

ORACLE®



ORACLE®

Querying and Aggregating Data

Lesson 3

Objectives

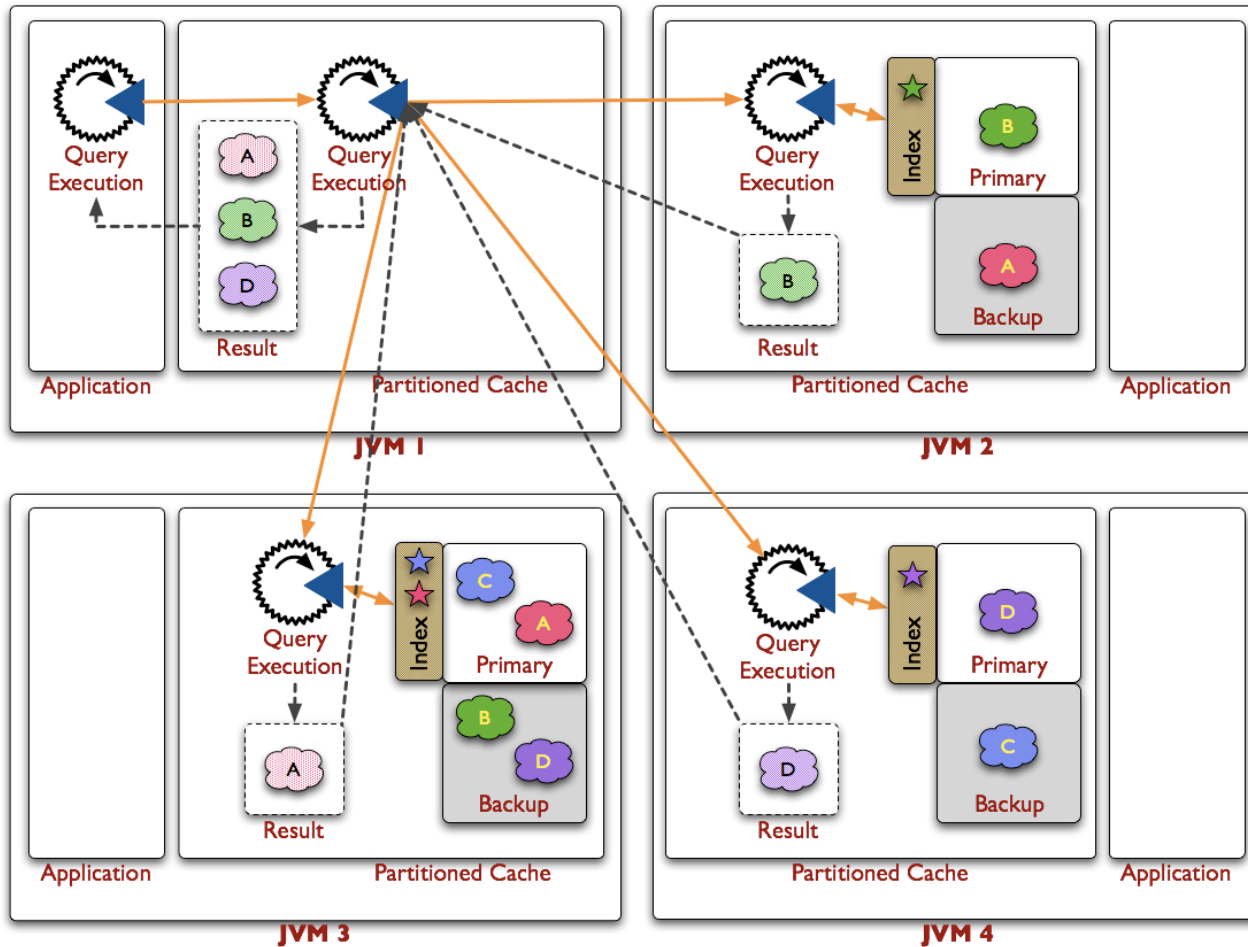
After completing this lesson, you should be able to:

- Describe the QueryMap interface
- Understand how to return a set of data given a query criteria
- Describe the various filters available
- Understand how to aggregate data in a cache using InvokableMap interface
- Understand how to apply an index to help aggregation performance

QueryMap Interface

- Use `com.tangosol.util.QueryMap` interface to search for Values or Keys
- Use Filters to restrict searching and thus results
- Filtering occurs at Cache Entry Owner
 - ie: In Partitioned Topology, Primary Partitions do the filtering
- Use QueryMap interface to define Indexes to allow for search optimization
- Create Continuous View of entries based on a Filter with real-time events dispatch
 - Perfect for client applications “watching” data

QueryMap Interface



QueryMap Interface - methods

- **Set entrySet(Filter filter)**
 - Return a set view of the entries that satisfy the criteria expressed by the filter.
- **Set entrySet(Filter filter, Comparator comparator)**
 - As above but iteration over the set will occur in ascending order according to the comparator.
- **Set keySet(Filter filter)**
 - Return a set view of the keys contained in this map for entries that satisfy the criteria expressed by the filter.
- **void addIndex(ValueExtractor extractor, boolean fOrdered, Comparator comparator)**
 - Add an index to a QueryMap.
- **void removeIndex(ValueExtractor extractor)**
 - Remove an index from this QueryMap.

QueryMap Interface - Examples

- A set containing all of the open trades

```
Set openTrades = trades.entrySet(new  
    EqualsFilter("isOpen", BOOLEAN.TRUE));
```

- A set containing people with a last name beginning with “Mac”

```
Set macPeople = people.entrySet( new  
    LikeFilter("getLastName", "Mac%"));
```

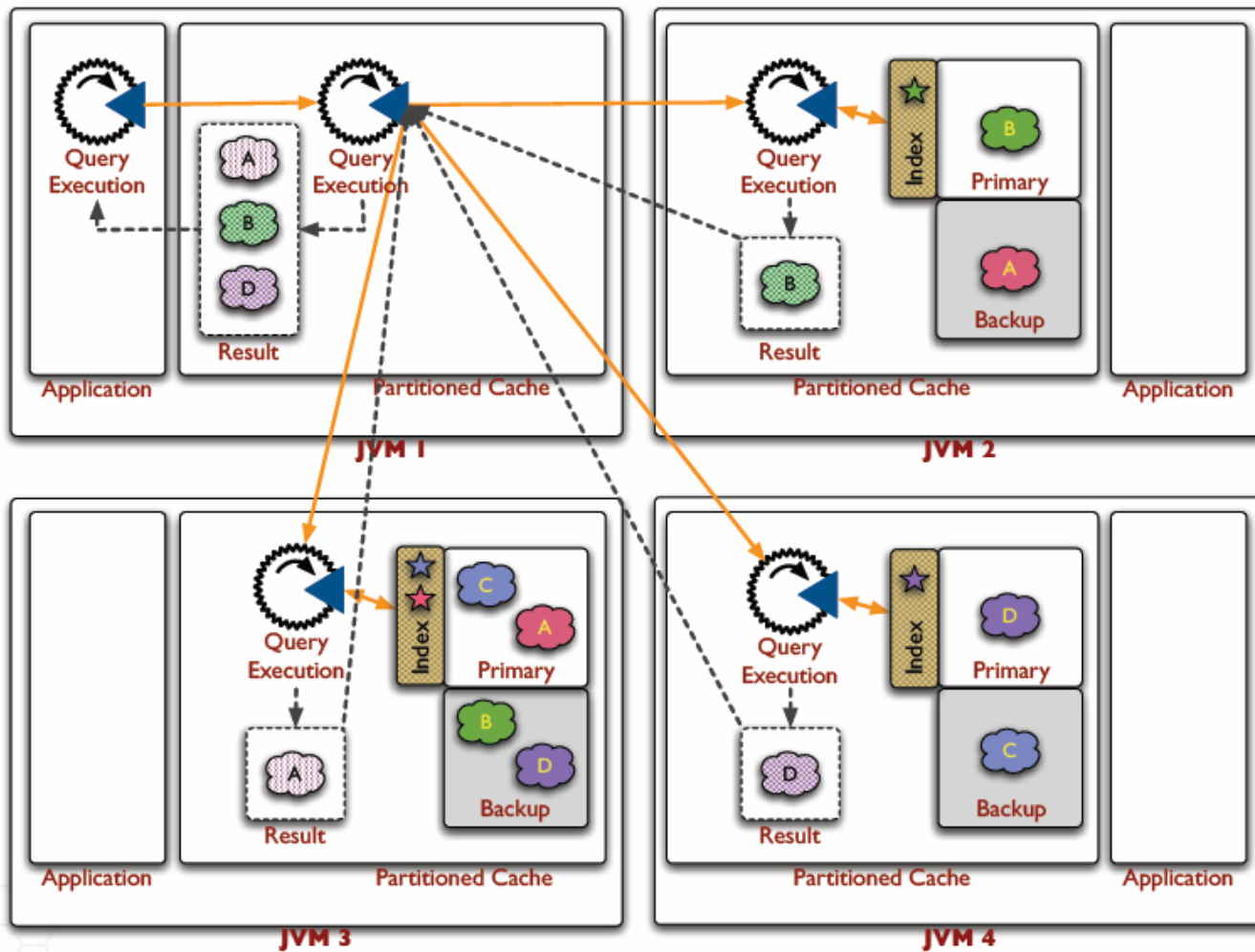
- A set of keys of people with a last name beginning with “Mac”
• or “Mc”

```
Set macPeopleKeys = people.keySet(  
    new OrFilter(  
        new LikeFilter("getLastName", "Mac%"),  
        new LikeFilter("getLastName", "Mc%")));
```

Indexes

- Each application using Coherence may suggest the same set of indexes when it starts
- There is no downside to an application blindly suggesting indexes regardless of whether another application has already suggested the same indexes
- Indexes are maintained by Cache Entry Owners
 - ie: For Partitioned Topology, the Primary Partitions maintain their own indexes

Indexes



Index Examples

- Suggest an index for trades based on their portfolio. Ensure the index is ordered, but use natural ordering (hence the null).

```
trades.addIndex(  
    new ReflectionExtractor("getPortfolio"),  
    true,  
    null);
```

- Suggest an index for trades based on their market. Don't use ordering.

```
trades.addIndex(  
    new ReflectionExtractor("getMarket"),  
    false,  
    null);
```

Features : InvocableMap Interface

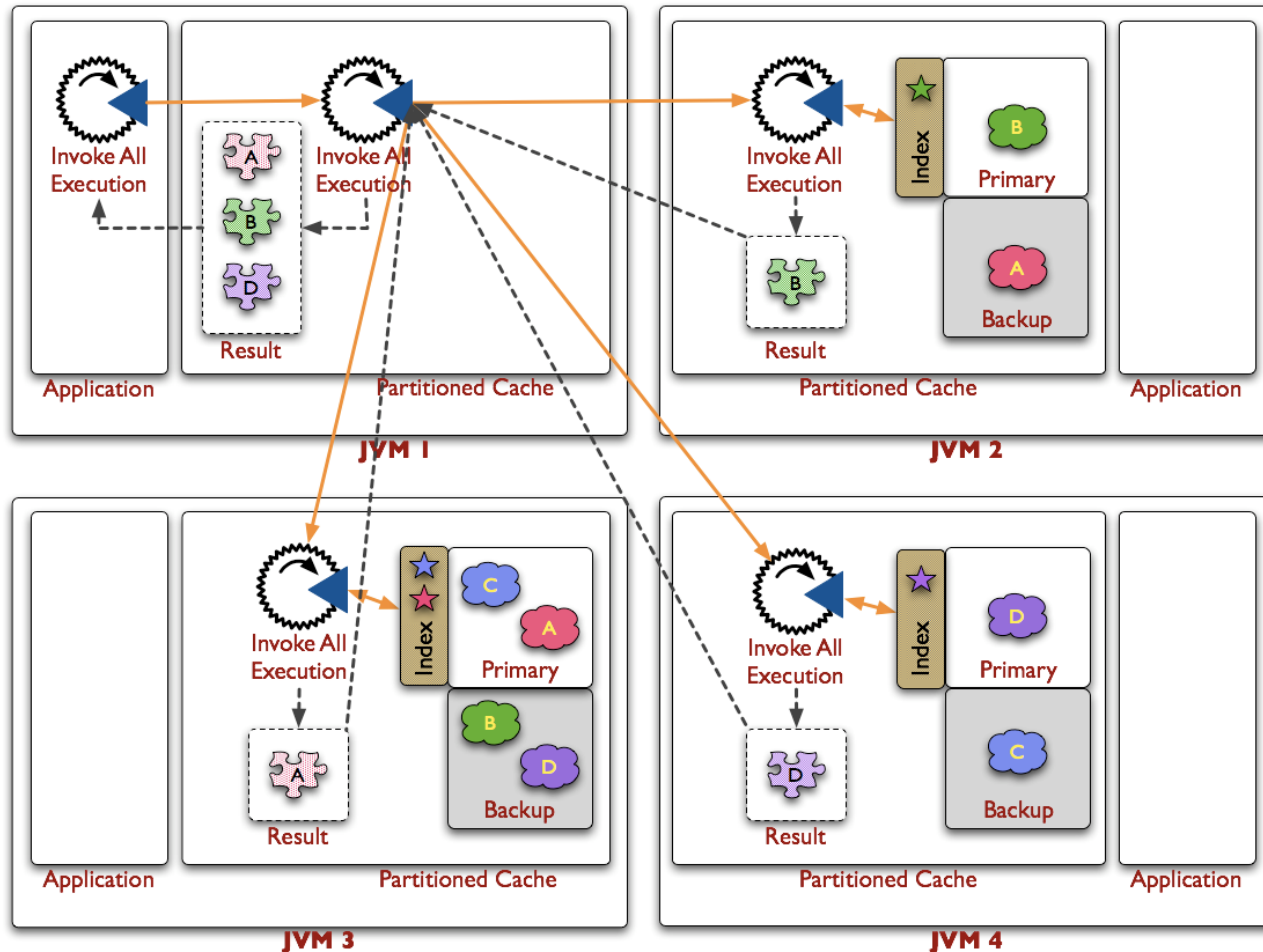
- Execute processors against an Entry, a Collection or a Filter
- Executions occur in parallel (aka: Grid-style)
- No “workers” to manage!
- Processors may return any value

```
trades.invoke(  
    new EqualsFilter("getSecurity", "ORCL"),  
    new StockSplit(2.0));
```

- Aggregate Entries based on a Filter

```
positions.aggregate(  
    new EqualsFilter("getSecurity", "ORCL"),  
    new SumFilter("amount"));
```

Features : InvocableMap Interface



Entry Aggregators

- `com.tangosol.util.InvocableMap.EntryAggregator` are agents that aggregate values from Entries
 - Sum, Average, Count, Max, Min, Distinct, GroupBy, Having...
- Equivalent to “agents” executing services in parallel on the data in the cluster
- Aggregation...
 - must not mutate Entries
 - is for data extraction and aggregation only!

Entry Aggregators

- **Object aggregate(Collection keys, InvocableMap.EntryAggregator aggregator)**
 - Perform an aggregating operation against the entries specified by the passed keys
- **Object aggregate(Filter filter, InvocableMap.EntryAggregator aggregator)**
 - Perform an aggregating operation against the set of entries that are selected by the given Filter

Examples

- The total value of the open orders

```
BigDecimal result = orders.aggregate(  
    new EqualsFilter("isOpen", Boolean.True),  
    new BigDecimalSum("getValue"));
```

- The categories of books on sale over \$25

```
Set categories = stock.aggregate(  
    new AndFilter(  
        new EqualsFilter("isOnSale", Boolean.True),  
        new GreaterThenFilter("getPrice", 25)),  
    new DistinctValue("getCategory"));
```

Summary

After completing this lesson, you should be able to:

- Describe the QueryMap interface
- Understand how to return a set of data given a query criteria
- Describe the various filters available
- Understand how to aggregate data in a cache using InvokableMap interface
- Understand how to apply an index to help aggregation performance

Lab 5

- Lab 5
 - Querying & Aggregating Data in a cache

ORACLE®