

Oracle Coherence

Liran.zelkha@alunasoft.com



Aluna

- Israel's leading Java/JavaEE and SOA consulting company
- Customers:



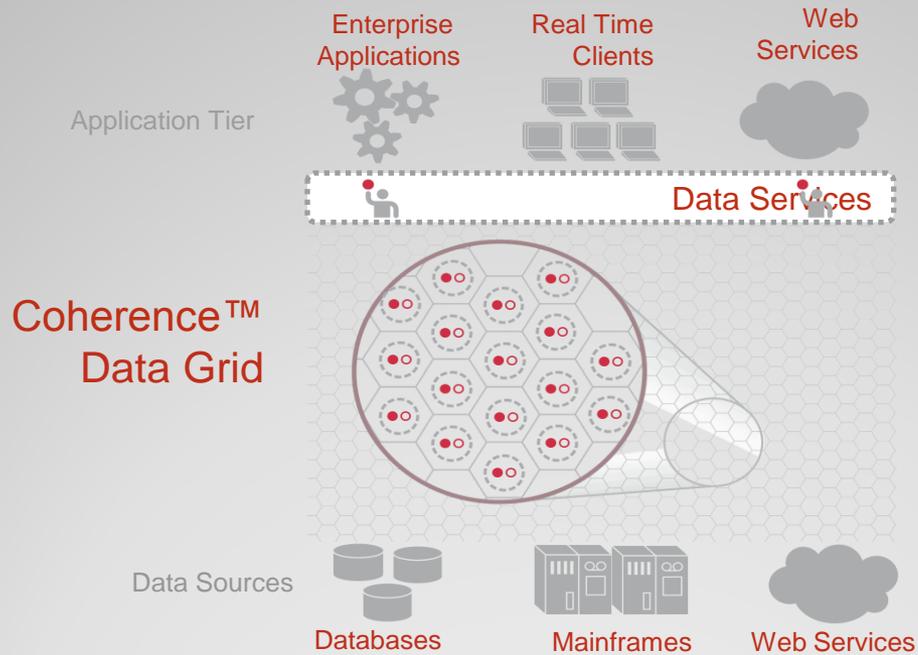
What is Oracle Coherence?

Distributed Memory Data
Management Solution
(aka: Data Grid)

How Can a Data Grid Help?

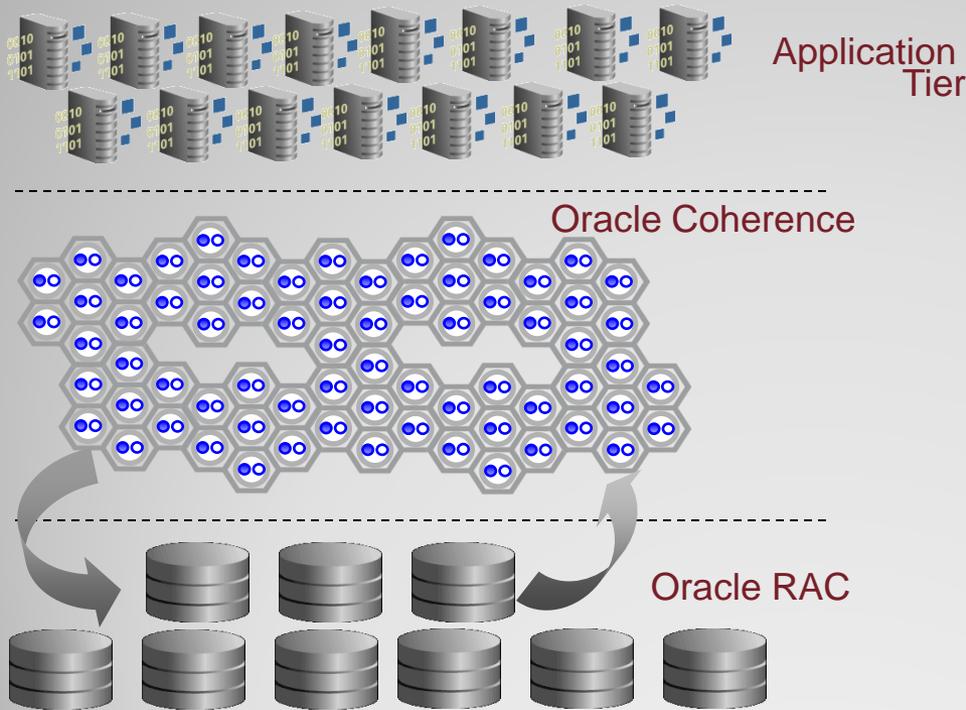
- Provides a reliable data tier with a single, consistent view of data
- Enables dynamic data capacity including fault tolerance and load balancing
- Ensures that data capacity scales with processing capacity

DataGrid View



Oracle Grid Computing: Enterprise Ready

Enterprise Application Grid
Extreme Transaction Processing XTP



- Common Shared Application Infrastructure (*Application Virtualization*)
- Data Virtualization (*Data as a Service*)
- Middle tier scale out for Grid Based OLTP
- Massive Persistent scale out with Oracle RAC

Requirements of Enterprise Data Grid

Reliable

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

Scalable

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

Universal

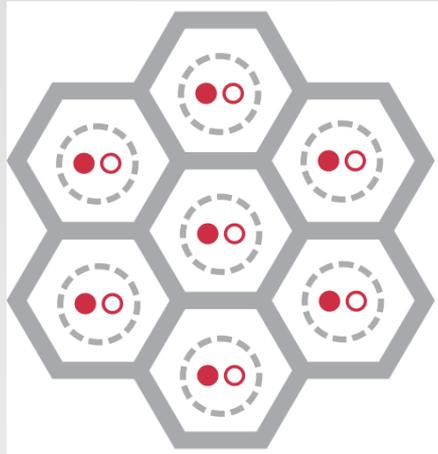
- Single view of data
- Single management view
- Simple programming model
- Any Application
- Any Data Source

Data

- Data Caching
- Analytics
- Transaction Processing
- Event Processing

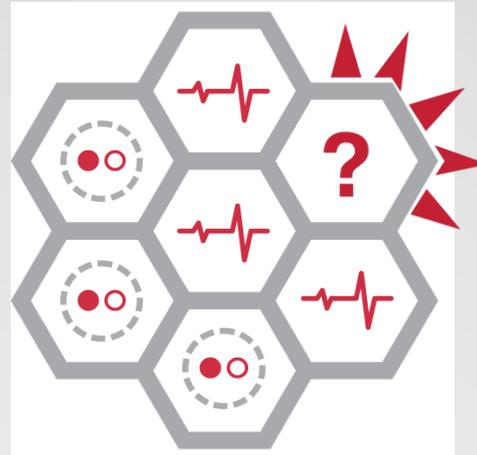
How Does Coherence Work?

- Cluster of nodes holding % of primary data locally
- Back-up of primary data is distributed across all other nodes
- Logical view of all data from any node



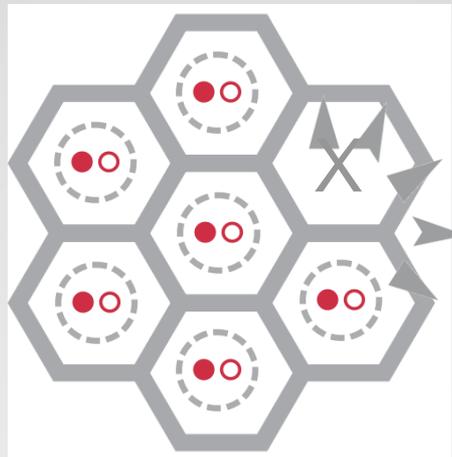
How Does Coherence Work?

- All nodes verify health of each other
- In the event a node is unhealthy, other nodes diagnose state



How Does Coherence Work?

- Unhealthy node isolated from cluster
- Remaining nodes redistribute primary and back-up responsibilities to healthy nodes

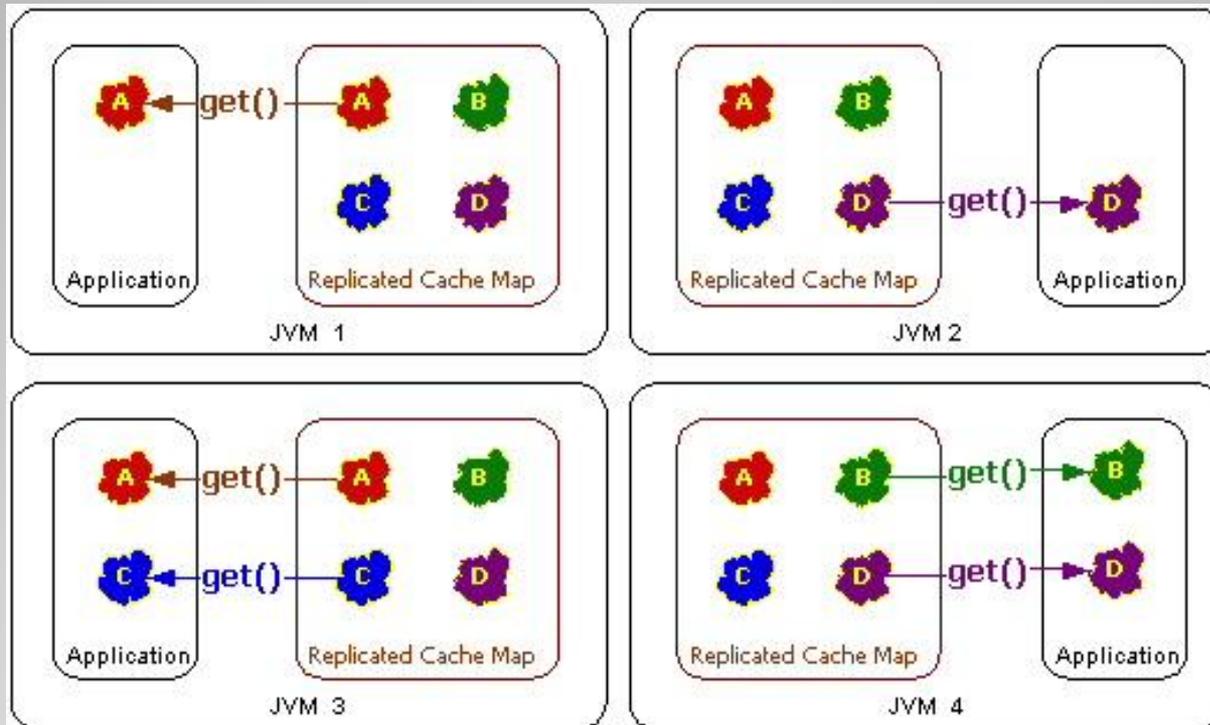


Customer Scenarios

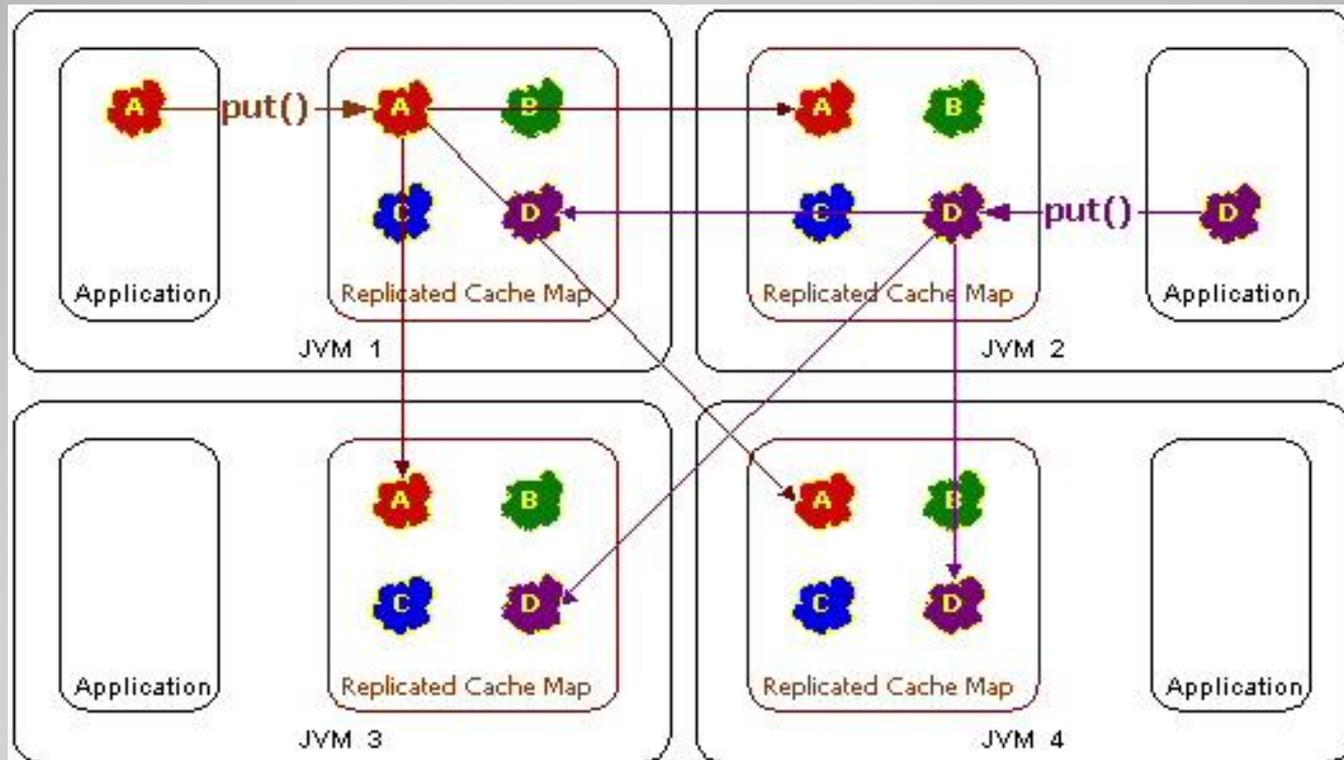
- Caching
 - Applications request data from the Data Grid rather than backend data sources
- Analytics
 - Applications ask the Data Grid questions from simple queries to advanced scenario modeling
- Transactions
 - Data Grid acts as a transactional System of Record, hosting data and business logic
- Events
 - Automated processing based on event

Demo

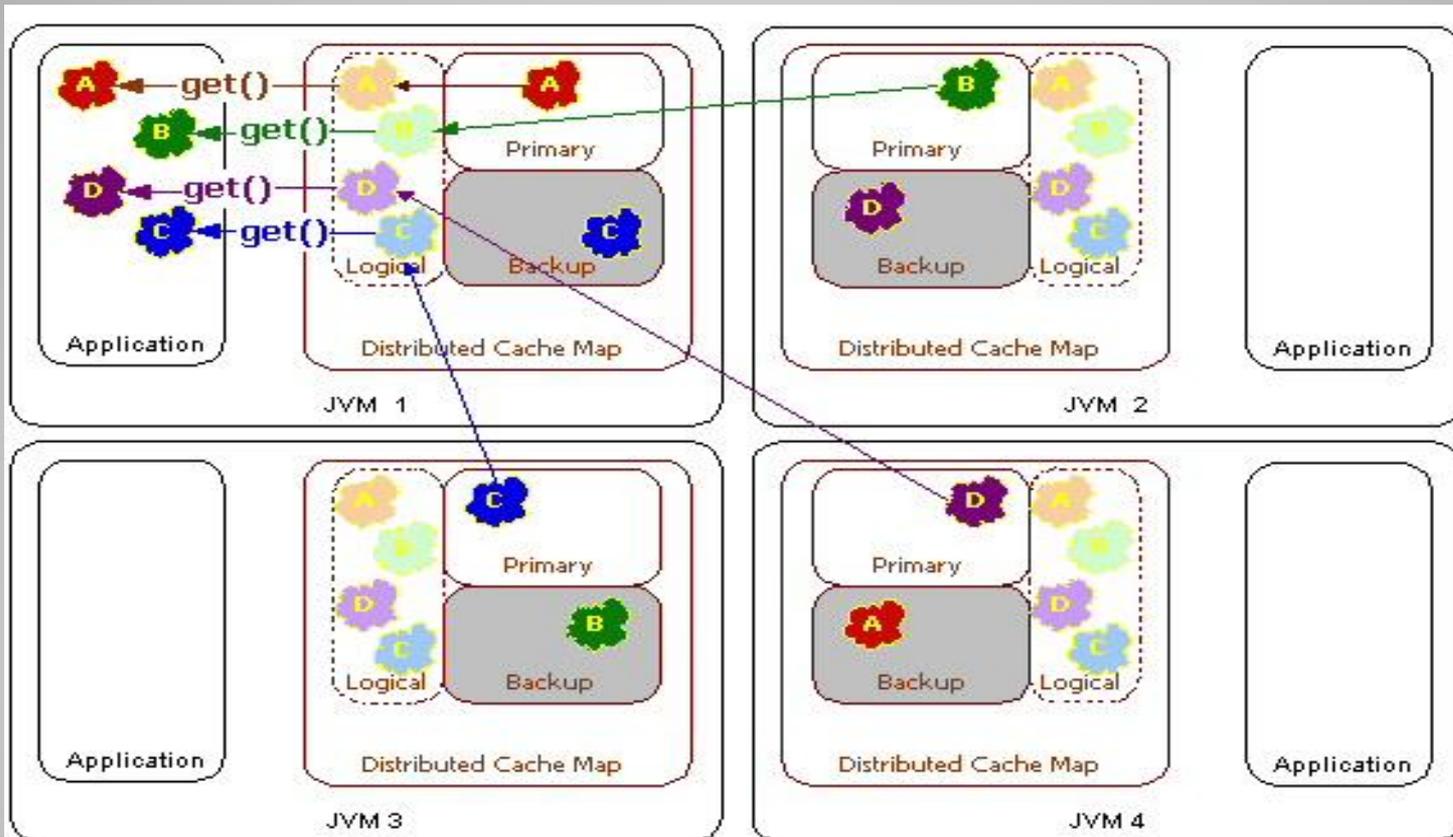
Topology #1 - Replicated Cache



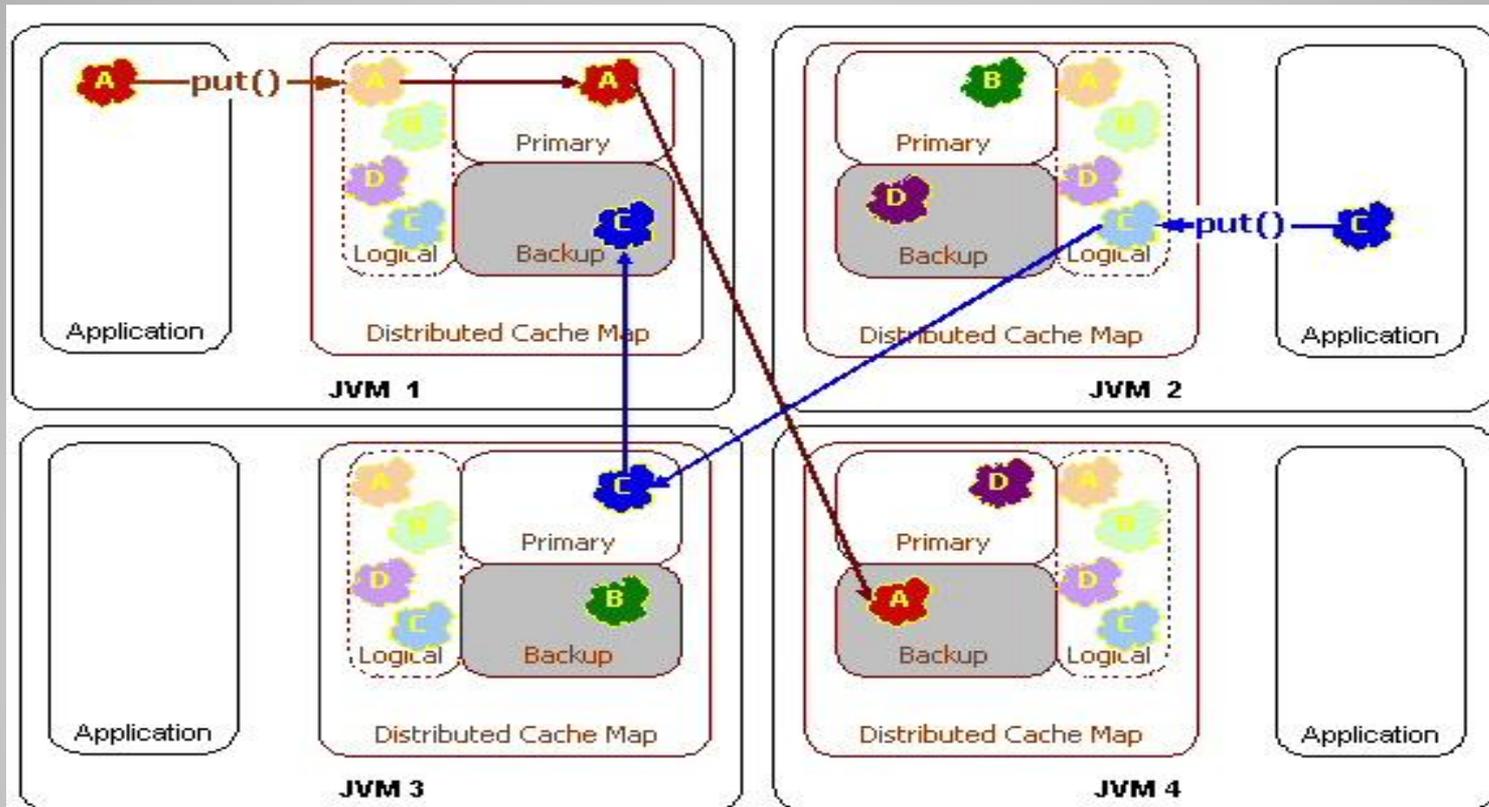
Topology #1 - Replicated Cache



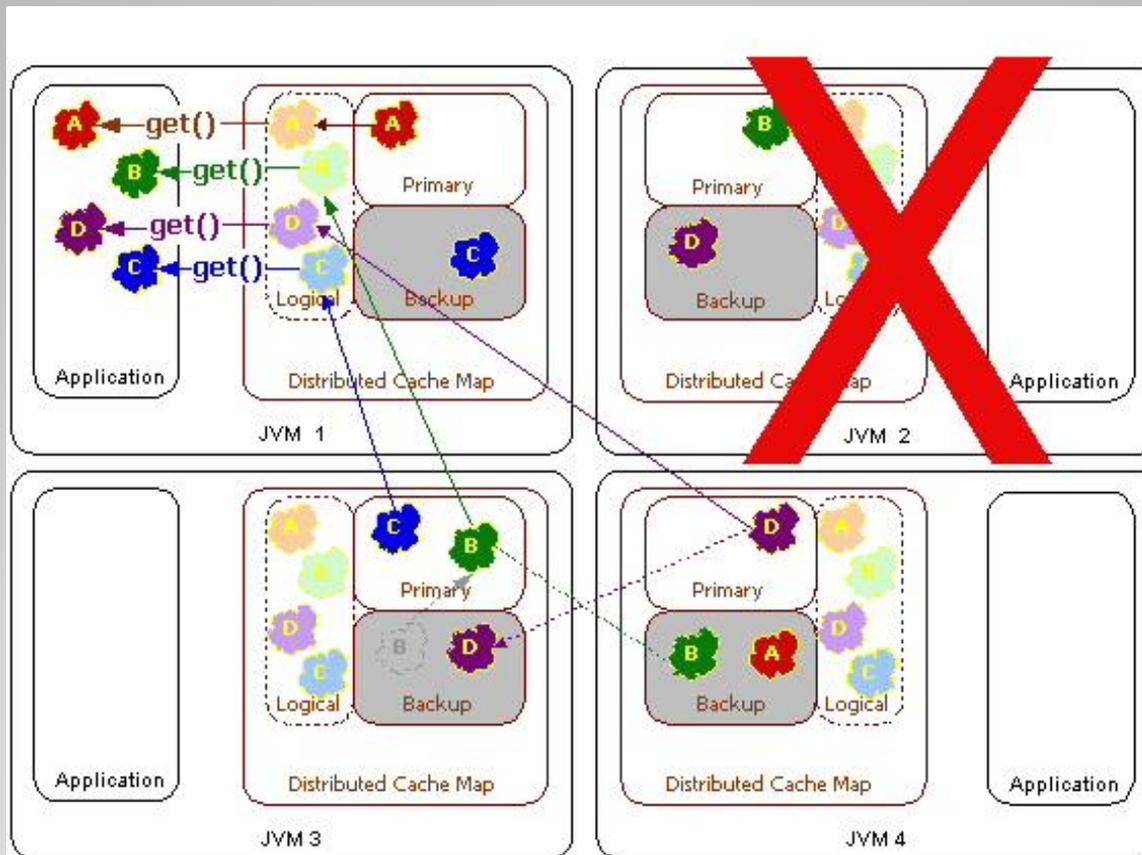
Topology #2 - Partitioned Cache



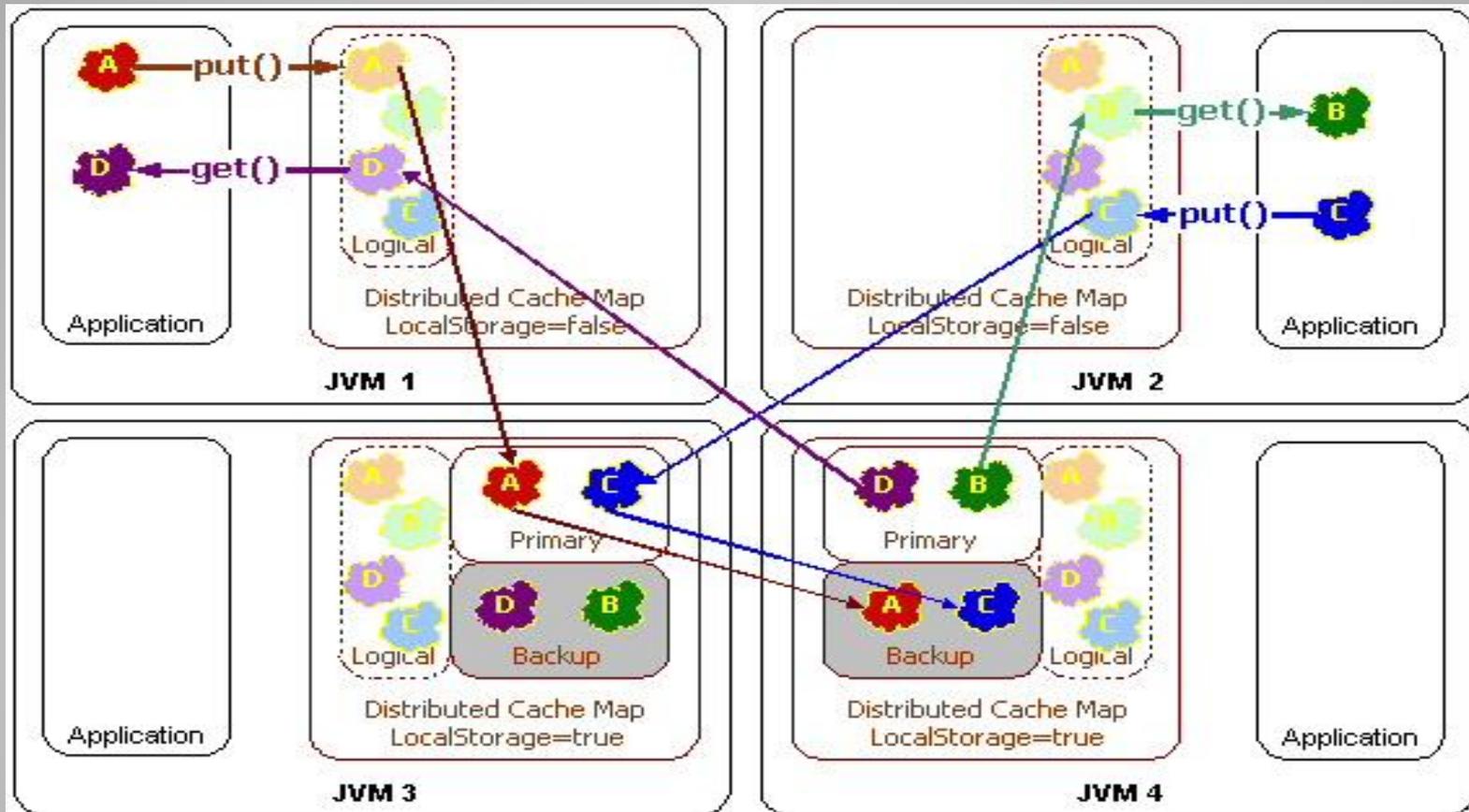
Topology #2 - Guaranteed Cluster Resiliency



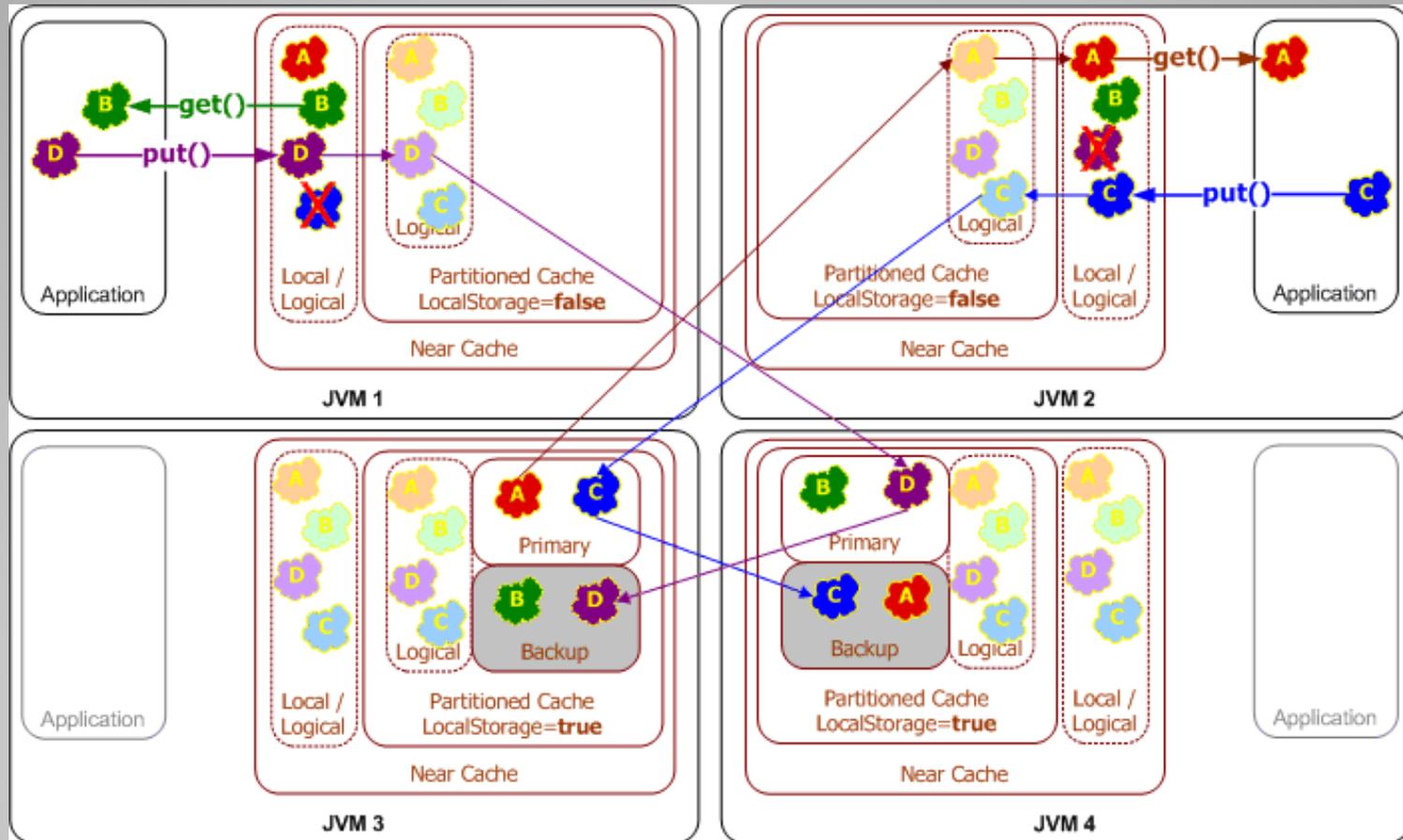
Topology #2 - Partitioned Failover



Topology #2a – Cache Client/Cache Server



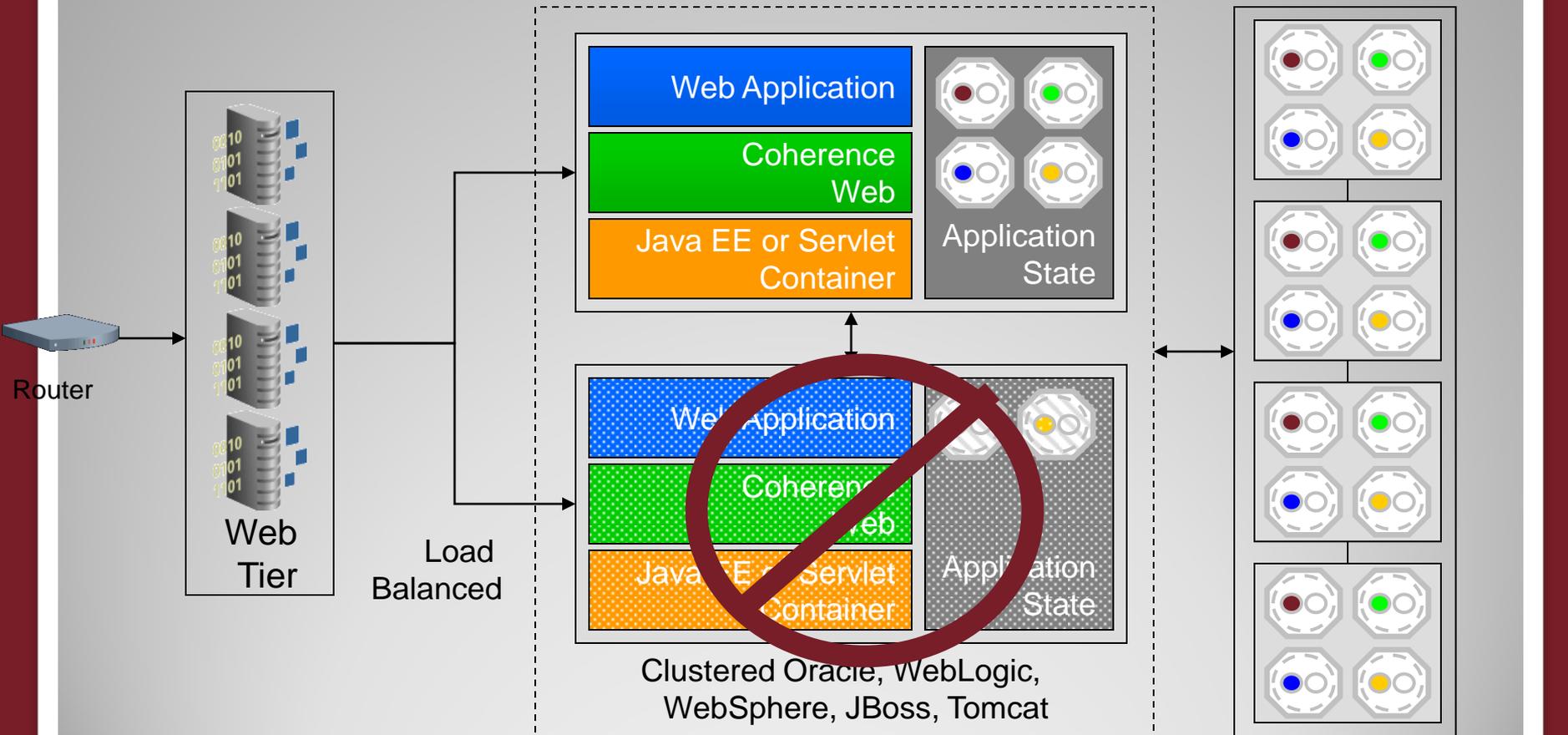
Topology #3 - Near Cache



Use Case: Coherence*Web

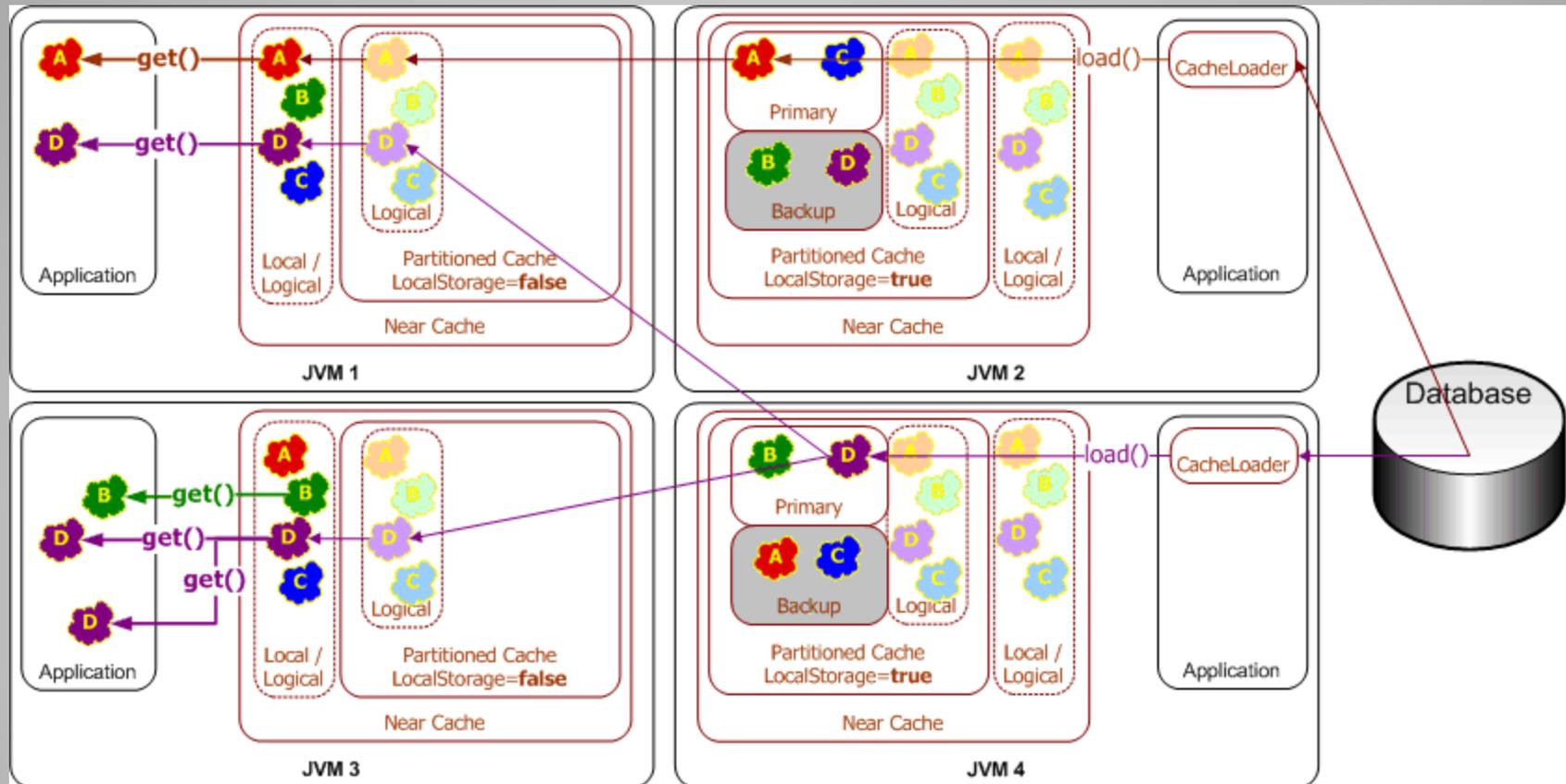
- Coherence*Web is an HTTP session-management module (built-in feature of Coherence)
- Supports a wide range of application servers.
- Does not require any changes to the application.
- Coherence*Web uses the NearCache technology to provide fully fault-tolerant caching, with almost unlimited scalability (to several hundred cluster nodes without issue).
- Heterogeneous applications running on mixed hardware/OS/application servers can share common user session data. This dramatically simplifies supporting Single-Sign-On across applications.

Coherence*Web: Session State Management

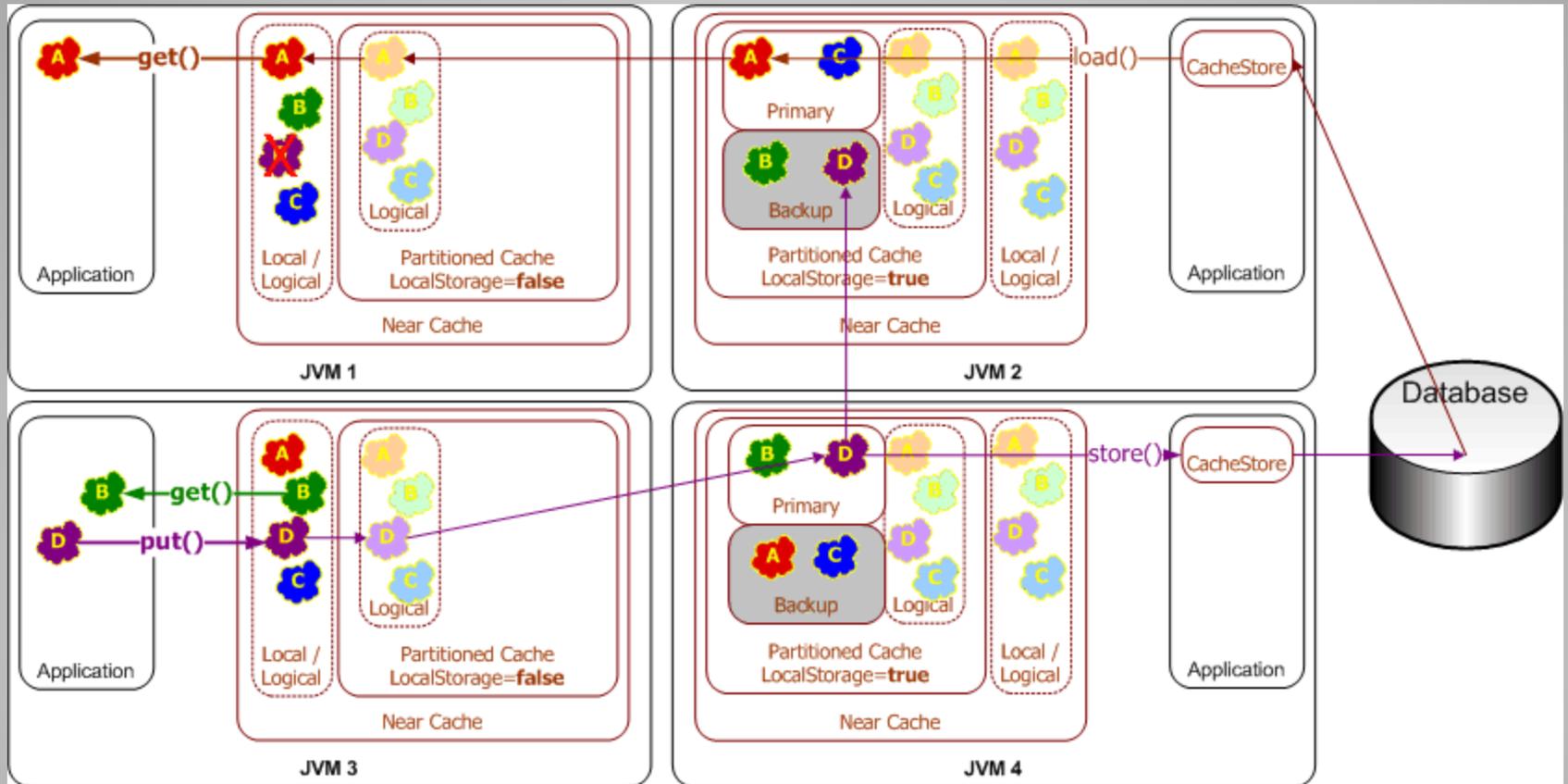


In Memory Coherence Data
Grid for Session State

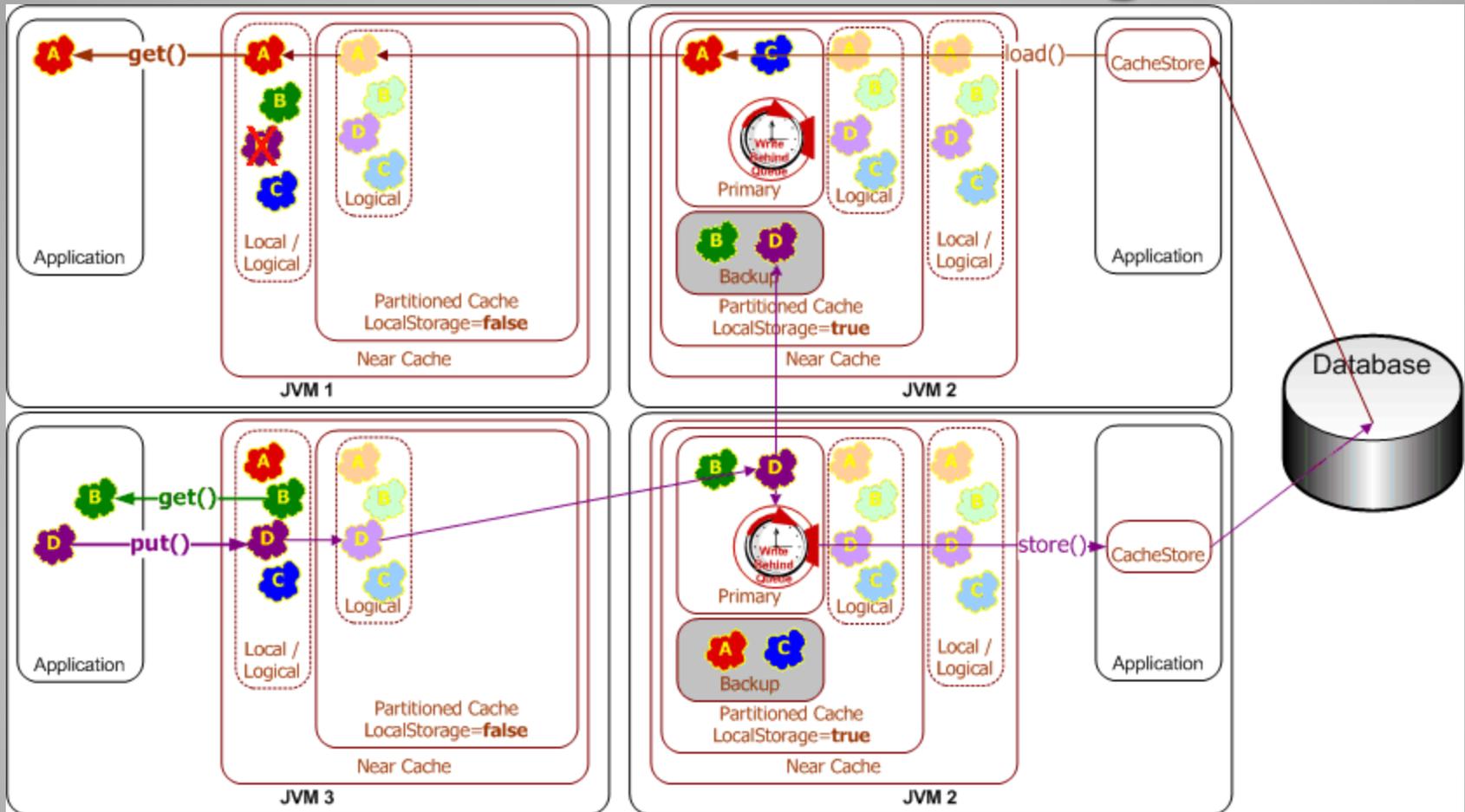
Read-Through Caching



Write-Through Caching



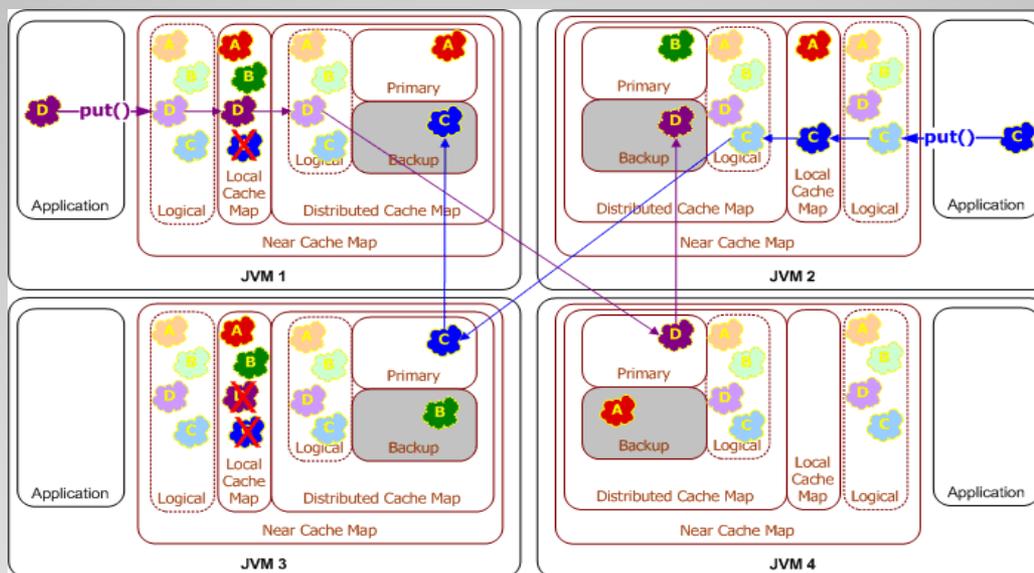
Write-Behind Caching



Transaction

- Implicit: Queuing of operations
 - Virtual queue & thread per entry
- Explicit: Pessimistic locking
 - Grid-Wide Mutex
- Transactions: Unit of work management
 - Both optimistic and pessimistic transactions
 - Isolation levels from read-committed through serializable
 - Integrated with JTA

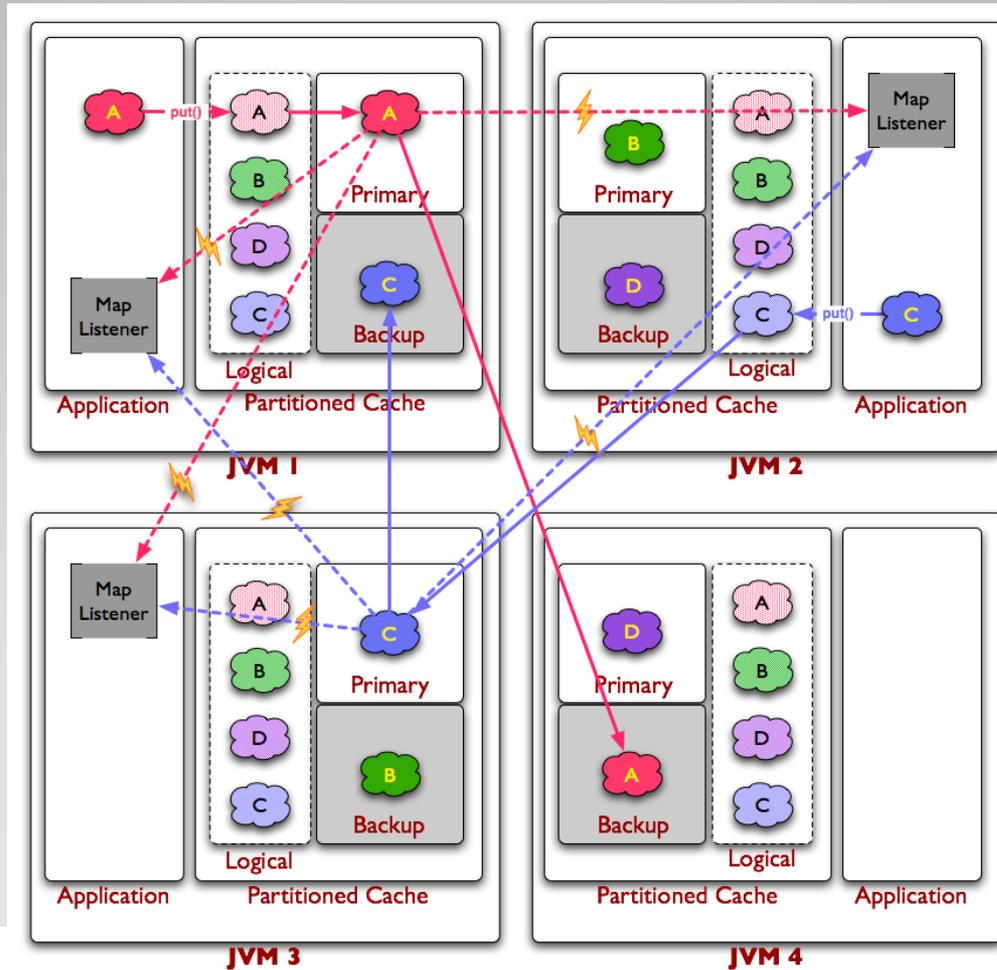
Transaction



Events

- Universal: All data sets provide events, regardless of the topology.
- Distributed: The events are always delivered efficiently to the interested listeners.
 - Regardless of originating node
- Flexible:
 - Listen to entire data sets, specific identities, and even to queries!
 - Provides “before” and “after” state
 - Both sync and async event models

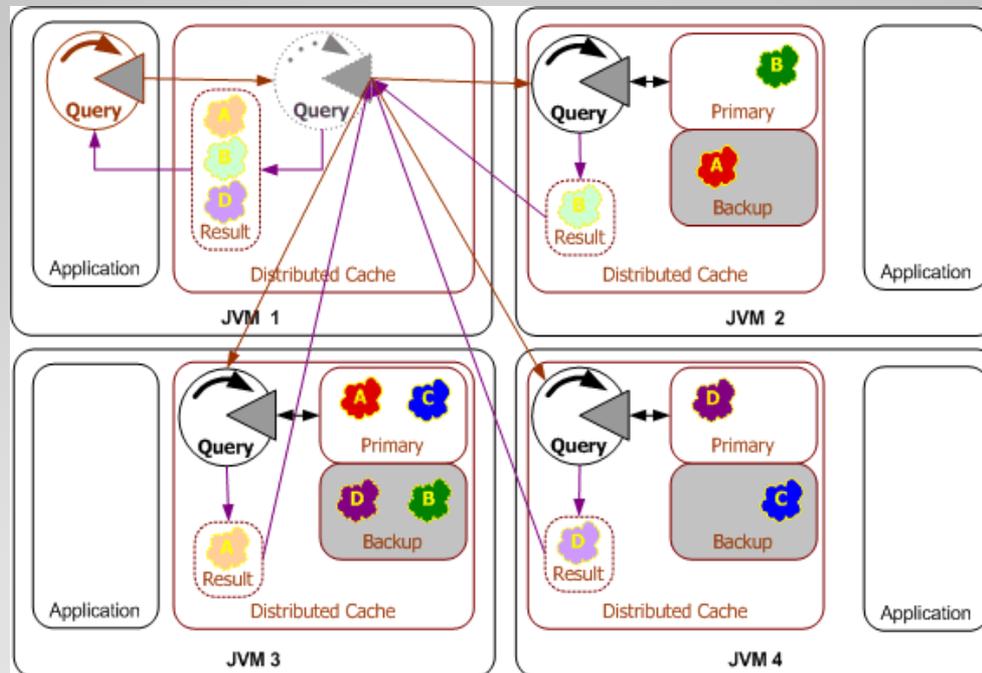
Events



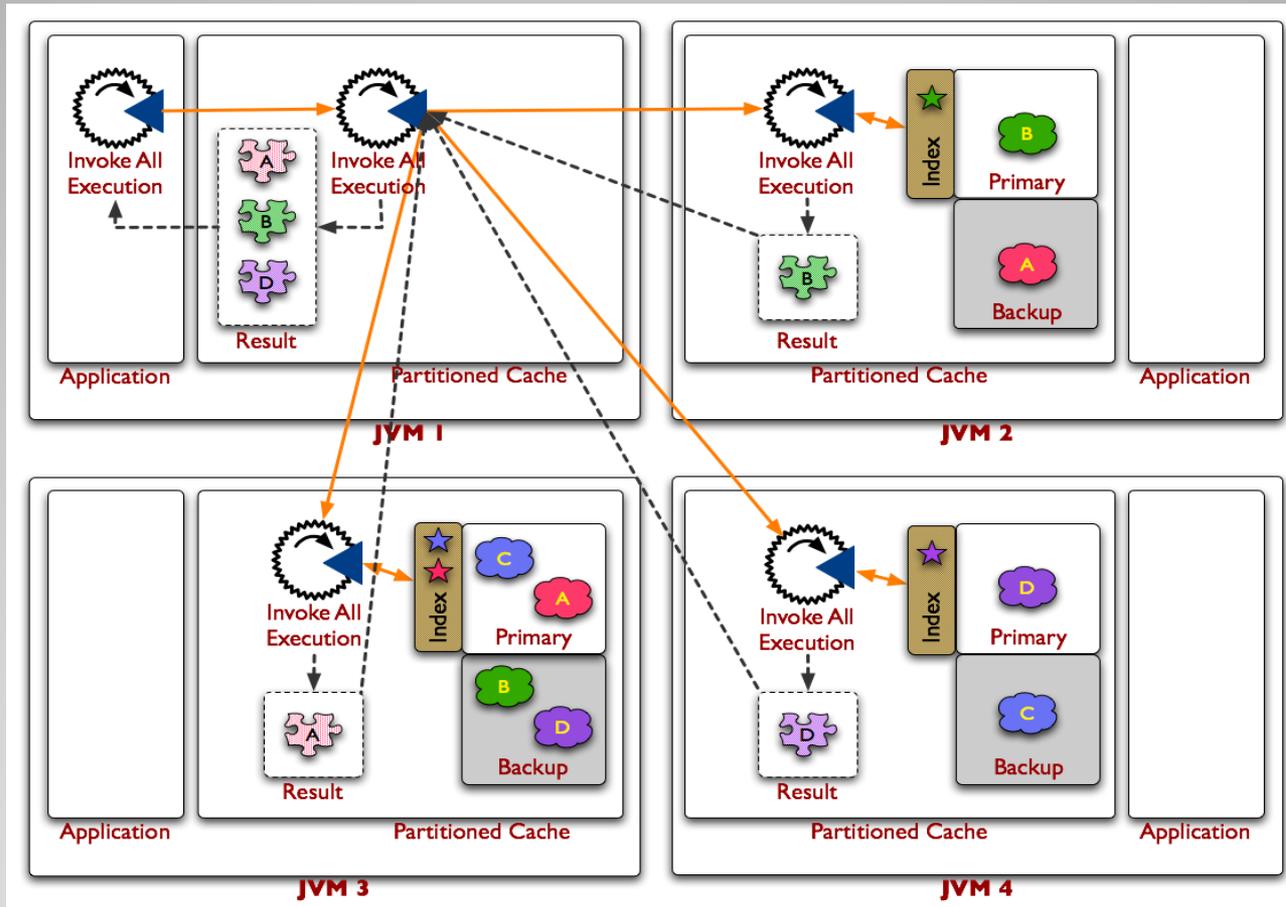
Query

- **Parallel Query:** A query is performed in parallel across the Data Grid, using indexing and a iterative Cost Based Optimizer.
 - Customizable predicates
 - Custom indexes
 - Custom aggregators
- **Continuous Query:** Combines a query with events to provide a local materialized view.
 - Result is up-to-date in real-time
 - Like the Near Topology, but it always contains the desired data

Query



InvocableMap – Server Side Processing



Coherence*Extend

- Supports “fat client” real-time applications such as trading desks, as well as other server tiers
- WAN support
- Connection to the cluster is over TCP
- Continuous query can be used to maintain real-time query results on the desktop!

Tangosol Cluster Management Protocol (TCMP)

- Coherence's own protocol between cluster members
 - TCMP utilizes UDP
 - Massively scalable
 - Asynchronous
 - Point-to-point
- UDP Multicast is used for:
 - New JVMs to join the cluster automatically
 - Maintaining cluster membership
 - Multicast is not required; it may be disabled with Well Known Addresses (WKA)
- UDP Unicast is used for most communication
 - Very fast and scalable
 - TCMP guarantees packet order and delivery
 - TCP/IP connections do not need to be maintained

Clustering is about Consensus!

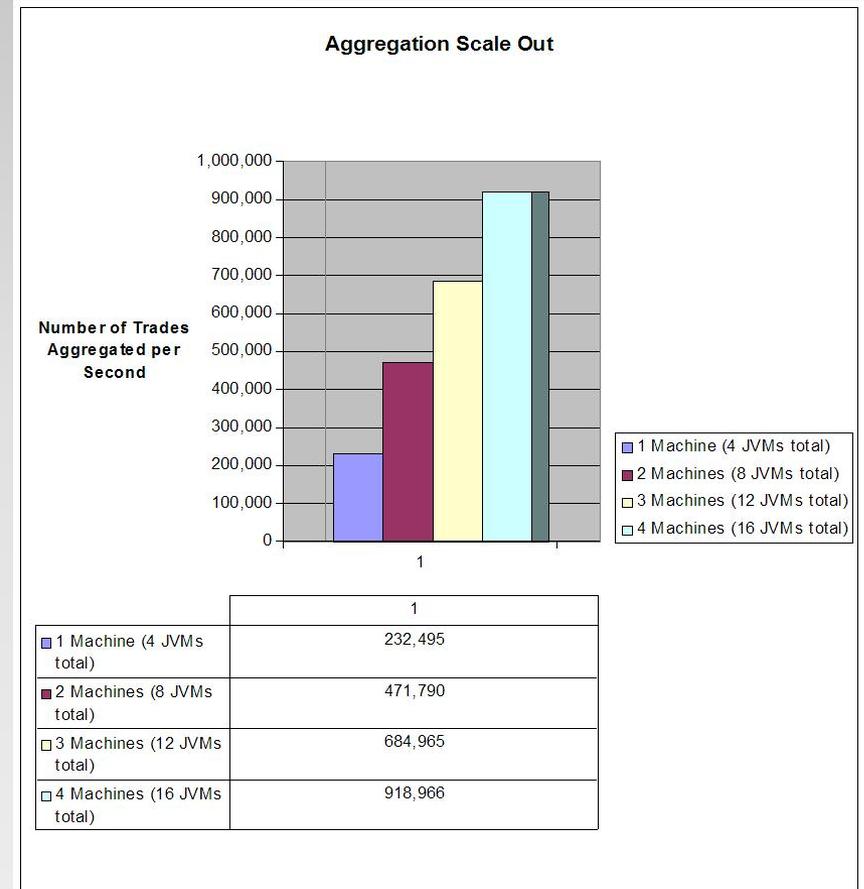
- Oracle Coherence Clustering is very different!
- Goal:
 - Maintain Cluster Membership Consensus all times
 - Do it as fast as physically possible
 - Do it without a single point of failure or registry of members
 - Ensure all members have the same responsibility and work together to maintain consensus
 - Ensure that no voting occurs to determine membership

Clustering is about Consensus!

- Why: If all members are always known...
- We can partition / load balance Data & Services
- We don't need to hold TCP/IP connections open (resource intensive)
- Any member can "talk" directly with any other member (peer-to-peer)
- The cluster can dynamically (while running) scale to any size

Benchmarking Coherence

- Aggregation (DoubleSum) of Trade objects
- Scale out testing on Dual 2.3GHz PowerPC G5 Xserve
- Use of on index for direct access
- if you need to achieve 1,837,932 trade aggregations per second all that is required is to start 16 more cache servers across four more machines.



Coherence Management and Monitoring

- Management Features
 - Coherence provides standard JMX APIs
 - Cluster-wide JMX: Ability to monitor and manage the entire cluster from any node
 - Customizable web-based console
 - Does not require an mBean server or any JMX libraries on managed nodes
 - Support custom application mBeans
 - Support for Coherence*Web