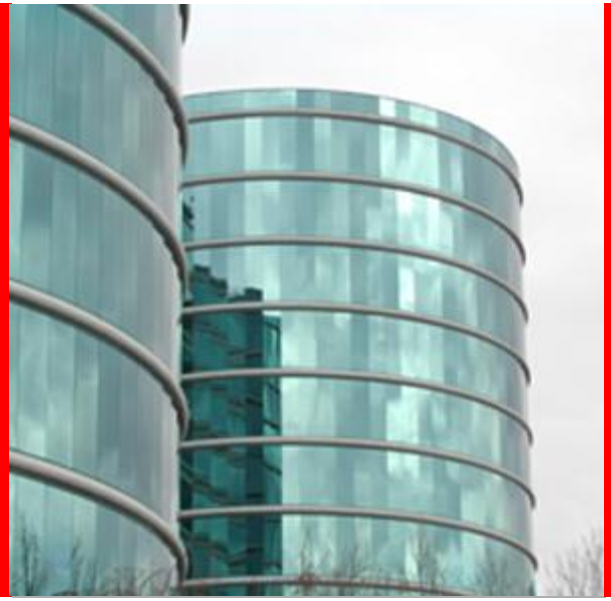


ORACLE®

Getting Coherence: Introduction to Data Grids
Jfokus Conference, 28 January 2009

Cameron Purdy
Vice President of Development

Speaker

- Cameron Purdy is Vice President of Development at Oracle. Prior to joining Oracle, Mr. Purdy was the CEO of Tangosol, whose revolutionary Coherence Data Grid product provides reliable and scalable data management across the enterprise.
- Mr. Purdy has over ten years of experience with Java™ and Java-related technology; he regularly participates in industry standards development and is a specification lead for the Java Community Process.

Disclaimer

Religious application of principles learned from a presentation attended at an industry tradeshow is not a substitute for common sense.

Concepts

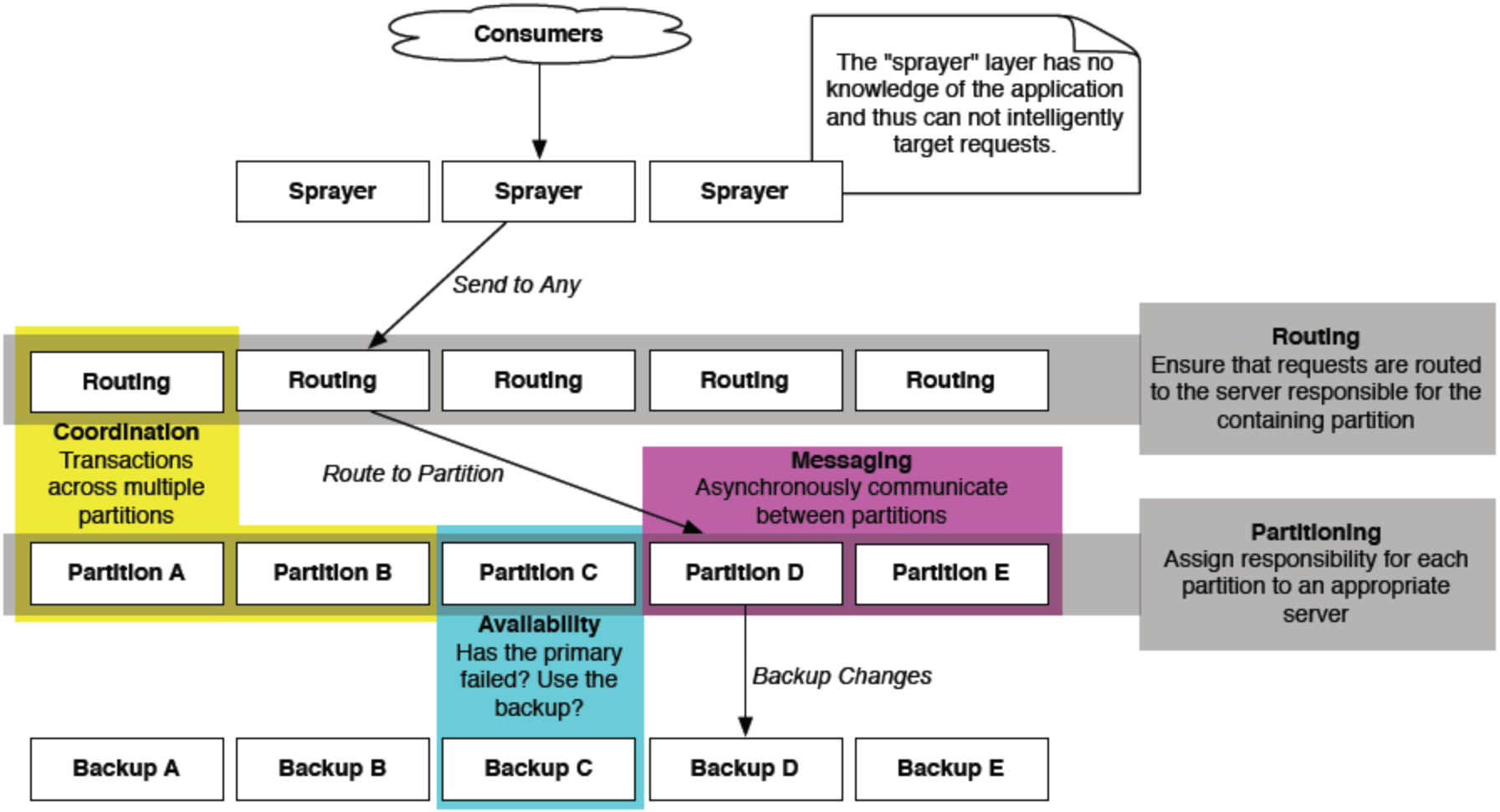
“But if *I* ran IT,”
Said young Gerald McGrew,
“I’d make a few upgrades.
That’s just what I’d do . . .”

Programming Models vs Messaging

- Efficient Programming Models
  - Single threaded
  - Reliable
  - In order
  - Zero latency invocation
  - Once and only once processing

- Messaging makes you pay
  - Ordering limits scalability
  - Reliability adds costs
  - Introduces latency
  - Once-and-only-once is hard

The Five Patterns of Stateful Scale-Out



Data Grid Introduction

The Oracle Coherence In-Memory Data Grid is a data management system for application objects that are shared across multiple servers, require low response time, very high throughput, predictable scalability, continuous availability and information reliability.

Data Grid

- Who: Oracle Coherence, IBM WebSphere XS
  - Also Gemstone Gemfire, Gigaspaces, ScaleOut Software
- What: Using a grid infrastructure to host both reference and transactional data, eliminating data bottlenecks and improving performance and efficiency
- Where: Inside the data center
- When: Usually 24x7x366, but sometimes only for
- How: Using clustering technology
  - Partitioning for scale (and reliability)
  - Replicating for availability (and performance)

Data Grid: Capability Summary

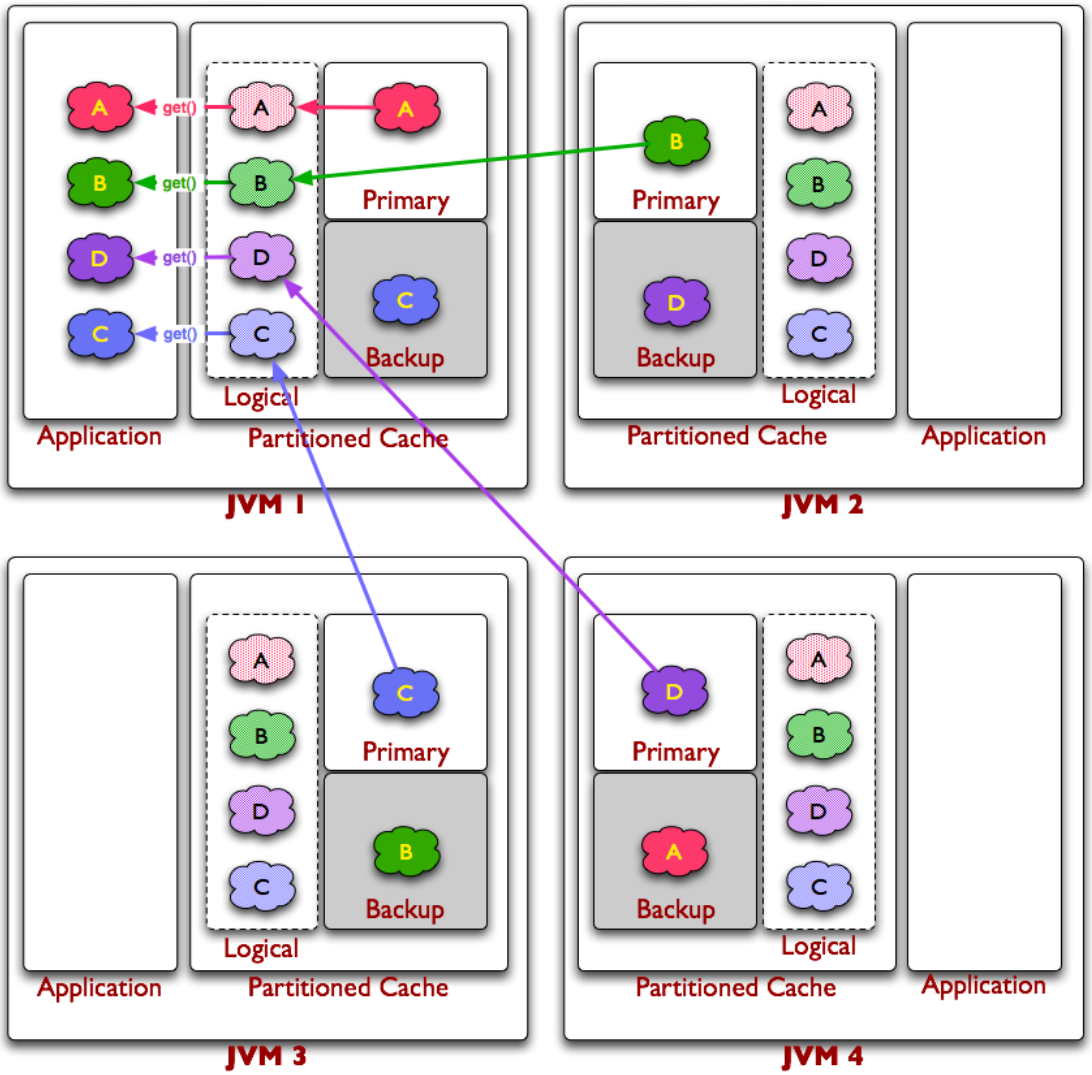
- A Data Grid combines data management with data processing in a scale-out environment
  - Some or all of the servers in the grid are responsible for reliably managing live information
  - Any or all of the servers in the grid are able to simultaneously access and manipulate a shared, consistent view of that information
  - Processing power far exceeds aggregate network bandwidth, so data access and manipulation must be parallelized and localized within the grid
  - Data locality is both dynamic and transparent

Oracle Coherence:

Technical Illustrations

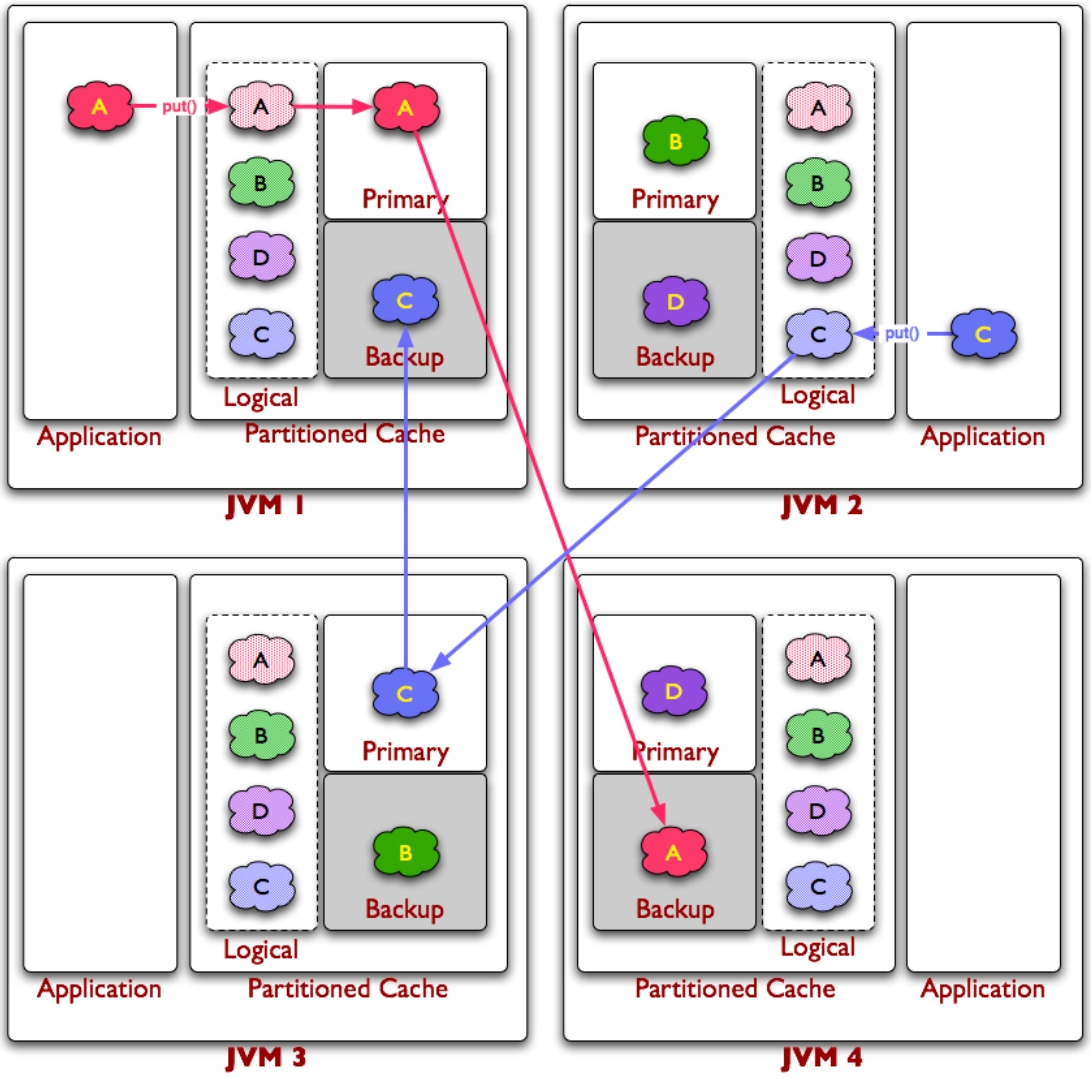
Partitioned Topology : Data Access

- Data spread and backed up across Members
- Transparent to developer
- Members have access to all Data
- All Data locations are known – no lookup & no registry!



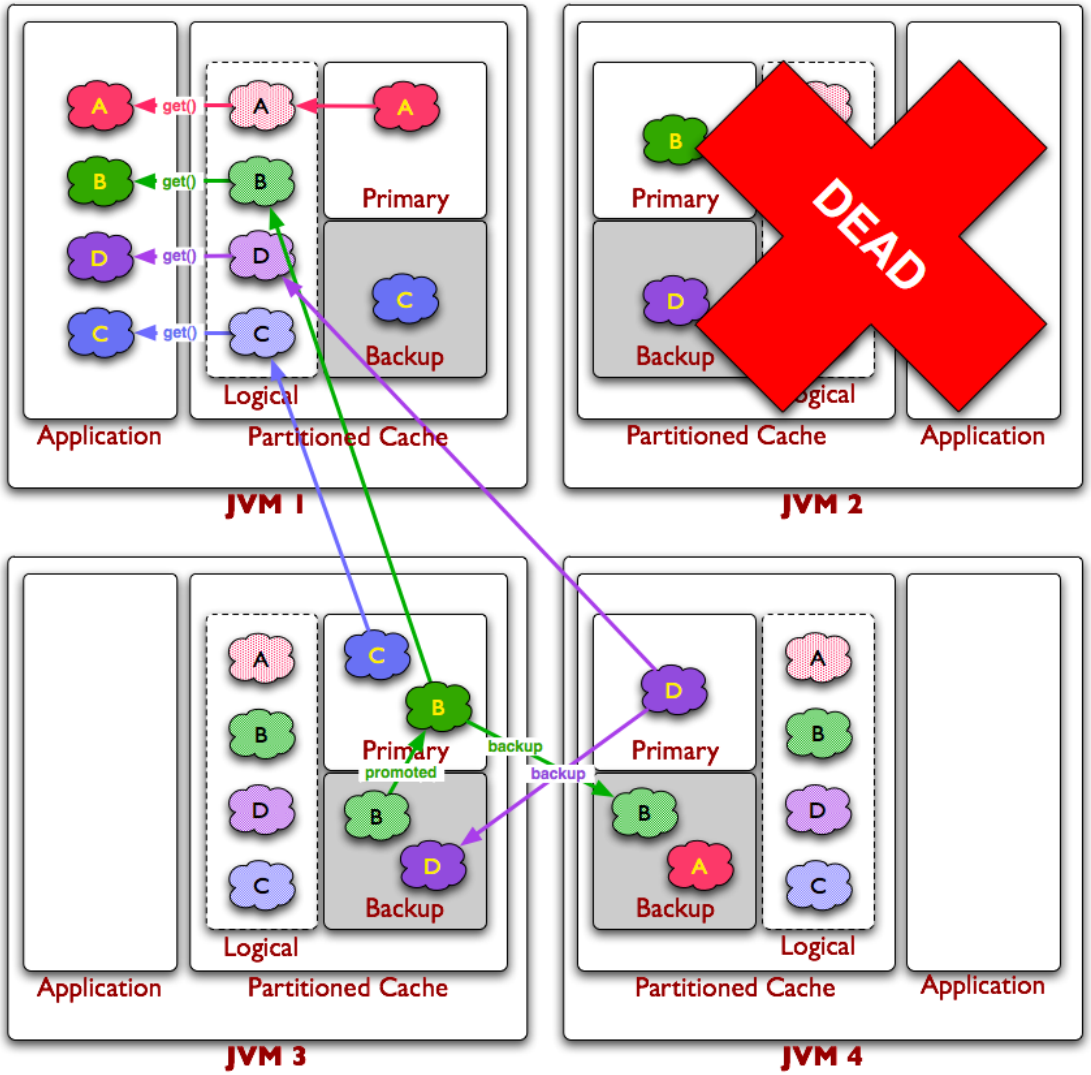
Partitioned Topology : Data Update

- Synchronous Update
- Avoids potential Data Loss & Corruption
- Predictable Performance
- Backup Partitions are partitioned away from Primaries for resilience
- No engineering requirement to setup Primaries or Backups
- Automatically and Dynamically Managed



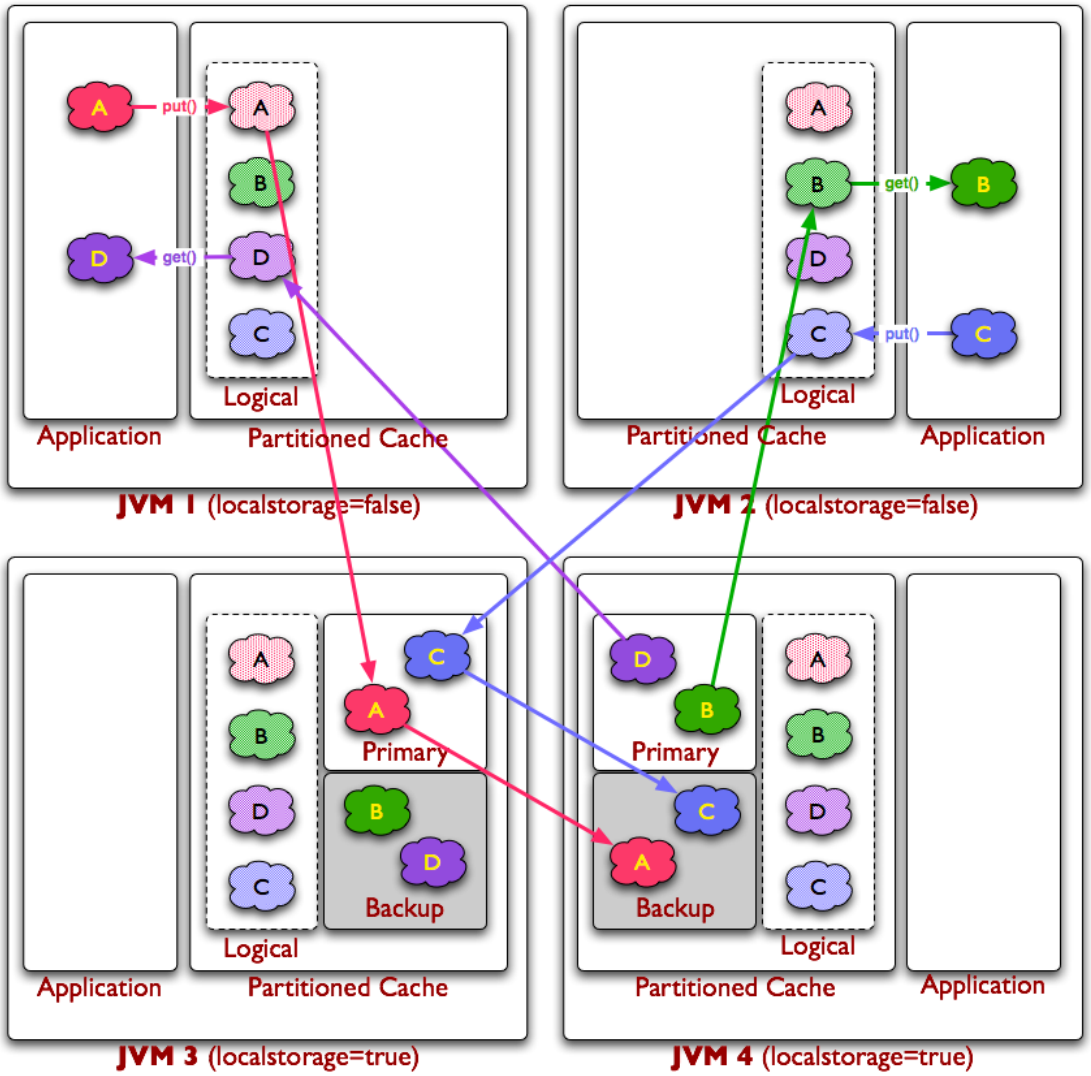
Partitioned Topology : Recovery

- Membership changes (new members added or members leaving)
- Other members, using consensus, recover and repartition automatically
- No in-flight operations lost, no availability gap!
- Some latencies (due to higher priority of asynchronous recovery)
- Information Reliability & Continuous Availability are the priorities



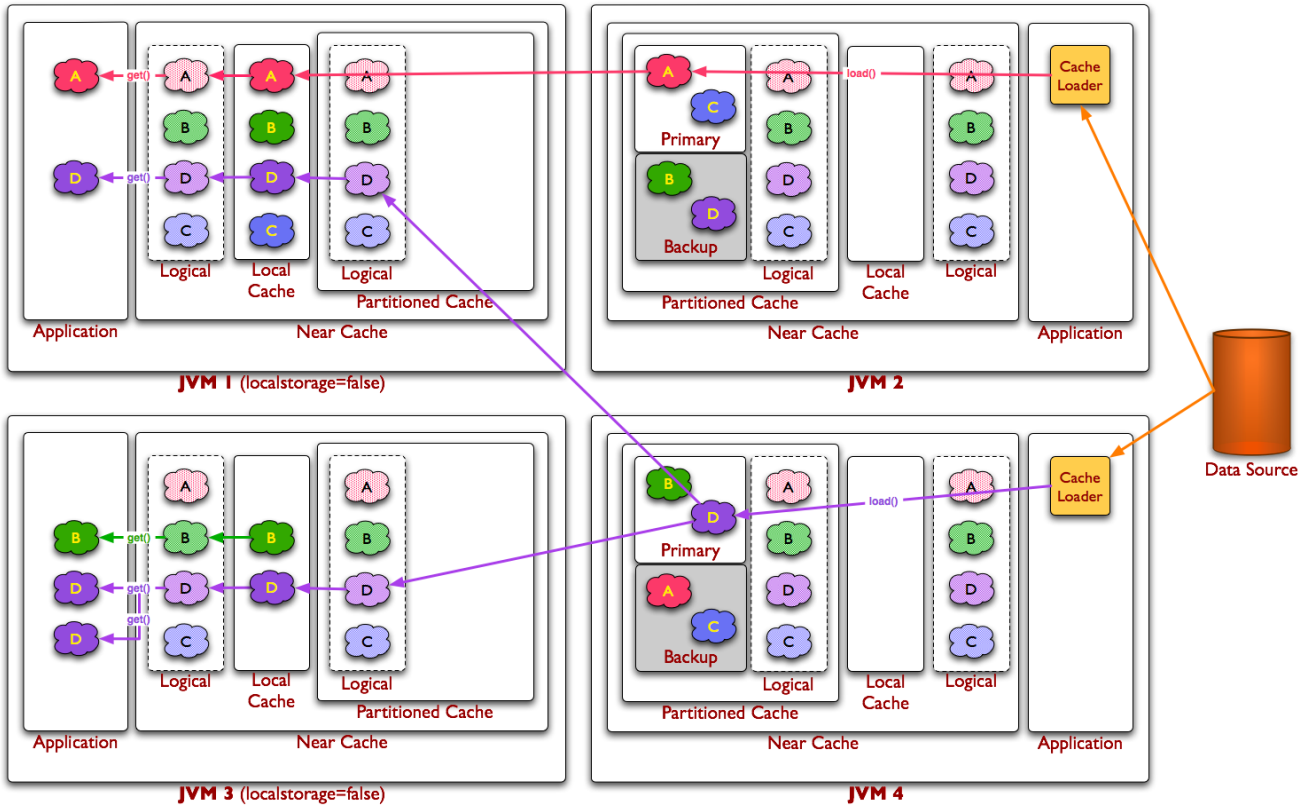
Partitioned Topology : Local Storage

- Some members are used to manage data
- Other members are temporary in a cluster, or do not have memory to spare for managing data
  - They should not cause repartitioning
- Specialization of roles within a Data Grid: Clients and Servers



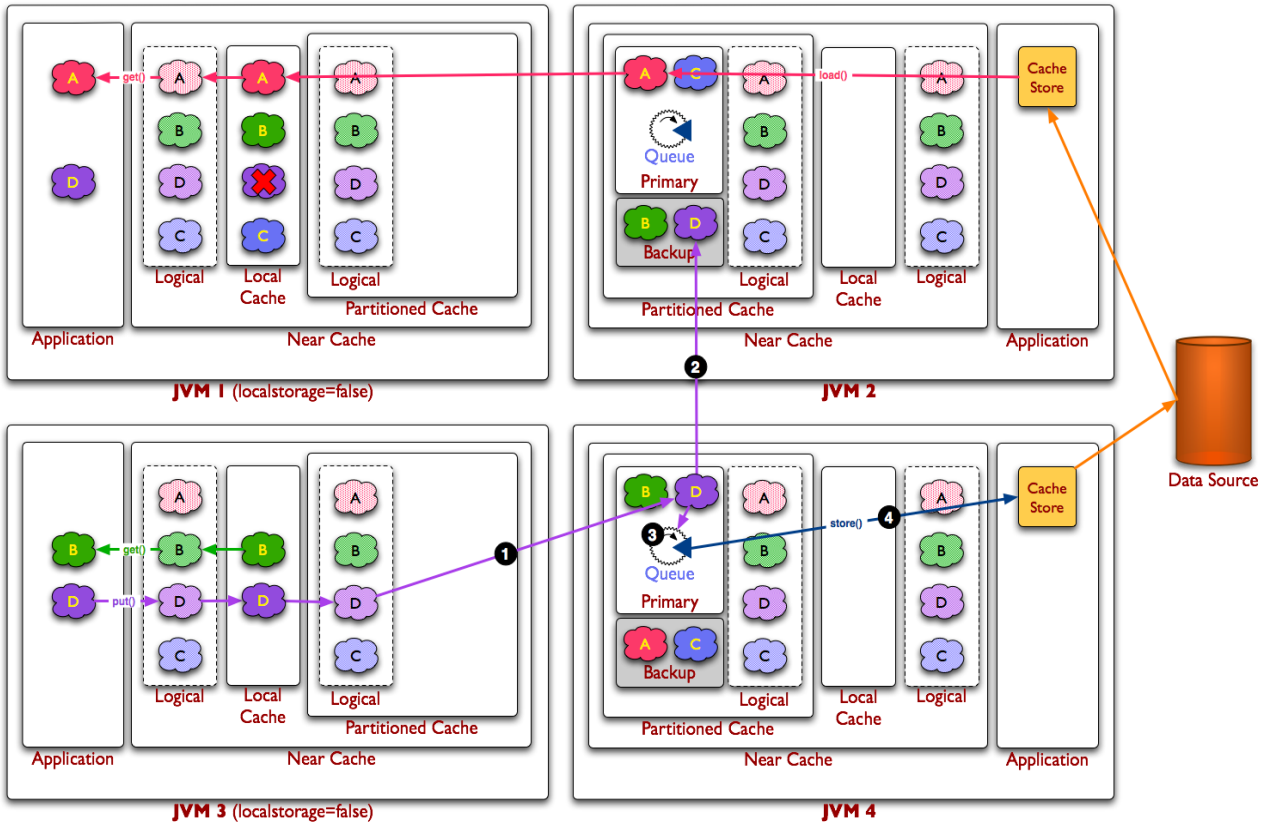
Read-Through & Write-Through

- Access to the data sources go through the Data Grid.
- Read and write operations are always managed by the node that owns the data within the Data Grid.
- Concurrent accesses are combined, greatly reducing database load.
- Write-Through keeps the in-memory data and the database in sync.

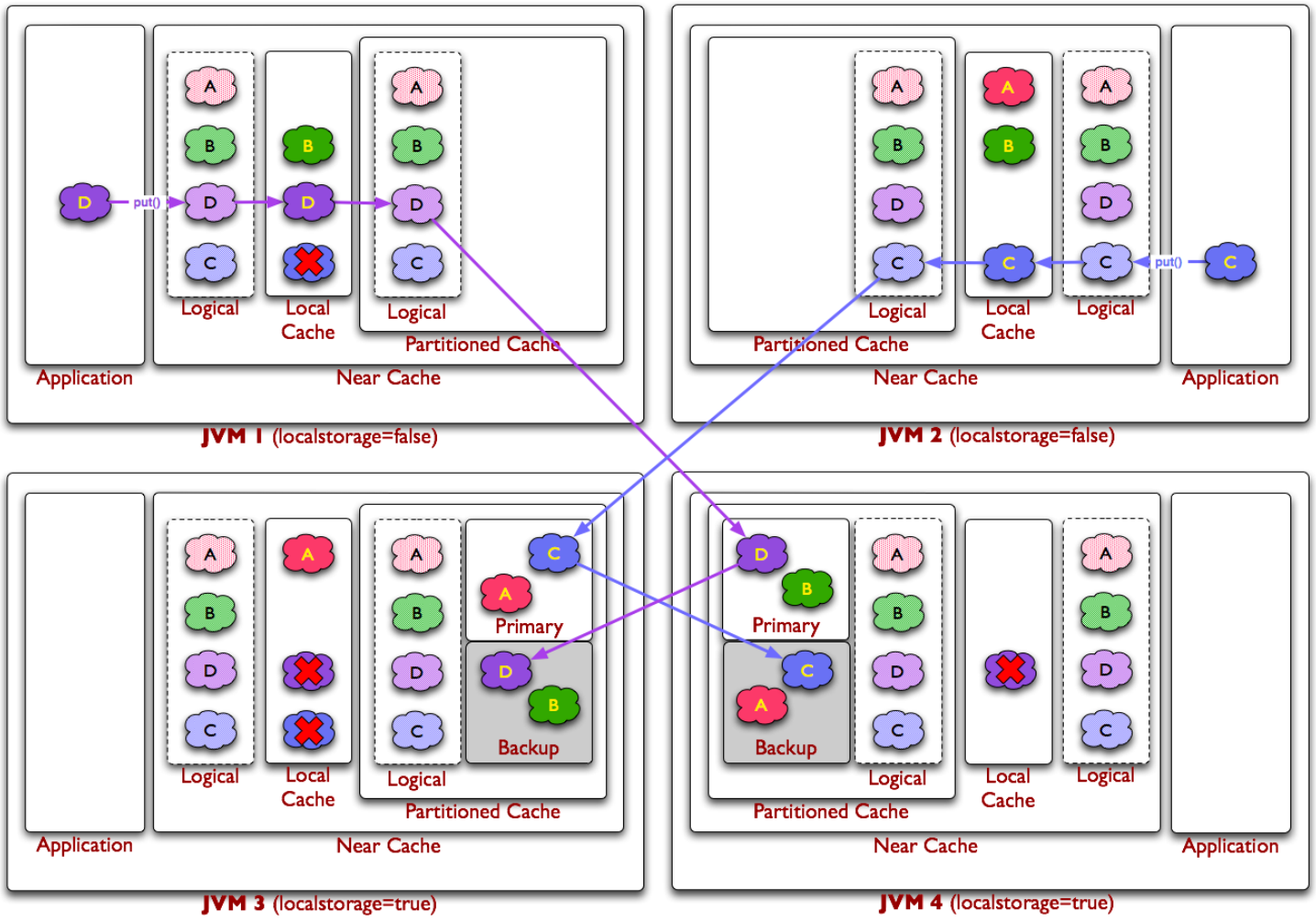


Write-Behind

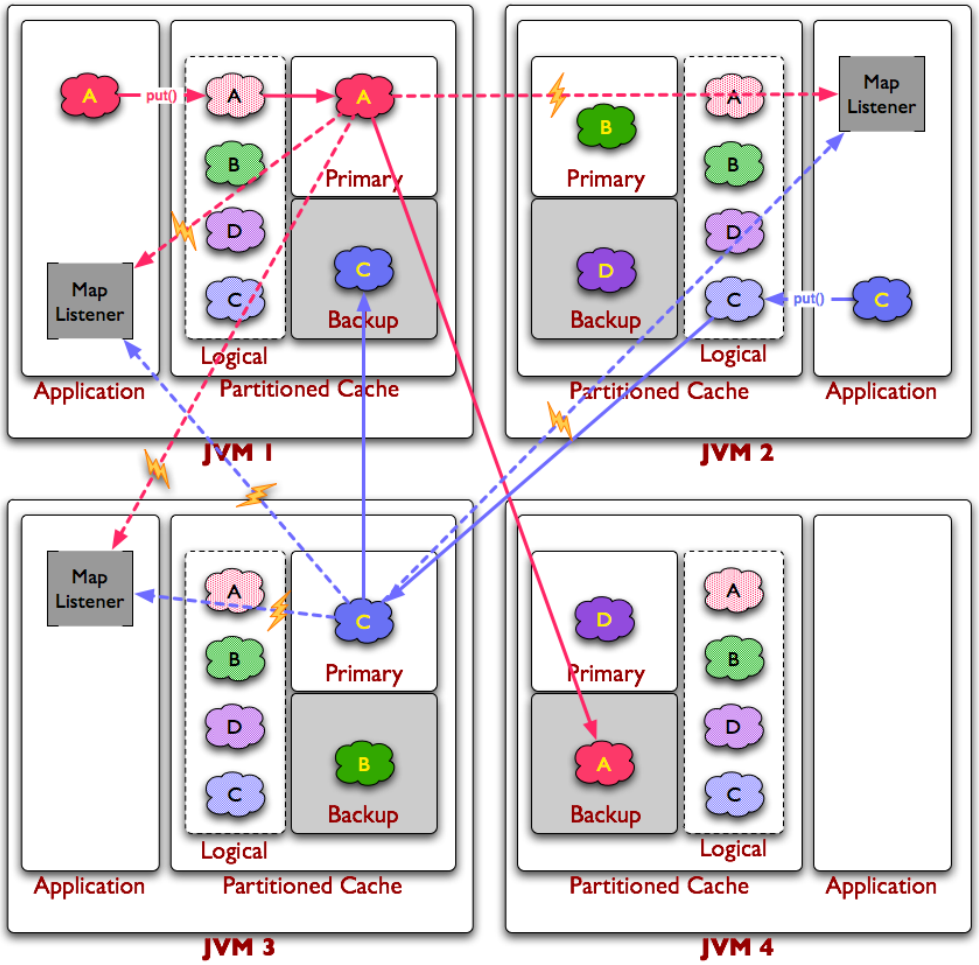
- Write-Behind accepts data modifications directly into the Data Grid
- The modifications are then asynchronously written back to the data source, optionally after a specified delay
- All write-behind data is synchronously and redundantly managed, making it resilient to server failure



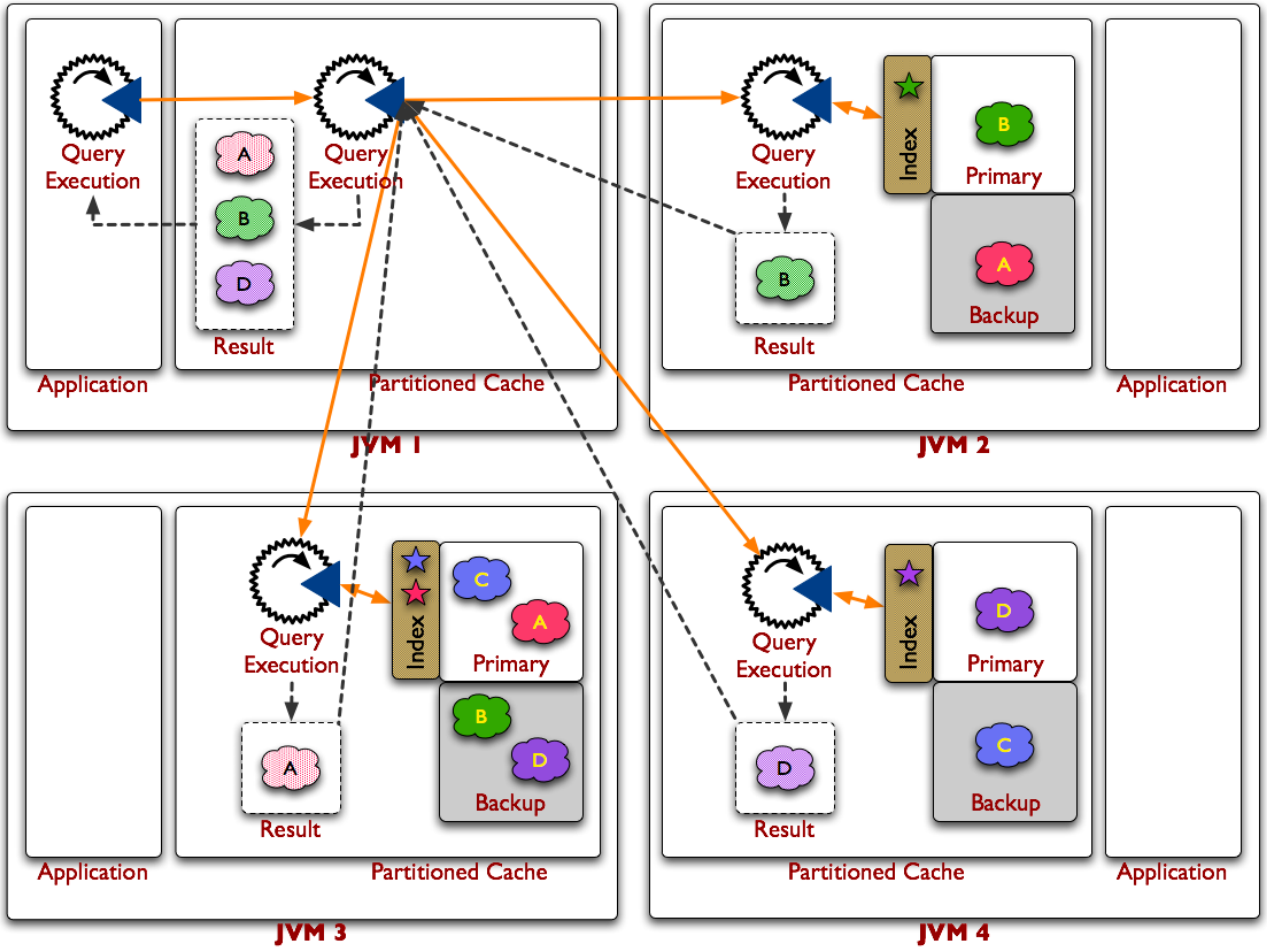
Topology Composition : Near Topology



Features : Observable Interface

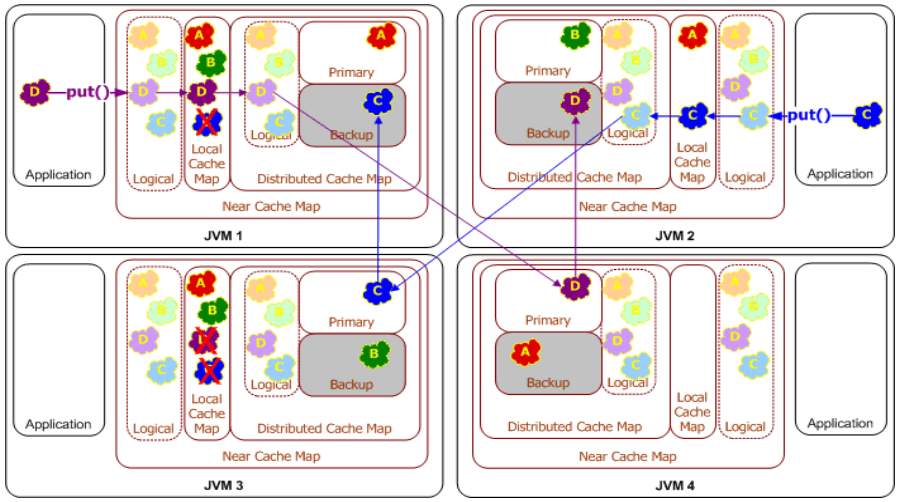


Features : QueryMap Interface



Concurrency

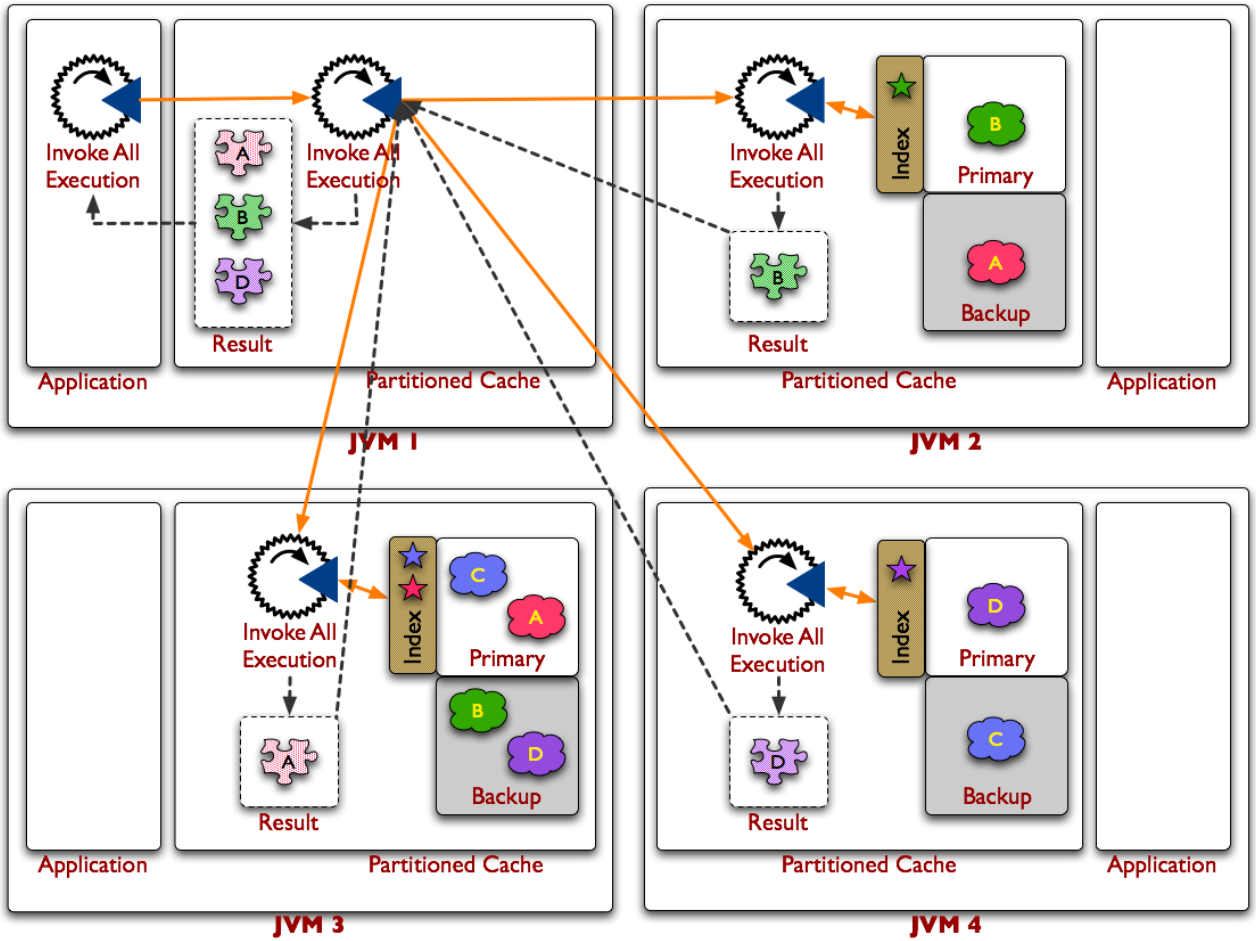
- Implicit: Queueing of operations
  - *Virtual* queue & thread per entry
- Explicit: Pessimistic locking
  - Grid-Wide Mutex
- Transactions: Unit of work management
  - Both optimistic and pessimistic transactions
  - Isolation levels from read-committed through serializable
  - Integrated with JTA



Features : InvocableMap Interface

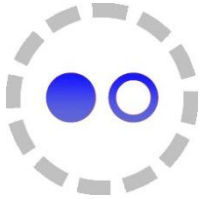
- Localized
- Parallel
- Once-And-Only-Once

- *"There are only two things you can distribute in a distributed system: State and Behavior. You either move the State to the Behavior or the Behavior to the State.."*



Data Grid Use Cases

General Use Cases for a Data Grid

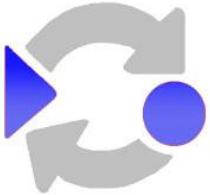


Caching

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

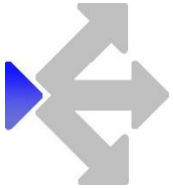
Analytics

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

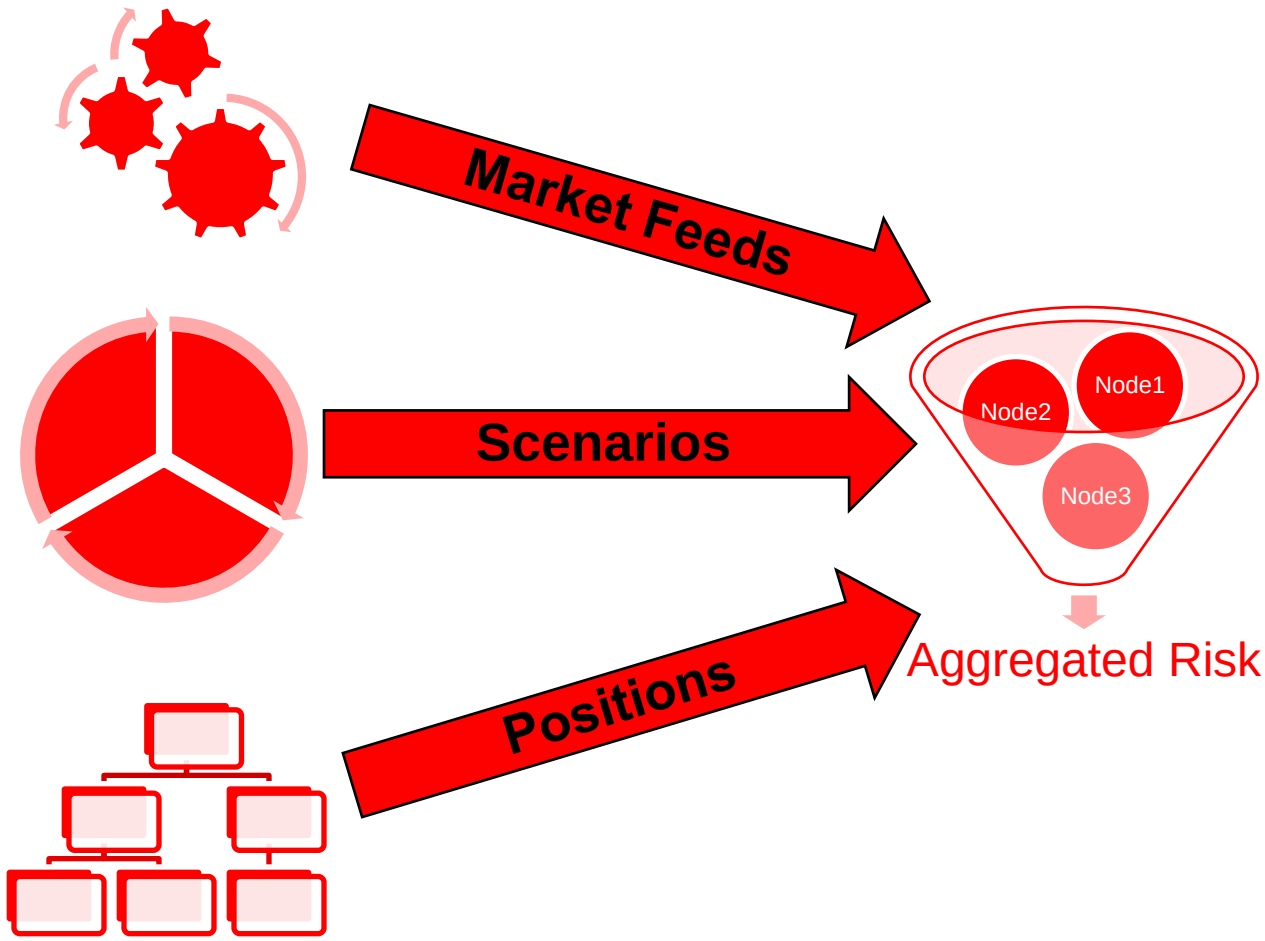


Events

Automated processing based on event

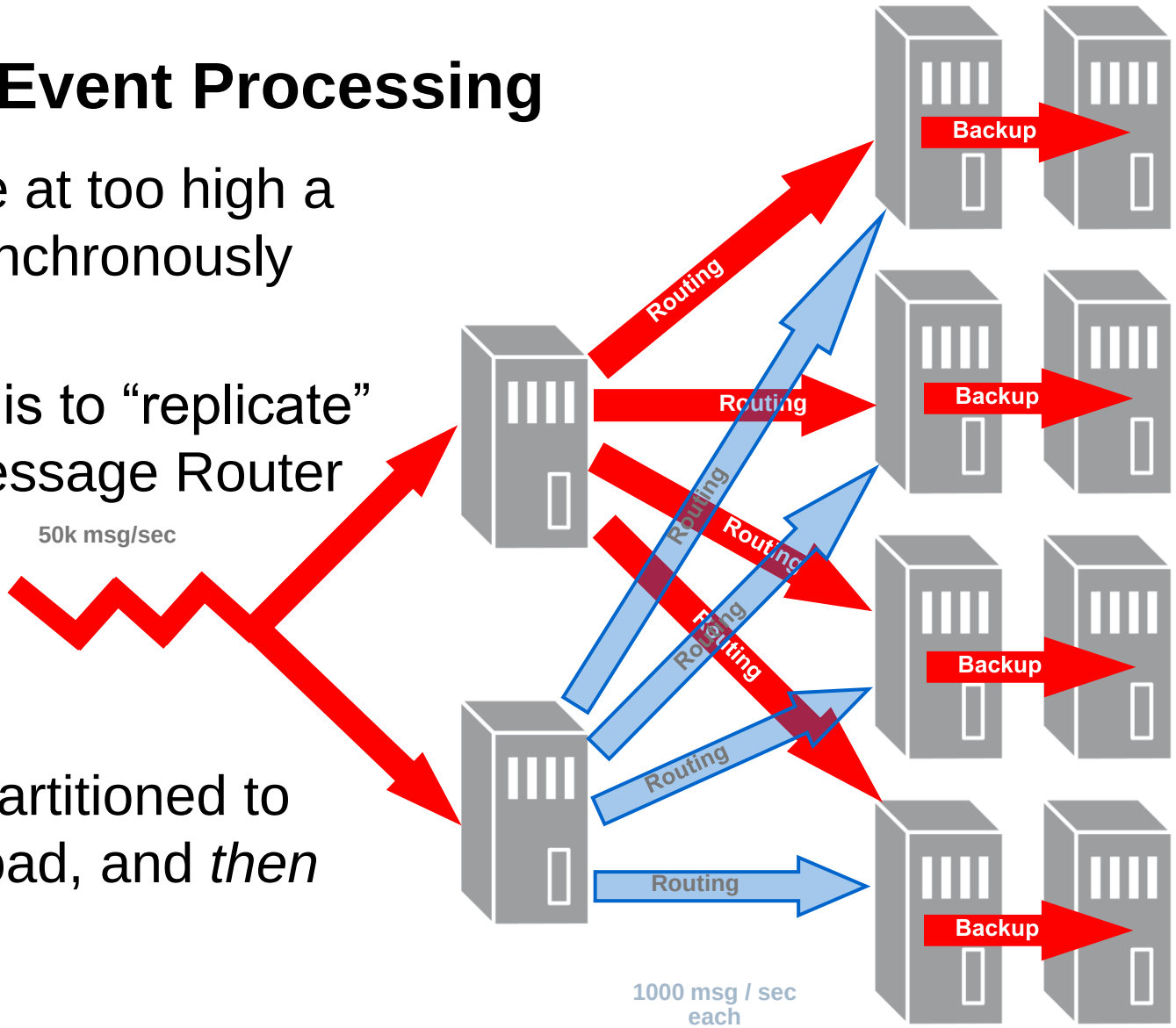
Financial Institution: Risk Analytics

- Large Data Sets
- Parallel Aggregation
- Capable of Absorbing Real Time Feeds
- Calc on Demand / Recalc on Changes
- 100% Data Locality for computations
- True Linear Scale
- Real World Results:
  - 50 Days -> 1 Hour



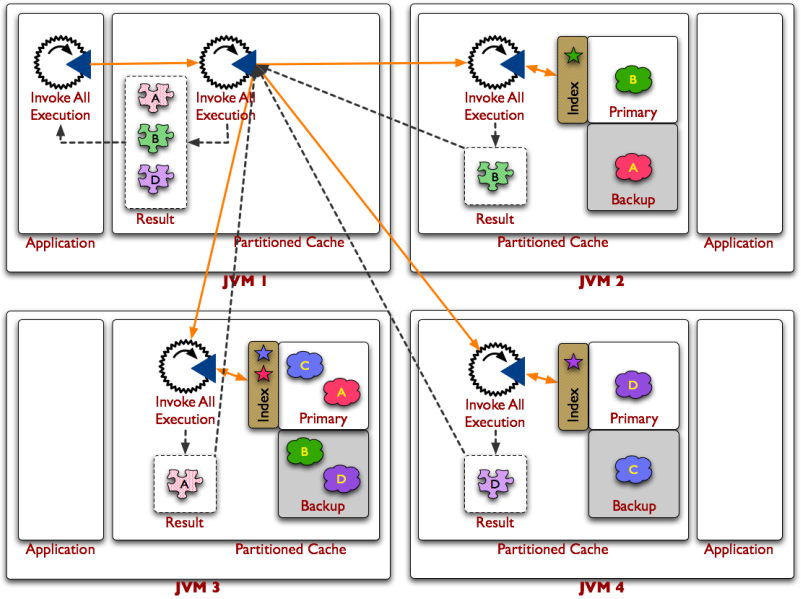
Example: Complex Event Processing

- Events arrive at too high a rate to be synchronously replicated
- The solution is to “replicate” the entire Message Router
- Events are partitioned to spread the load, and *then* replicated



“XTP”: What’s so extreme about it?

Grid-Based Transaction Processing



- There is a *Scalability Chasm*
  - Not an incremental solution
- Extreme Transaction Volumes
  - Sustained rates of over one million TPS on commodity blade servers
- Stock exchanges, utilities, banks, and the world’s busiest websites

Rethinking high scale architectures …

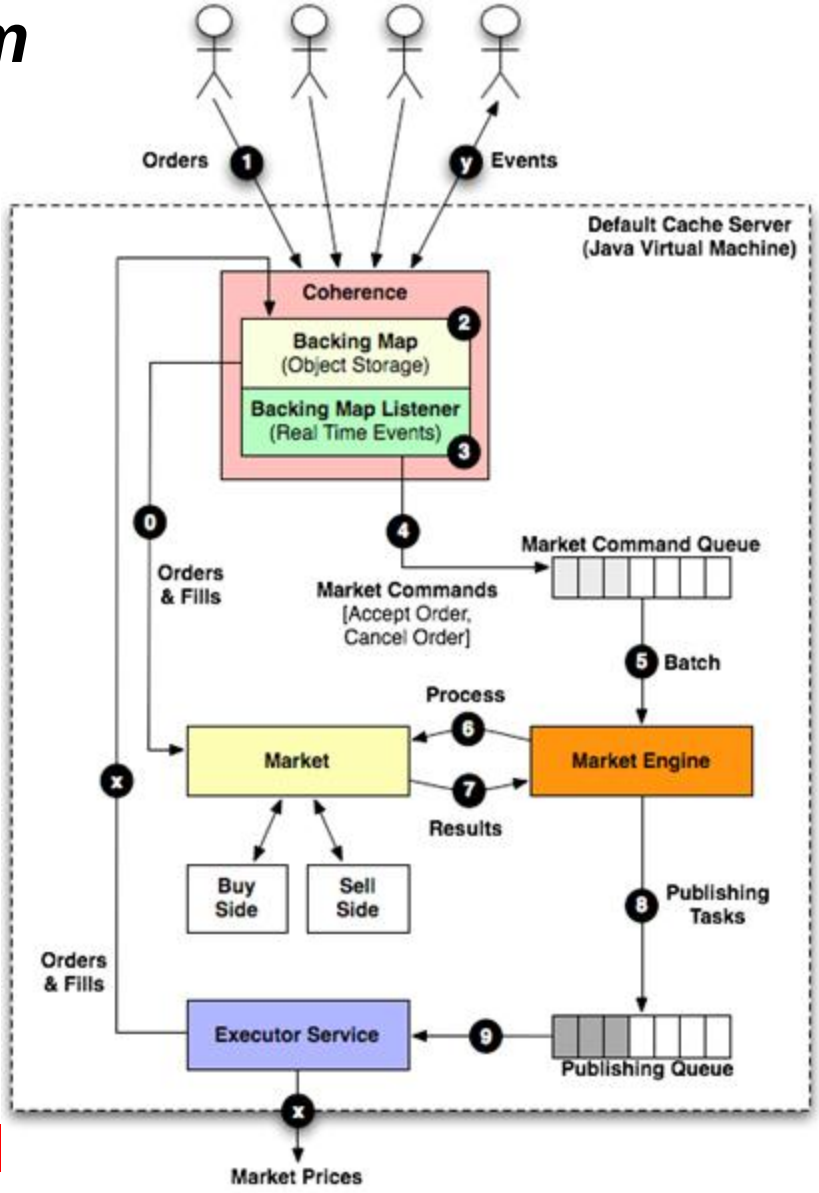
Data Grid: Building Blocks for XTP

- Partitionable Data Model
  - Key/Value Pairs
  - Reference by Identity
- Agents
  - Localized command pattern in a distributed environment
- Triggers and Events
  - One transaction's side-effects are another transaction's *raison d'etre*
- Persistence
  - Data are managed as Objects in the Data Grid
  - Persistence is usually relational

Example: Foreign Exchange

An Event-Driven FX System

- Architecture
  - Partitioned and Load-Balanced
  - System is a Finite State Machine
  - Data Affinity: Markets, Orders, Fills
  - Process Affinity: Markets & Matching Engines
  - Only Orders and Fills are replicated
  - Market state and Matching Engine is reconstructed on failover
- Results
  - 1000x improvement in throughput
  - 95% hardware cost savings by using a commodity hardware grid
  - 99%+ Orders processed in <3ms



Coherence

Tangosol Coherence

2000-2007

- Pioneer of Data Grid technology
  - First to provide Coherent Clustered Caching
  - First to provide Dynamic HA Partitioning
  - First to localize Data Processing
  - First to achieve Parallel Aggregation
- Consistent annual growth over 100%
- Innovation driven by a loyal, growing customer base

- *#1 in Data Grids*
- *#1 in Transaction Volumes*
- *#1 in Install Base*

Tangosol acquires Oracle

2007

- Kick in the Afterburner
  - Continuing Innovation: Increased investment in development
  - Growing Adoption: Leveraging Oracle's scale
  - Vibrant Ecosystem: Worldwide support, service, consulting, partners
- Increased annual growth
- Innovation continues to be driven by our customers

- *Like a kid in a toy-store …*
  - *Berkeley DB*
  - *TopLink Grid*

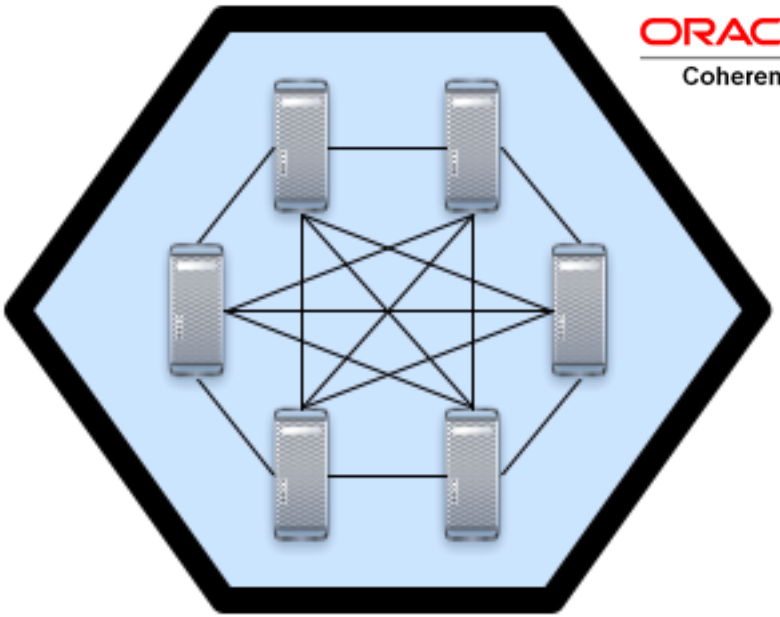
Orasol acquires BEA

2008

- Coming full circle: Immediate realized synergies
  - Real-Time Data Grid: jRockit RealTime and Coherence
  - WebLogic Suite and WebLogic Application Grid
  - WebLogic Liquid Operations Control
  - WebLogic Server and Portal Integration
  - CEP Engine Integration (OSGi)

- *Oracle now leads in every middleware category according to Gartner Magic Quadrants!*
- *Leading Developer Mindshare & Best of Breed Technology*

Oracle Coherence: A Unique Approach

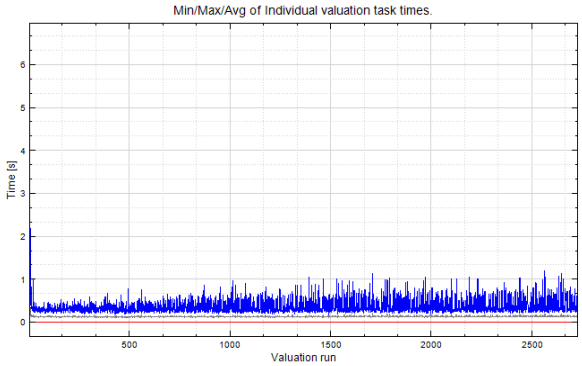


- Peer-to-Peer Clustering and Data Management Technology
- No Single Points of Failure
- No Single Points of Bottleneck
- No Masters / Slaves / Registries etc
- All members have responsibility to;
  - Manage Cluster Health & Data
  - Perform Processing and Queries
  - Work as a “team” in parallel
- Communication is point-to-point (not TCP/IP) and/or one-to-many
- Scale to limit of the back-plane
- Use with commodity infrastructure
- Linearly Scalable By Design

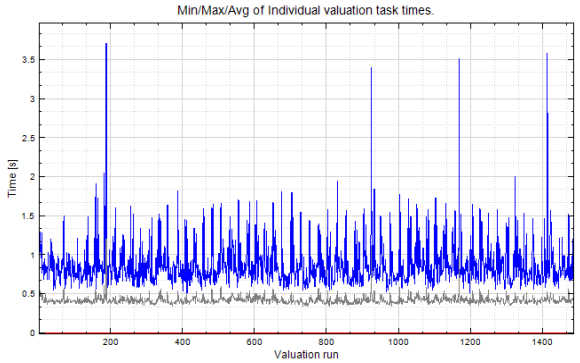
Oracle Coherence: Real World Results

- **From a presentation by Royal Bank of Scotland:**

  - Further competitive evaluation conducted that ratified Coherence as best of breed
    - Easiest to configure
    - Stable performance under load
    - Much harder to break



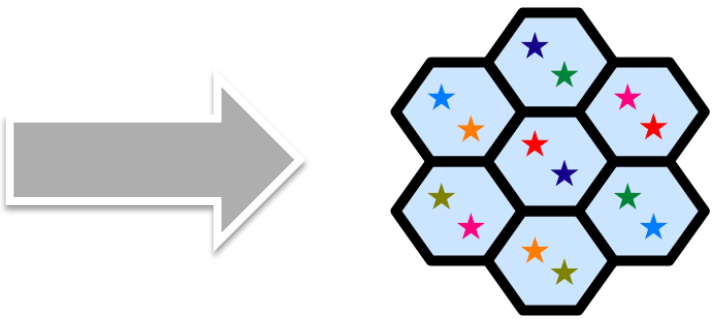
Coherence



Best of rest

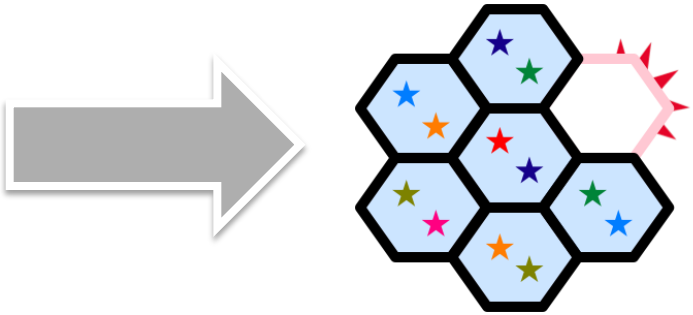
Oracle Coherence: A Unique Approach

- Data is automatically partitioned and load-balanced across the Server Cluster
- Data is synchronously replicated for continuous availability



- Servers monitor the health of each other
- When in doubt, servers work together to diagnose status

- Healthy servers assume responsibility for failed server (in parallel)
- Continuous Operation: No interruption to service or data loss due to a server failure



Oracle Coherence 3.4

Launched at Oracle OpenWorld

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

- POF: End-to-end support for Portable Object Format
- Pluggable Serializers: Control wire & memory formats
- Feature parity across Java, C# & .NET, C++ *(New!)*
- Triggers: Validate, modify and/or reject operations
- Event Transformers: Alter events at their origin
- Backup-Count-After-Write-Behind: Purges backup copies once data has been committed to a database
- JMX Reporter: Cluster-wide black-box recorder
- Java Platform Mbeans: Cluster-wide JVM statistics

Oracle TopLink Grid

Launched at Oracle OpenWorld

http://www.oracle.com/technology/products/ias/toplink/tl_grid.html

- JPA for Oracle Coherence
  - Oracle TopLink is the JPA Reference Implementation
- TopLink Grid stores all or part of an application's domain model in a Coherence Data Grid
- Features:
  - Simple configuration using annotations that align with standard JPA
  - Ability to choose which entities are stored in the grid versus those stored directly in the backing data store
  - Support for queries being executed against either the Grid or directly against the data store
  - Support for sequence generation and locking using either the Grid or the backing data store

Oracle Incubator

Launched 15 October 2008!

http://coherence.oracle.com/display/INCUBATOR/Home

- Command Pattern - This is a scale-out and Highly Available distributed implementation of the classic "Command Pattern".
- Store and Forward Messaging Pattern - This is a scale-out and Highly Available framework for distributed Store and Forward Messaging.
- Push Replication Pattern - This is a powerful framework for replicating the transactional data in real-time systems from one data center to another; it supports bi-directional WAN replication with application-customizable collision-reconciliation policies.

Q&A

- Learn more online:
  http://www.oracle.com/technology/products/coherence/