ORACLE

Oracle Coherence &
Extreme Transaction Processing (XTP)

Oracle Coherence Workshop

ORACLE Coherence

Agenda

- Extreme Transaction Processing Defined
- Coherence as a Data Grid
- Coherence for Data Queries, Transaction and Event
- Case Studies
- What is Coherence? And Why Coherence?
- Q/A

Extreme Transaction Processing

- What is XTP?
- Introduction to Oracle Coherence
- Coherence Value Proposition

Company

Oracle

- **Leading Provider of Data Grid Solutions**
- **Key Player in Virtualization Space**

Coherence™ Data Grid

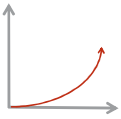
- Reliable**, Distributed, In-Memory Data Management-** Clustered Caching
- **Data Services for**
  - **SOA, Virtualization, Compute Grids**

Quick Facts

- Over 1500 production applications using Coherence
- JCP: JCache (JSR-107) Spec Lead; WorkManager (JSR-236/237) Expert Group Member

Oracle Acquires Tangosol

- Acquisition closed June 1st
- Tangosol Coherence will remain a stand alone product line and will continue to be developed as such
- Like other products in Oracle's Fusion Middleware group, Coherence will be hot pluggable and continue to fully support many different middleware software vendors.



XTP Defined

An application style aimed at supporting secure, large-scale, high-performing transactions across a distributed environment on commodity hardware and software

Gartner Group

XTP Evolution

Problem

- **Explosive growth of user bases overloading system capacity**
- **Rate of change faster than IT's ability to re-architect systems**

Challenge

- **Do or Die Market Pressure to Meet User Expectations**
  - Changing Service Level Agreements
  - Constant pressure to add new features and services
- **Intense Pressure on Cost:** Rapid infrastructure growth

Solution

- **Unlimited, predictable scalability** with capacity on demand
- **Extreme performance** with zero latency
- **Dynamic by nature,** able to easily change
- **100% transactional integrity** and data reliability
- **Continuous availability**

For data intensive high volume mission critical applications

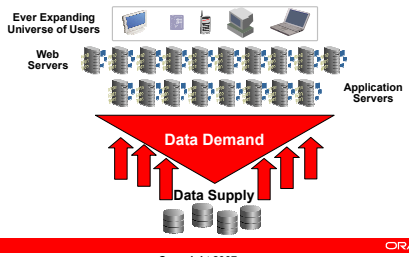
XTP & Oracle Coherence

Coherence was designed for:

- Brokering supply and demand of data and processing capacity between the application tier and backend data sources
- Massive scale out of the middle tier
- Reliable transaction processing at any load
- Continuous availability to ensure business continuity

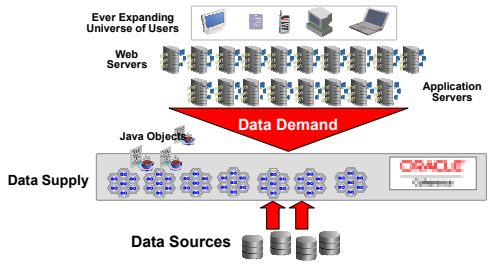
Oracle Extreme Transaction Processing (XTP)

- **Data Demand outpacing Data Supply**
- **Rate of growth outpacing ability to cost effectively scale applications**



Oracle Extreme Transaction Processing (XTP)

- Oracle Coherence brokers Data Supply with Data Demand
- Scale out Data Grid in middle tier using commodity hardware



Customer Story: GEICO Insurance

Profile

- 4th largest private auto insurance company
- 3rd largest P&C insurer in US
- $22 Billion in assets

Implementation Details

- 150 CPUs in production

Scenario

- One of the largest online insurance providers in the US with self- service website for customers
- Scaled out application tier built to handle high volume of traffic

Problem

- Large Database was heavily loaded by persistence of enormous user profiles (>1MB each) for thousands of concurrent users
- Challenge expanding environment for spikes in usage or additional services offered

Solution

- Oracle Coherence allows all customer data to be managed in-memory for fast access to user profiles
- Updates to profiles are actively held in the Data Grid and only persisted to the database once, at the end of the user session

Over 10X increase in application tier capacity

Data Grid Uses

Caching
Applications request data from the Data Grid rather than backend data sources

Analytics
Applications ask the Data Grid questions from simple queries to advanced scenario modeling

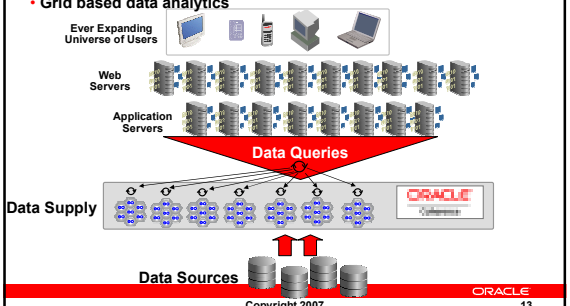
Transactions
Data Grid acts as a transactional System of Record, hosting data and business logic

Events
Automated processing based on event

Oracle Extreme Transaction Processing (XTP)

Comprehensive Infrastructure to deliver XTP

- Oracle Coherence supports powerful parallel Queries
- Grid based data analytics



Customer Story: Wachovia

Profile

- 46th on Fortune 500 list
- About 3400 locations
- $754 Billion in assets

Implementation Details

- 300 CPUs in production

Scenario

- Wachovia Investment Bank introducing "Service Oriented Infrastructure (SOI)"
- Requires absolute data availability for complex Grid Computations

Problem

- Existing Compute Grid infrastructure suffering from data latency and throughput problems
- Complex calculations so lengthy as to be outdated

Solution

- Data Grid overlay on Compute Grid
- Enable risk calculations to fully utilized the grid hardware by having real time access to in-memory data as well as parallelization .
- Reduced critical risk computation from **several days** to under **1 hour**!

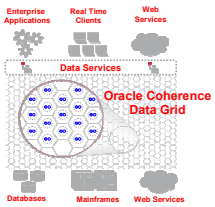
Oracle Coherence

Over 100 Direct Customers and 1,500+ Production Installations

Coherence Technical Overview

Oracle Coherence Data Grid

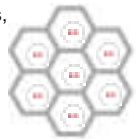
Distributed in Memory Data Management



- Provides a **reliable data tier** with a single, consistent view of data
- Enables dynamic data capacity including **fault tolerance** and load balancing
- Ensures that **data capacity scales** with **processing capacity**

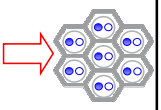
Universal Access & Management

- All data in the Data Grid is accessible from any single node
  - Single System Image = Simple programming paradigm
  - Automatic data partitioning and redundancy
- Optimizes data locality in Grid based on usage or access
  - Move state or behavior
- Parallelizes data loading, data queries, data managed in grid
- Database integration
  - Blocking write-through (Synchronous)
  - Reliable write-behind (Asynchronous)



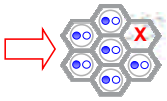
How Does Oracle Coherence Data Grid Work?

- Data load-balanced in-memory across a cluster of servers
- Data automatically and synchronously replicated to at least one other server for continuous availability
- Single System Image: Logical view of all data on all servers



- Servers monitor the health of each other
- In the event a server fails or is unhealthy, other servers cooperatively diagnose the state

- The healthy servers immediately assume the responsibilities of the failed server
- Continuous Operation: No interruption of service or loss of data due when a server fails



Coherence: A Unique Approach

- In Coherence…
  - Members **share** responsibilities (health, services, data…)
  - Completely Peer-to-Peer
  - No Single Points of Bottleneck (SPOBs)
  - No Single Points of Failure (SPOFs)
  - Linearly scalable to thousands of servers by design
- Servers form a full "mesh"
  - No Masters / Slaves etc.
  - Data Grid members work together as a team
  - Communication is almost always point-to-point
    - Designed for commodity switched infrastructures
    - Scalable throughput up to the limit of the backplane

Architectural Integration Approaches

- Architect Solutions with Coherence
  - Simple Java programming API for J2EE
  - .NET integration using C#
  - Read-Through / Write-Through / Write-Behind or Cache Aside
- Plug into Existing Applications
  - Hibernate/TopLink Integration
  - Session state scaling with Coherence Web
    - .NET or Java EE session state
- Pluggable Integration with Oracle (roadmap)
  - SOA Suite
  - WebCenter
  - Business Intelligence
  - Content Management
  - …

Oracle Coherence broad integration

- **Hot pluggable**
  - Broad support for leading App Servers: Oracle Websphere, Weblogic JBoss, Sun, ect…
- **Helps any back end DB environment**
  - Oracle, Sybase, DB2, SQL Server
- **Any vertical, Any application**
  - (java, .net, soon C++) that needs **Predictable Scalability**

Oracle Coherence Advantage

- **Protect the Database Investment**
  - Ensure DB does what it does best, limit cost of re-architecture
- **Scale as you Grow**
  - Cost effective: Start small with 3-5 servers, scale to hundreds of servers as business grows
- **Enables business continuity**
  - Providing continuous data availability

What is Coherence (Finally)

Coherence Is ..

- An object-oriented data manager for the grid
  - Data is managed in memory
  - Runs inside Java Virtual Machines (JVMs)
  - Client applications can be Java or C#/.NET
  - Distributed/partitioned across potentially hundreds of JVMs and dozens of servers
- Automatic "scale out" / horizontal scaling
  - If you add new servers, they automatically join the cluster and re-distribute the data evenly
  - Hardware is usually inexpensive commodity servers
- Automatic high availability
  - Every object in memory has a backup on another server
  - If the primary server fails, the backup takes over and makes another backup
  - A "Consensus" algorithm keeps track of which object is the "primary" (more later)

What is Coherence?

- Coherence (deployment perspective)
  - Single Library*
  - Standard Java Archive "JAR" for Java
  - Standard Dynamically Linked Library "DLL" for .NET connectivity (.Net 1.1 and 2.0)
  - *Other libraries for integration (Databases, Spring…)
  - No 3rd party dependencies!
  - Minimal "invasion" on standard code*
  - Configurable implementations of standard Map / Dictionary interfaces (NamedCache)
  - Provides Predictable Scalable Caching
  - "RemoteException" free distributed computing

What is Coherence?

- Coherence (architectural perspective)
  - Scale-out Applications State
  - Reliable Data Management / Data Abstraction Layer
  - Effortlessly Cluster Applications (clustering infrastructure)
  - Web (session management)
  - Front, Middle, Back Tiers
  - Thick Clients (AWT, Swing, Console, RCP…)
  - JSE or JEE
  - Remote Connectivity
  - Business Continuity and Disaster Recovery
  - Provide a Data Grid

Different Needs Require a Different Data Manager

APPLICATION TIER

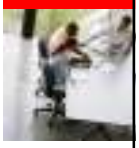
- Manage data as objects
- Manage data in memory – or the applications won't scale
- Scale horizontally
  - Add new, inexpensive nodes when more capacity is needed
- Query by object ID, meet the immediate needs of the app

DATABASE TIER

- Manage data as rows/columns
- Manage data on disk
  - Long-term persistent store
- Scale vertically
  - Usually SMP, big boxes, big storage devices, etc.
- Ad-hoc query, data warehouse query, SQL

Very different needs = different data management

What Is Oracle Coherence … NOT?



What Oracle Coherence isn't!

- NOT a "drop in solution"
  - You must write or change Java code to implement Coherence
- You can perform database-like operations …:
  - Persist data transiently in memory on the grid
  - Do queries – in parallel
  - Perform indexing (like a DB)
  - Do things like Stored Procedures
  - Establish real-time views (like Materialized Views)
- However, Coherence is NOT a database
  - No SQL or ad-hoc query language
  - Queries must be predictable
  - It's not TimesTen

What Oracle Coherence isn't!

- It's not a messaging system
  - You can use Events and Listeners for data inserts, updates, deletes
  - You can use Agents to handle data changes
  - You can use Filters to filter events
- It's not "just" a Cache!
  - Caches expire data
  - Customers actually turn expiry off!
  - Customers use Coherence for mission-critical transactional applications
  - Why do they take that risk? Why do we let them?
  - Data management is based on reliable clustering technology. It's stable, reliable and highly available.

Why Oracle Coherence?

Why Oracle Coherence?

- Scale-out stateful applications
- "If you need business agility!"
- Save resources!
  - Avoid managing clusters
  - Avoid designing systems around specialized "cluster masters"
  - Avoid manually "coding in" data and service partitions
- If you want to share a collection of Data and Services
  - Oracle Coherence does more than just Caching!
  - Oracle Coherence manages clusters
  - Oracle Coherence can manage a Grid
  - Oracle Coherence can manage your data in a Grid
  - Oracle Coherence can provide your services within a Grid to clients
- If you want truly native language support! No wrappers or embedded third-party libraries

Why Oracle Coherence?

- If you want predictable scale-out costs, without re-coding or reconfiguring!
- If you want the most trusted and recognized best-of-breed clustering and Data Management platform inside your solutions
- If you want a platform on which to reliably perform eXtreme Transaction processing

Why Oracle Coherence?

- Built for continuous operation
- Data Fault Tolerance
- Self-Diagnosis and Healing
- "Once and Only Once" Processing

- Dynamically Expandable
- No data loss at any volume
- No interruption of service
- Leverage Commodity Hardware
- Cost Effective

- Single view of data
- Single management view
- Simple programming model
- Any Application
- Any Data Source

- Data Caching
- Analytics
- Transaction Processing
- Event Processing

For More Information

http://search.oracle.com

Coherence

or

http://www.oracle.com/products/middleware/coherence/index.html

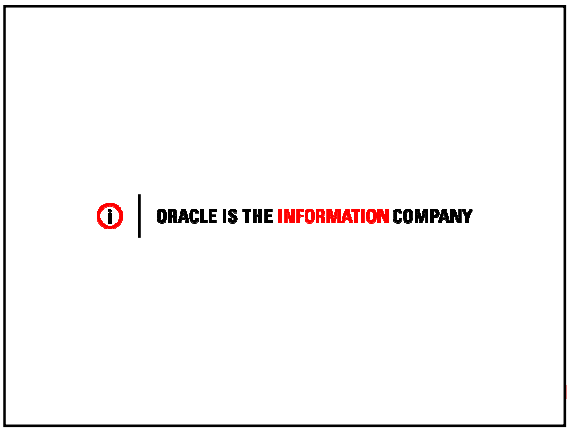
Lab Environment

- Software requirements
  - Oracle Coherence
  - JDeveloper 11g (or Eclipse if you prefer)
  - Lab workbook
- For this High Level Workshop, the Key is just to understand how us Coherence to access objects from within the case.
- Only Labs 1-4 will be done

Labs 1 & 2

- Lab 1
  - Install Oracle Coherence and configure JDeveloper 11g for the course labs
- Lab 2
  - Run the sample Coherence application

Q&A


ORACLE IS THE INFORMATION COMPANY


ORACLE