

ORACLE®



**ORACLE®**

**Coherence 3.7.1  
Query Explain Plan**

**Noah Arliss  
Development Manager (Sheriff) Oracle Coherence**



## Coherence Queries:

- Programmatic query mechanism
- Queries performed in parallel across the grid
- Standard indexes provided out-of-the-box and supports implementing your own custom indexes
- Cost-based analysis of Filter application
- Standard Filters provided out-of-the-box (e.g. OR, AND, ALL, EQUALS, etc.)

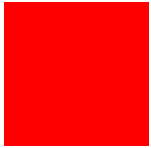


## Data Processing: Parallel Query

```
// get the "myTrades" cache
NamedCache cacheTrades =
    CacheFactory.getCache("myTrades");

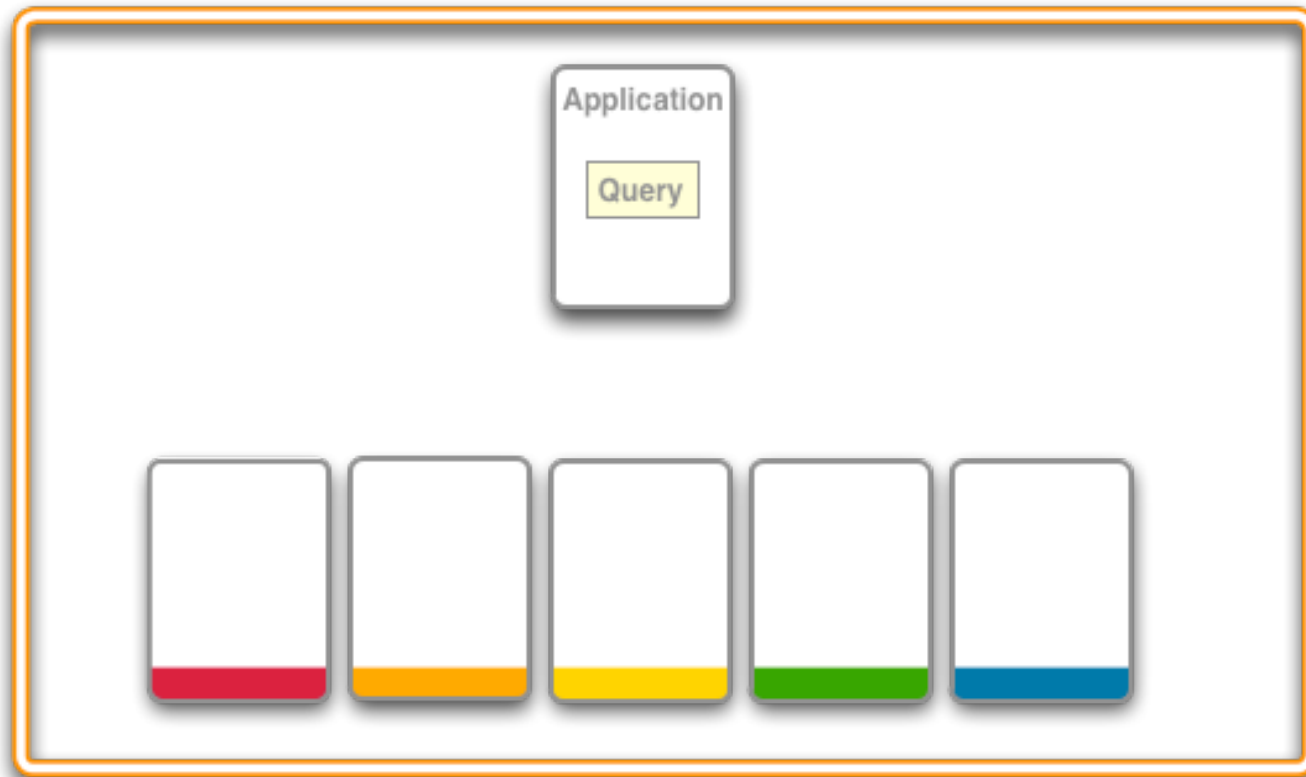
// create the "query"
Filter filter =
    new AndFilter(new EqualsFilter("getTrader",
        traderid),
        new EqualsFilter("getStatus", Status.OPEN));

// perform the parallel query
Set setOpenTrades = cacheTrades.entrySet(filter);
```



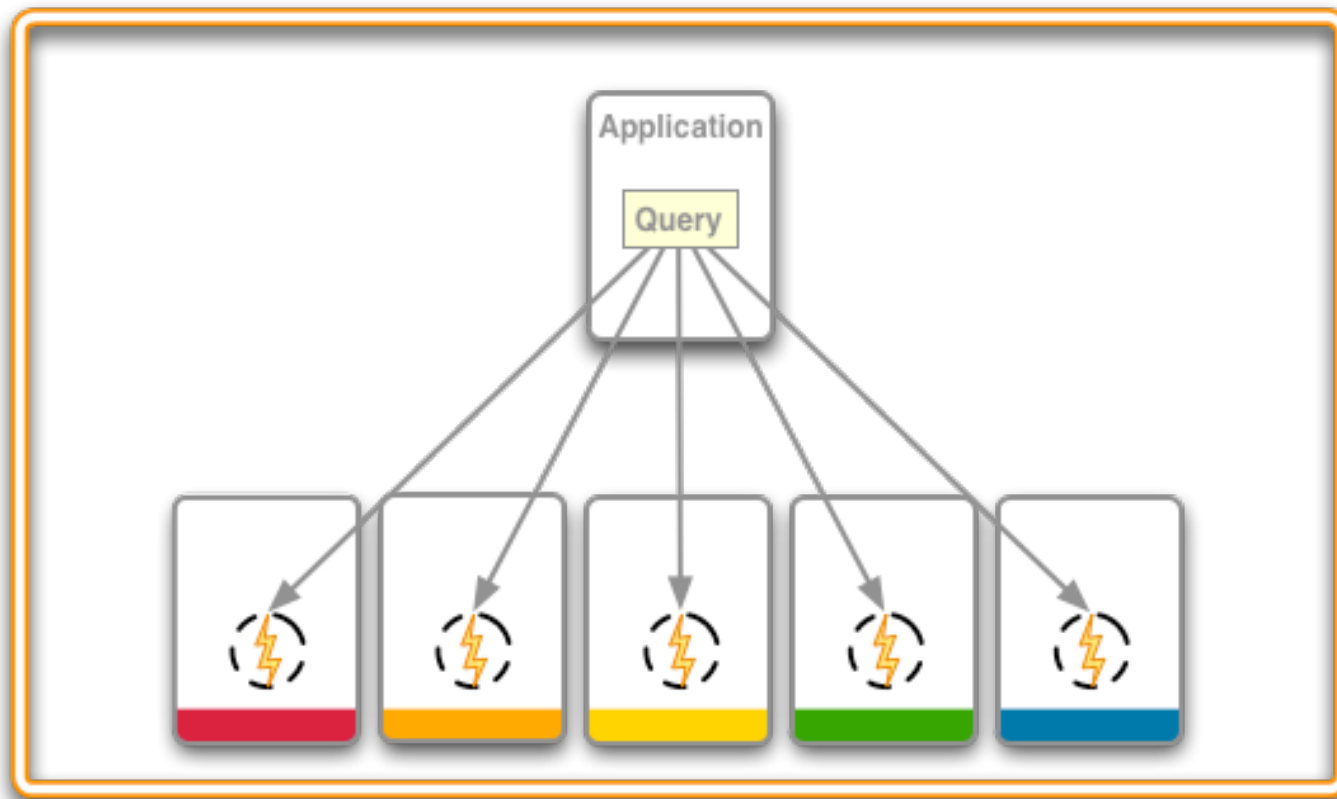
# Data Processing: Parallel Query

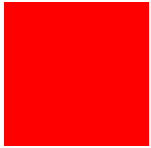
## Coherence Cluster



# Data Processing: Parallel Query

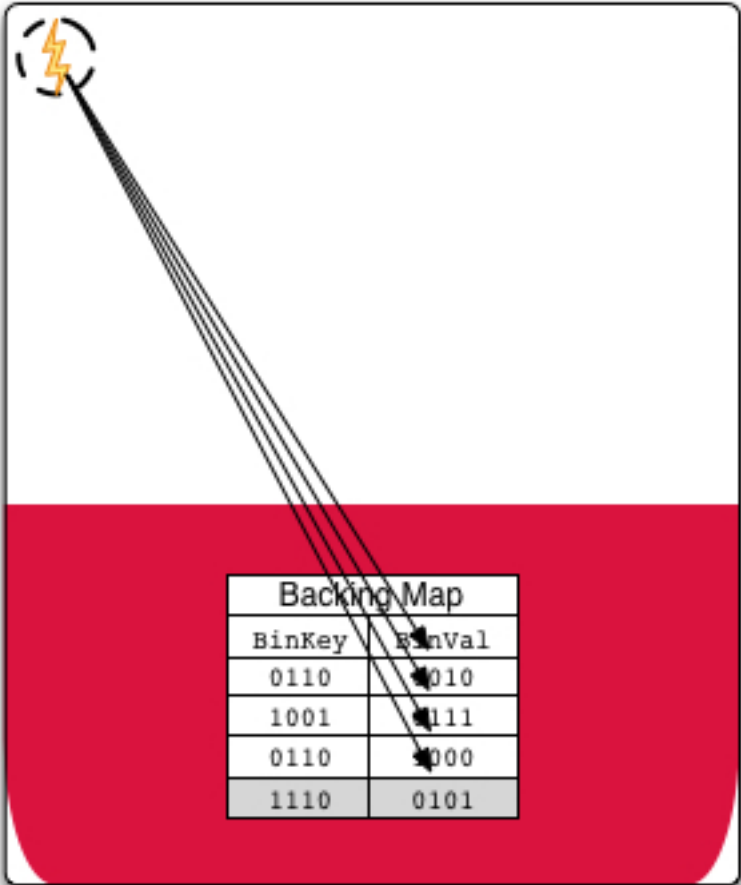
## Coherence Cluster



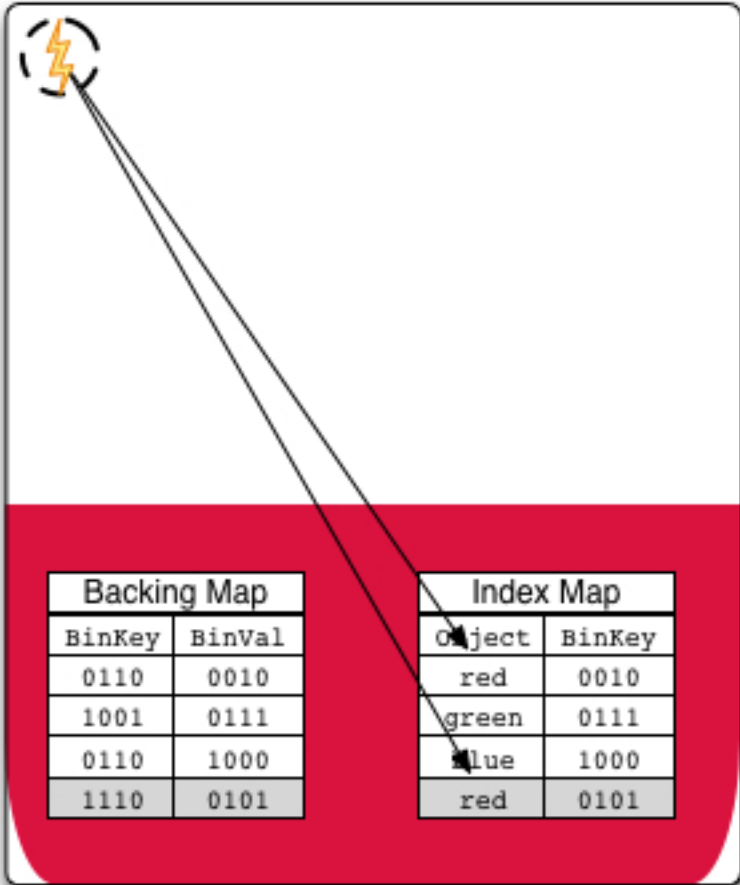


# Query in more detail

No Index

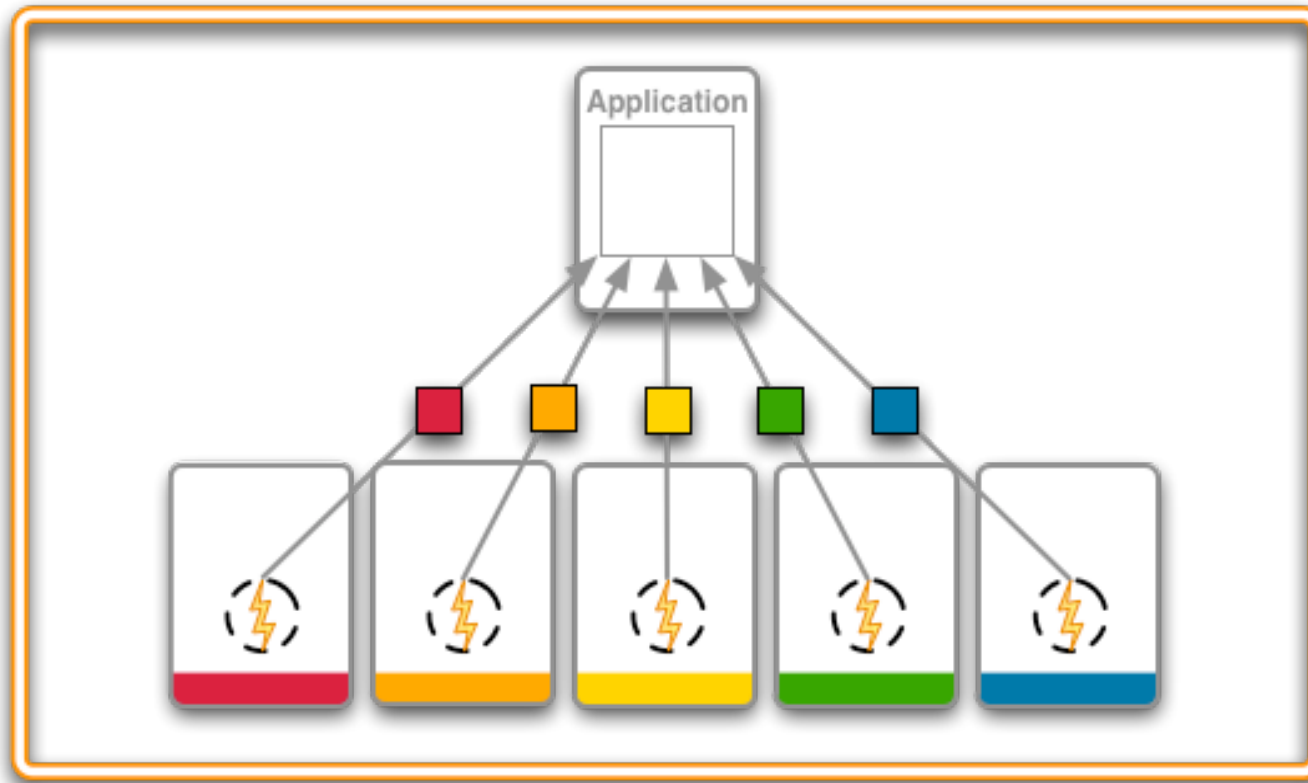


Index

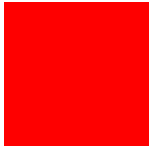


# Data Processing: Parallel Query

## Coherence Cluster

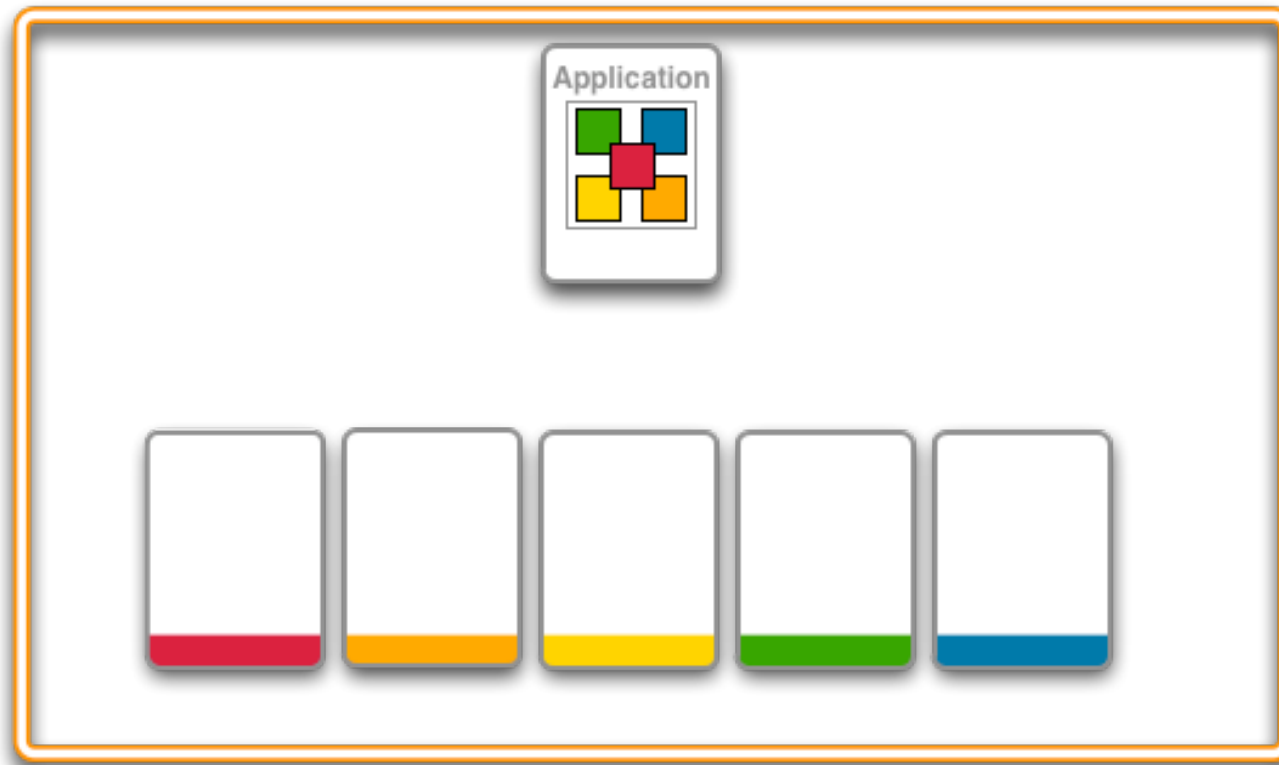






# Data Processing: Parallel Query

Coherence Cluster





## Optimizing Parallel Queries

- Indexing will greatly improve the performance of our queries
  - By reducing the number of entries to scan...
  - Because Indexed value objects are already de-serialized...
- How can I determine what to index



# The Query Explain Plan

- Evaluate a query's cost and effectiveness.
- Uses existing aggregation API.
- Produces a query record.
  - Contains information for each step (filter) that makes up the query.
- **EXPLAIN PLAN**
  - Estimated cost of evaluating a filter as part of a query operation.
- **TRACE**
  - Provides the actual cost of evaluating a filter as part of a query operation.
  - Actually runs the query in the cluster.



# How to Create a Query Record?

## QueryRecorder & QueryRecord

- **QueryRecorder** is a special `EntryAggregator`.
  - Signals Coherence to return a query record from the aggregation.
  - Aggregates the partial results into a query record.
  - Supports two types ... `EXPLAIN` and `TRACE`.
- **QueryRecord** carries information regarding the estimated or actual execution cost for a query operation.

```
// Get a query record for the given filter
```

```
QueryRecord record = (QueryRecord) cache.aggregate(filter,  
    new QueryRecorder(RecordType.EXPLAIN) );
```



# Query Explain Plan

Demo