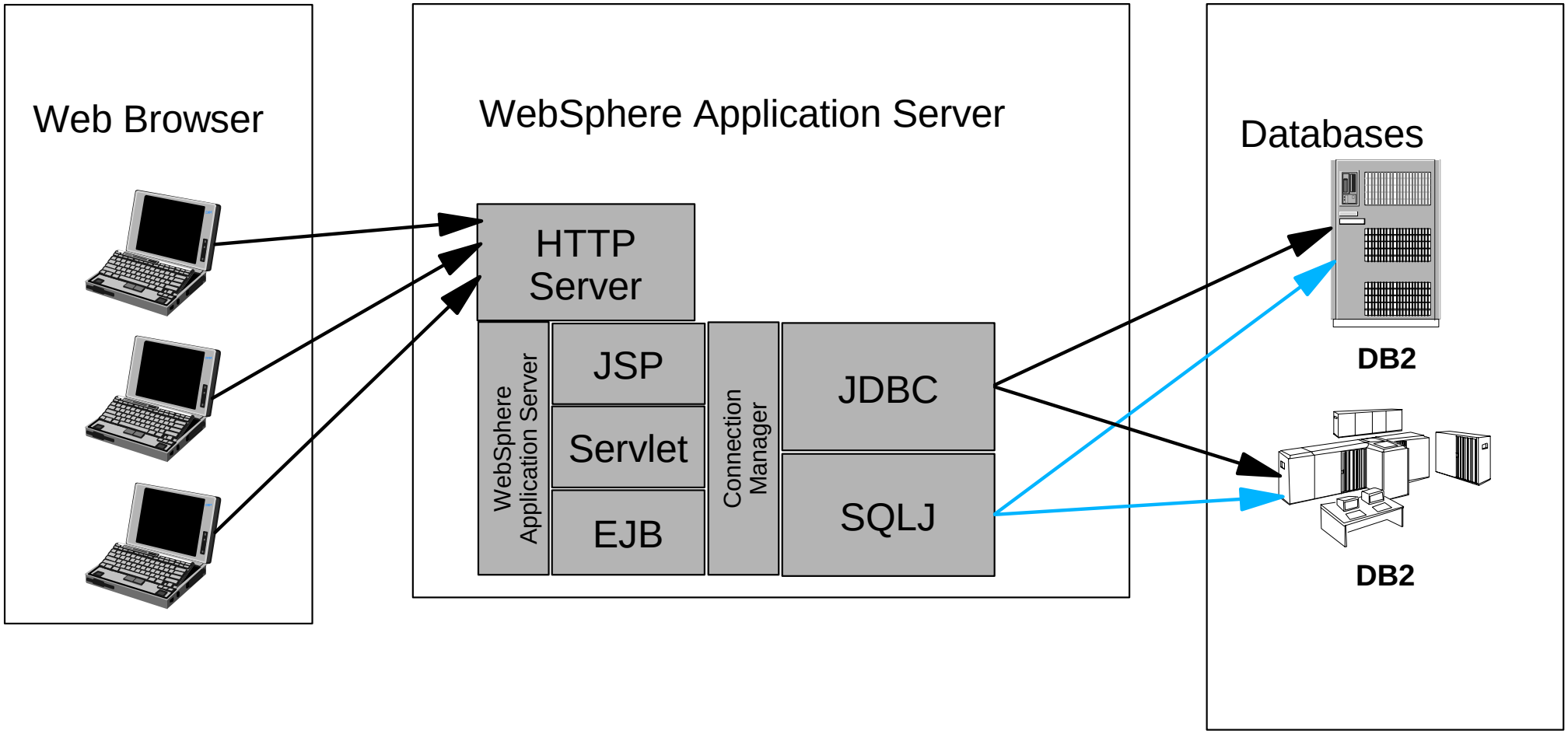
Monitoring WebSphere Access to DB2 for OS/390 cont.



Scalability of DB2/WAS



J2EE Architecture Overview



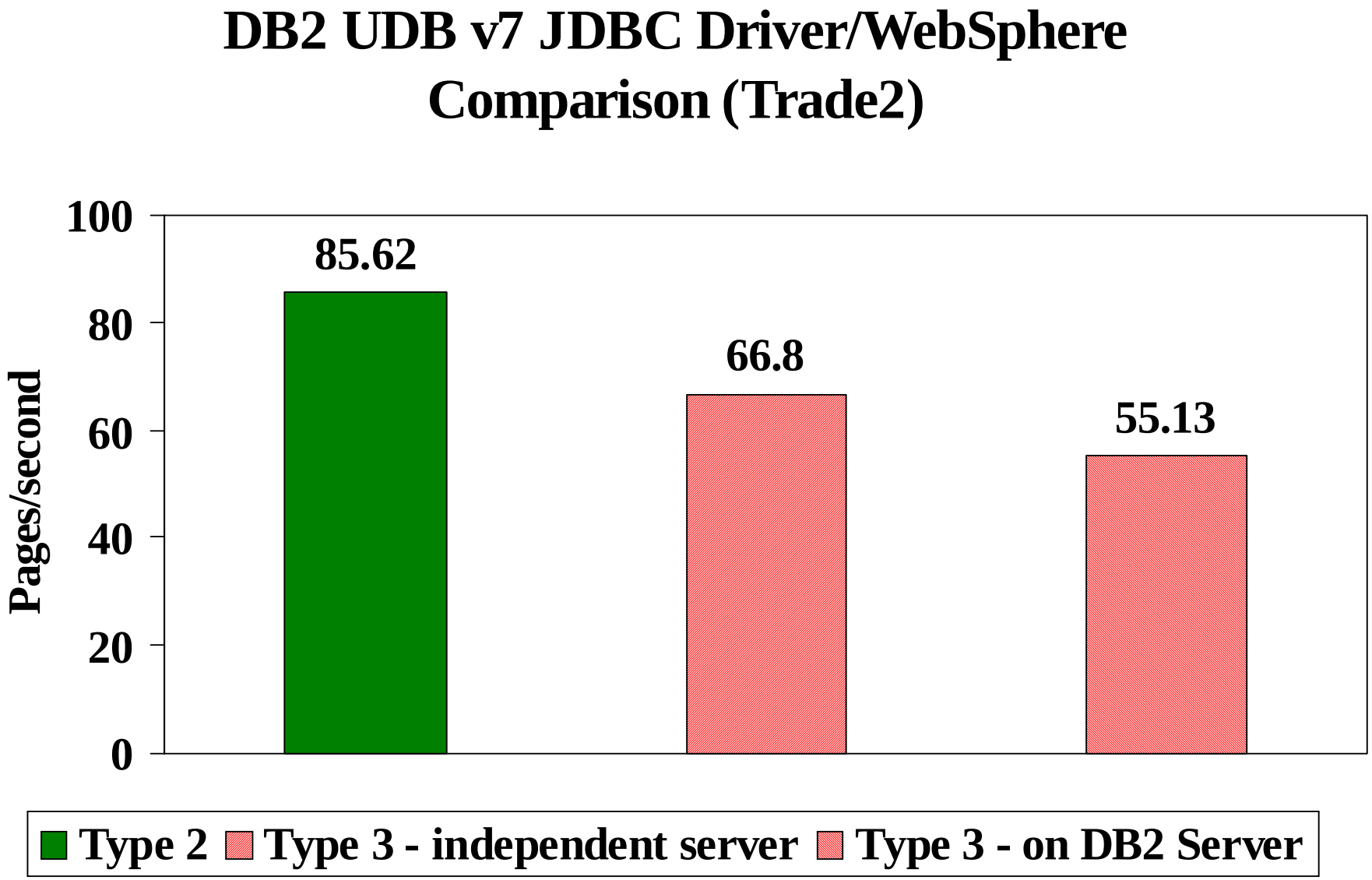
DB2 Developing Applications with Java

- JDBC – Dynamic SQL interface
  - Subset of Java™ 2 Platform, Standard Edition (J2SE 2)
    - JDBC 2.x
    - JDBC 3.x
    - Defined by Sun Microsystems
- SQLJ – Embedded SQL in Java language
  - Defined by SQLj.org and ANSI
    - SQLJ: Embedded SQL in Java methods – Part 10:SQL/OLB standard (ANSI)
    - SQLJ: SQL Routines - calling Java static methods as SQL stored procedures and user-defined functions – Part 1
    - SQLj: SQL Types – Part 2

DB2 JDBC Driver Architecture

- Type 2 – J2EE 1.3 Certified
  - As of DB2 UDB Version 7 FixPak 5
    - API changes have been made in FixPak 5 to fit J2EE specifications
  - Used for WebSphere Applications
  - High performance
- Type 3 – Ease of use
  - Native Java driver
  - Requires DB2 JDBC applet server running
- DRDA-based JDBC drivers coming
  - Type 2 – highly optimized for DB2
    - Based on DRDA
  - Type 4 – coming to a DB2 near you !!
- http://www.opengroup.org/dbiop/

DB2 JDBC Architecture - Performance



Coding around Network traffic

- The **Statement** interface has two methods that can be used to provide hints to the JDBC driver
  - setFetchSize
    - setFetchSize should be tuned to match DB2's record blocking parameter (RQRIOBLK)
    - FetchSize attributes works with FORWARD ONLY and SCROLLABLE ResultSets (as of FP7)
  - setFetchDirection
- The values supplied to these methods are applied to each result set produced by the Statement object
- The ResultSet interface contains the same methods that can be used to supply hints for just that result set

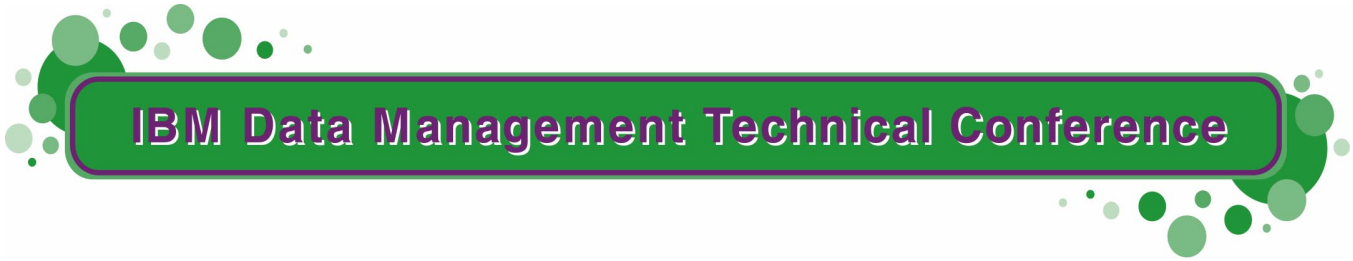
Additional Information

- DB2 links
  - Developer Domain / Samples
    - http://www-3.ibm.com/software/data/developer/samples/video/ - IBM Video Central Sample
    - http://www.ibm.com/software/data/developer - Great articles on DB2 Development
  - Integration
    - http:/www.ibm.com/software/data/integration/websphere (external)
    - http://www.ibm.com/software/data/webservices
  - J2EE 1.3 Certification - 12/20/2001
    - http://java.sun.com/j2ee/compatibility.html
  - DB2/WebSphere benchmark result (Trade2)
    - http://www-4.ibm.com/software/data/bi/teraplex/scaleebus.pdf
- **Redbooks (http://www.redbooks.ibm.com)**
  - DB2 UDB/WebSphere Performance Tuning Guide (SQ24-6417-00)
  - DB2 e-business Guide (SG-246539-00)

F19

DB2 UDB and WebSphere Application Server at Light Speed

Quentin Presley



Anaheim, CA **Sept 9 - 13, 2002**

RETURN TO INDEX