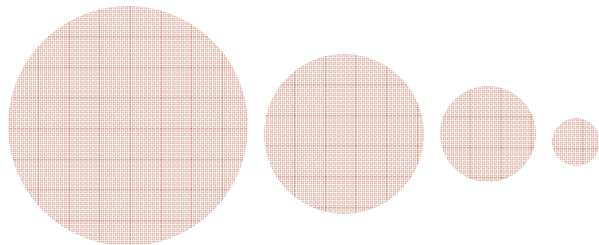


Data Fabric Enabling of Data Intensive JEE Applications

Asif H. Shahid
Architect,
GemStone Systems



IndicThreads.com Conference On Java Technology
26th & 27 Oct. 2007 Pune, India



Agenda

- **Challenges in clustered enterprise java applications**
- Solution - Enterprise Data Fabric (EDF)
- Data Distribution Model
- Caching Topologies
- Other Elements of Enterprise Data Fabric
- Real World Use Cases
 - Look at problems and solution architecture
- Summary



Challenges in clustered enterprise java applications

- **Increase Throughput**
 - Processes are limited by the inflow and outflow of data
- **Increase Scalability**
 - Load balancing a request to the optimal node with old states retained
- **Reliability and Quality of Service**
 - No SPOF
 - Transparent failover



How can these challenges be met?

- Data sharing and distribution among cluster JVM software
 - For load balancing & parallel computing data needs to be distributed and shared
- Data consistency management
 - In a multi VM multi threaded env., consistency guarantees needed
- Data access from disparate sources
 - Aggregation of data from xml files, RDBMS, EAI systems



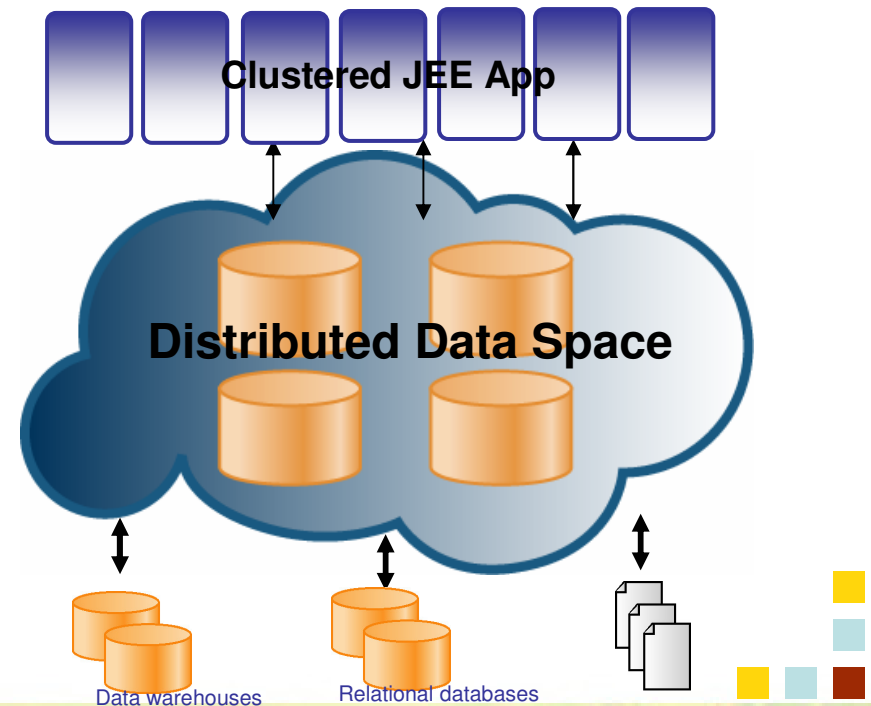
Agenda

- Challenges in clustered enterprise java applications
- **Solution - Enterprise Data Fabric (EDF)**
- Data Distribution Model
- Caching Topologies
- Other Elements of Enterprise Data Fabric
- Real World Use Cases
 - Look at problems and solution architecture
- Summary



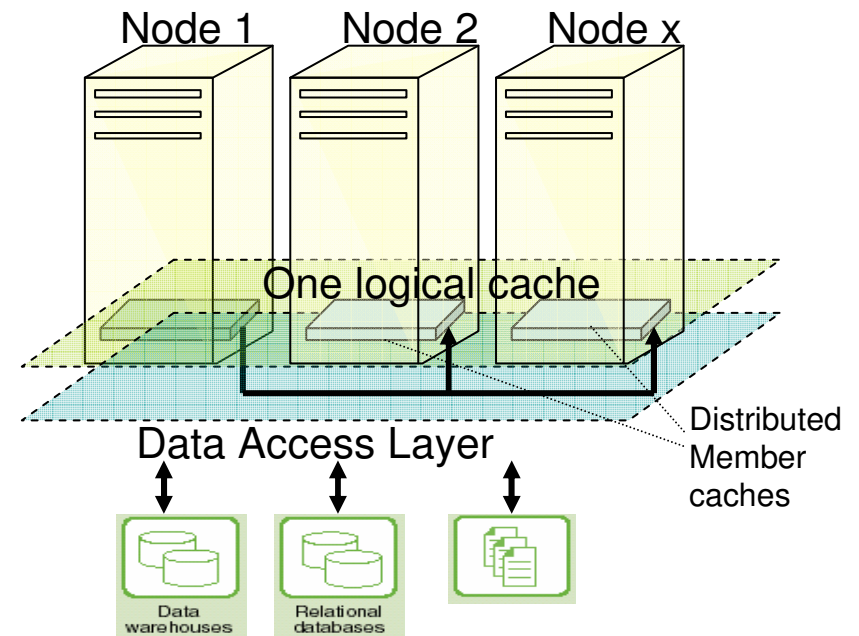
Solution - Enterprise Data Fabric

- Middle tier Distributed Data Infrastructure
 - Combines Distributed Caching (main-memory), DB semantics, Reliable Messaging/ Intelligent Routing and Continuous Analytics
- In a Data fabric
 - Data provisioned in the cluster nodes itself
 - Pool memory and disk across grid nodes
 - Access and synchronize data with data sources
 - Abstracts the app away from data sources



Enterprise Data Fabric : continued

- Fabric primarily manages data in memory
 - Eliminate disk bottlenecks
- Replicates data with consistency guarantees to multiple nodes
 - Harness CPU across nodes for Data Management
- Number of nodes used can dynamically increase or decrease
- Access is load balanced across many data servers
- Fabric has smarts to detect faults and failover



Summarising EDF

- An EDF
 - is a distributed, in-memory data caching software
 - provides ways to access & update data in the fabric
 - Synchs up with the underlying datasources



Agenda

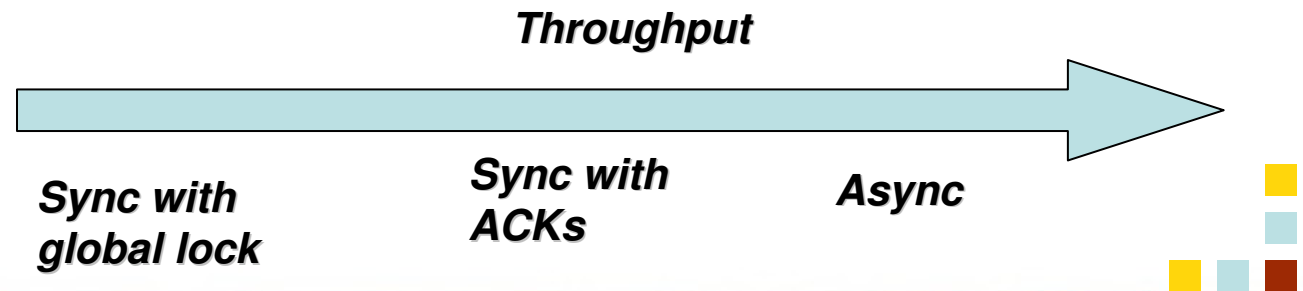
- Challenges in clustered enterprise java applications
- Solution - Enterprise Data Fabric (EDF)
- **Data Distribution Model**
- Caching Topologies
- Other Elements of Enterprise Data Fabric
- Real World Use Cases
 - Look at problems and solution architecture
- Summary



Data Distribution Model

Weak to strong cache consistency

- Asynchronous
- Synchronous
 - Very low latency
- Synchronous with ACK
 - Guarantee that message has reached destination
- Synchronous with App ACK
 - Guarantee that App has successfully processed message
- Synchronous with Global locking
 - Very pessimistic



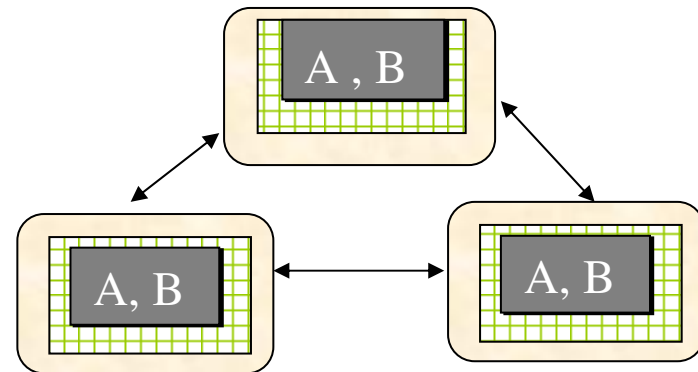
Agenda

- Challenges in clustered enterprise java applications
- Solution - Enterprise Data Fabric (EDF)
- Data Distribution Model
- **Caching Topologies**
- Other Elements of Enterprise Data Fabric
- Real World Use Cases
 - Look at problems and solution architecture
- Summary



Caching Topologies

- Replicated Topology
 - Cache Data is replicated to all members of the cluster
- Pros
 - Zero Latency Access
- Cons
 - Cost Per Update
 - Cost Per Entry



Caching Topologies

- Partitioned Topology

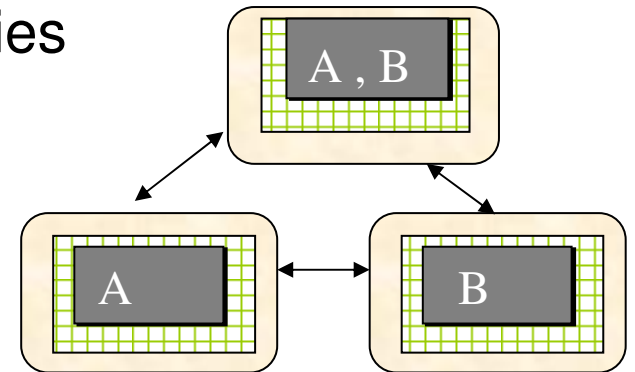
- Transparently partitions data across many nodes in the cluster
- Access is always at most one network hop
- HA preserved through redundant copies

- Pros

- Highly scalable
- Lower cost per update

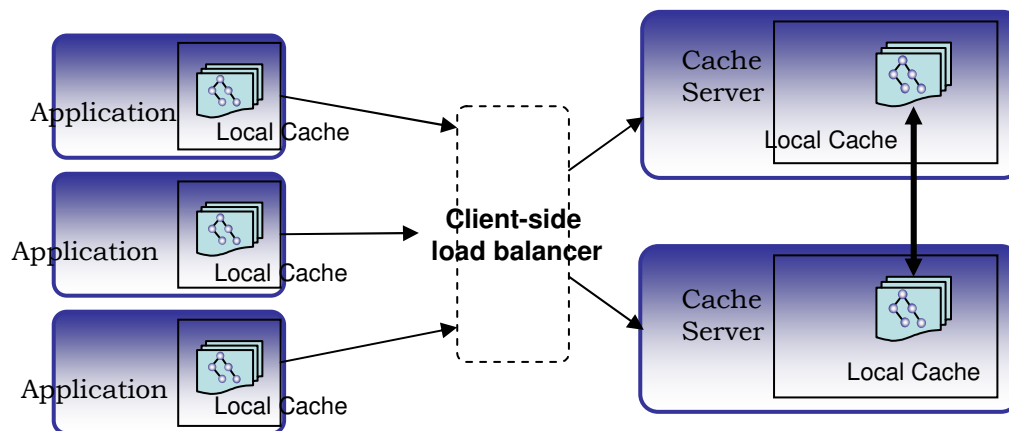
- Cons

- Cost of dynamic rebalancing of buckets
- Unexpected heap growth due to unfair partitioning



Caching Topologies

- Client side Data Caching
 - Repeated data access
 - Specific data needs of the client
 - Fabric manages consistency of data on client
 - Client polling is eliminated



Agenda

- Challenges in clustered enterprise java applications
- Solution - Enterprise Data Fabric (EDF)
- Data Distribution Model
- Caching Topologies
- **Other Elements of Enterprise Data Fabric**
- Real World Use Cases
 - Look at problems and solution architecture
- Summary



Other Elements of Enterprise Data Fabric

- Support for multiple languages
 - Java, C/C++, XML(Soap), C#
 - Manage data in different data formats
- Querying
- Distributed Transactions - JTA Compliance
- Distributed Transactions – Support JDBC/JCA compliant Resource Managers.
- Dynamic Provisioning
- Distributed data persistence

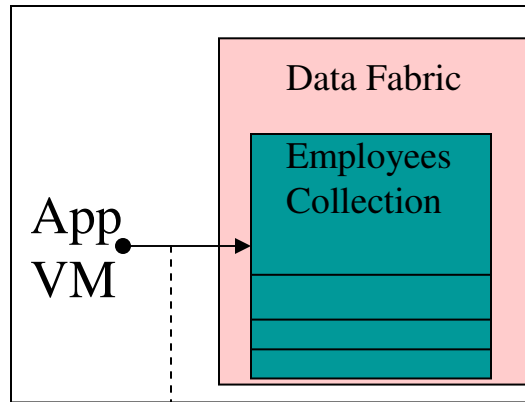


Other Elements of Enterprise Data Fabric

- Support for multiple languages
 - Java, C/C++, XML(Soap), C#
 - Manage data in different data formats
- Querying
- Distributed Transactions - JTA Compliance
- Distributed Transactions – Support JDBC/JCA compliant Resource Managers.
- Dynamic Provisioning
- Distributed data persistence



Other Elements of Enterprise Data Fabric



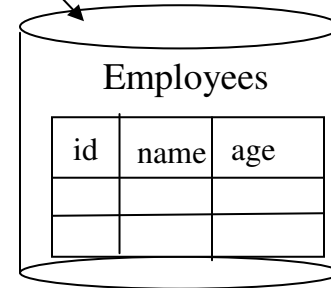
Select emp.name, emp.age
from Employees emp
where emp.age >17

OQL

App VM

SQL

Select name, age
from employees
where age > 20



Querying

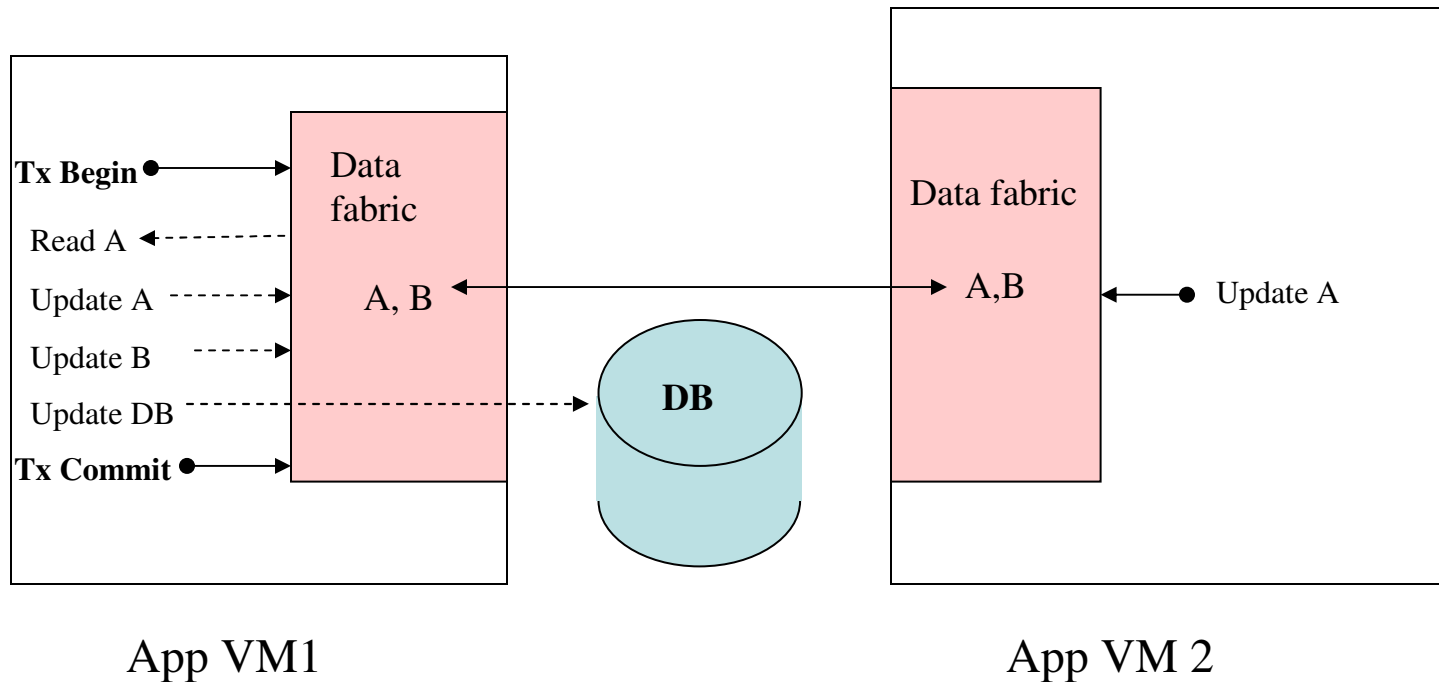


Other Elements of Enterprise Data Fabric

- Support for multiple languages
 - Java, C/C++, XML(Soap), C#
 - Manage data in different data formats
- Querying
- Distributed Transactions - JTA Compliance
- Distributed Transactions – Support JDBC/JCA compliant Resource Managers.
- Dynamic Provisioning
- Distributed data persistence



Other Elements of Enterprise Data Fabric



Transaction with EDF And DB
as TX Resources



Other Elements of Enterprise Data Fabric

- Support for multiple languages
 - Java, C/C++, XML(Soap), C#
 - Manage data in different data formats
- Querying
- Distributed Transactions - JTA Compliance
- Distributed Transactions – Support JDBC/JCA compliant Resource Managers.
- Dynamic Provisioning
- Distributed data persistence



Agenda

- Challenges in clustered enterprise java applications
- Solution - Enterprise Data Fabric (EDF)
- Data Distribution Model
- Caching Topologies
- Other Elements of Enterprise Data Fabric
- **Real World Use Cases**
 - Look at problems and solution architecture
- Summary



Use Case 1: Scalable Session State replication

- Requirements

- Sessions that never expire – Large Web Portals today
- Personalize every page accessed
 - Session state can be very large
- Support for thousands of concurrent users
 - Predictable QoS
- Session state to be managed in memory
- Session state should be highly available



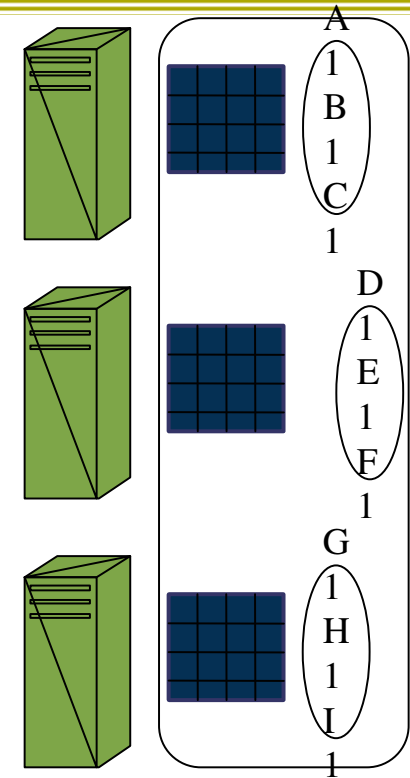
Use Case 1: Scalable Session State replication

- Issues
 - Replicating to every node in the cluster
 - Session replication slows down the entire app
 - Not designed with clustering in mind
 - HTTP session is a list of attributes
 - Session state becomes obese over time
 - In production, the CPU becomes very busy with excessive serialization
 - Session state cannot fit in server memory
 - And, server starts to experience large pauses



Use Case 1: Solution Architecture

- Move session state management out of the server VM
- Partition session state across many JVMs
- Replicate partitions to one or more nodes for HA



Use Case 2: Solution Architecture

- One logical cache with data transparently partitioned
 - Distributed hashing to remote buckets - single network hop
 - Implementation has to guarantee uniform data distribution
 - Ability to add/remove partition nodes - Dynamic rebalancing
- Look for implementations with custom serialization models
- "Delta" propagation
 - Byte code enhance domain classes participating in session state - AOP
 - Enhanced classes track changes or "delta"
 - Transmit "delta" at either *Session.putAttribute* or when Servlet thread returns

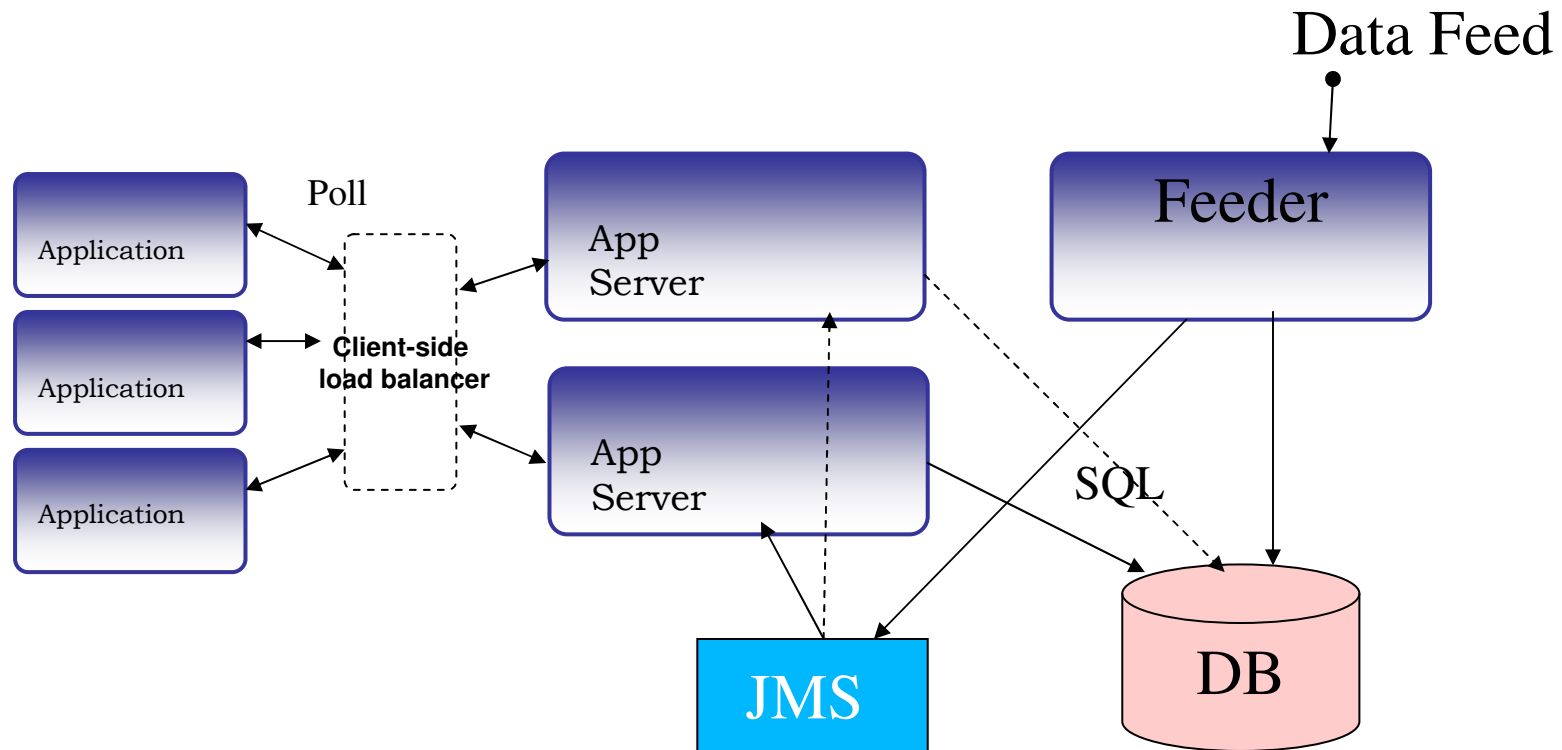


Use Case 2: Scalable Client Server Design

- Requirements
 - Real time analysis of rapidly changing data – Trading terminals
 - Latest data of those stocks which satisfy complex conditions
 - Latency < 1 second
 - High Availability of the data
 - Transparent fail over
 - Support thousands concurrent clients



Use Case 2: Existing Design



Use Case 2: Scalable Trading Application

- Issues

- Real time feed from exchange updates the Database
- Trading Clients poll the application servers causing heavy loading of servers
- Application servers firing complex SQL queries which will scan the whole tables
- Quite often the state may not change, making poll a waste

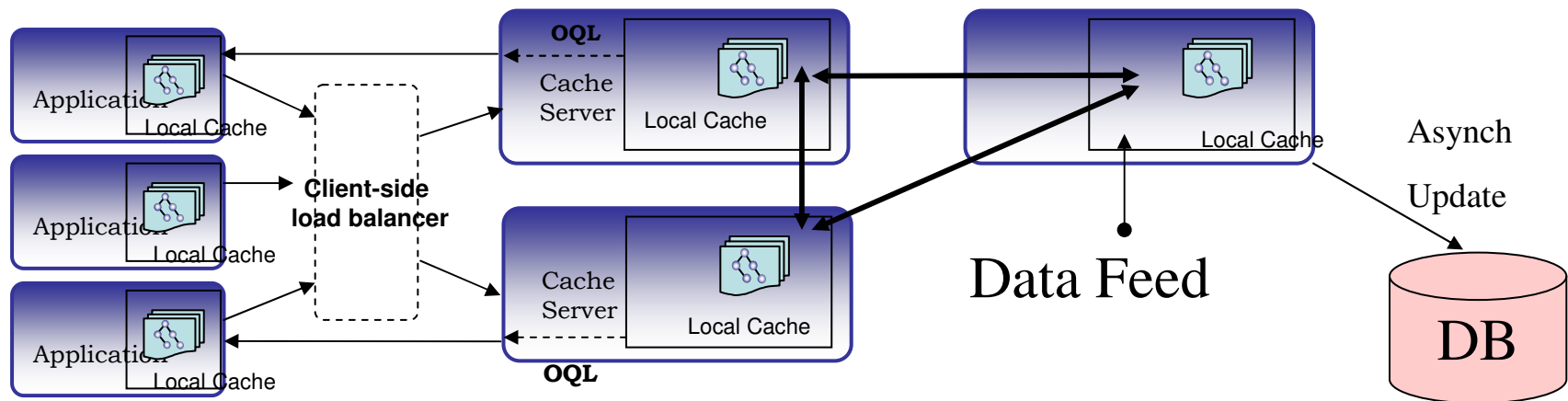


Use Case 2: Solution Architecture

- Trading clients register complex OQL with the application servers (cluster nodes)
- Real time data feed enters the EDF
- EDF moves the feed to the nodes
- Registered OQL query is fired on the modified in memory Object(Not all the objects !)
- Conditionally *delta* resultset is pushed to the client
- Copies of resultset kept on the server nodes for HA



Use Case 2: Solution Architecture



Agenda

- Challenges in clustered enterprise java applications
- Solution - Enterprise Data Fabric (EDF)
- Data Distribution Model
- Caching Topologies
- Other Elements of Enterprise Data Fabric
- Real World Use Cases
 - Look at problems and solution architecture
- **Summary**



Summary

- Enterprise Data Fabric boosts the performance and scalability of the Clustered JEE Applications
- Abstracts the application away from Data Sources
- Allow seamless exchange of data across the nodes of the cluster running dissimilar processes
- Aggregates data from disparate Data Sources



For more details

- Log on to www.enterprisedatafabric.com and www.gemstone.com
- GemStone CD in the welcome kit



Q & A



IndicThreads.com Conference On Java Technology

26th & 27 Oct. 2007 Pune, India

