

How to Run a Query in Distributed Mode – An Aspect

ORACLE

Resource Management in the Oracle Big Data Discovery Query Engine

ORACLE

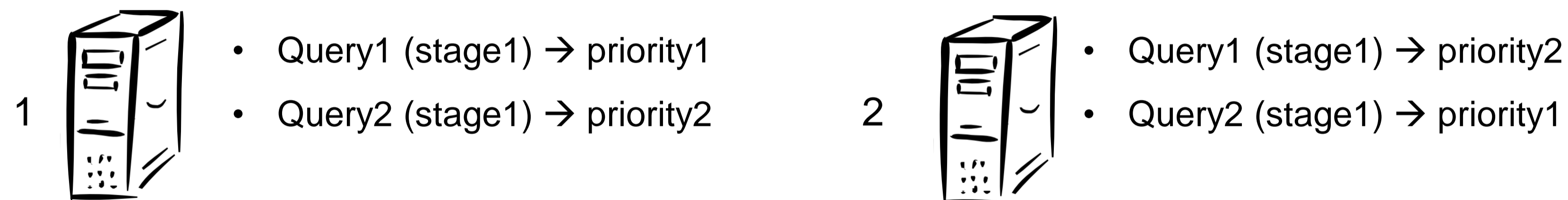
Ligia Nistor
ligia.nistor@oracle.com

Oracle Corporation, Cambridge, MA, USA

Executing a Query on Multiple Machines

- On **how many nodes** in the cluster will this query run?
- Will **all** the input data be available on all nodes or only **a partition** will be available on each node?
- Will the cluster have a **single master** or it will be a **peer to peer** system?
- How will we make sure that there is **no deadlock**?
- How will we make sure that **all the queries that start eventually finish**?

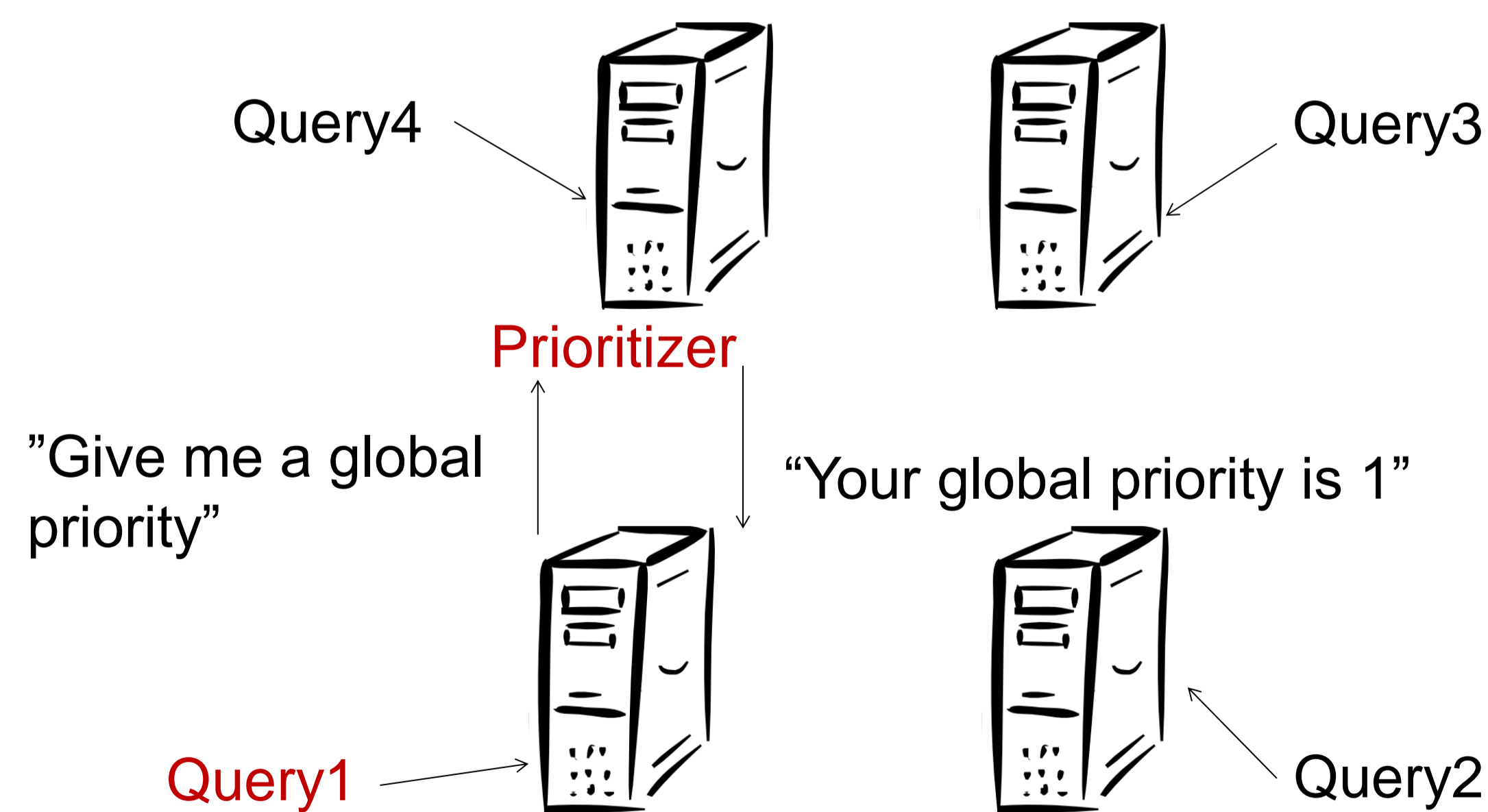
Deadlock Problem



- We need Query1(stage1) to finish on both machines to advance to Query1(stage2).
- It is a **deadlock** because Query1(stage1) is waiting for Query2(stage1) to finish and vice-versa. ✘

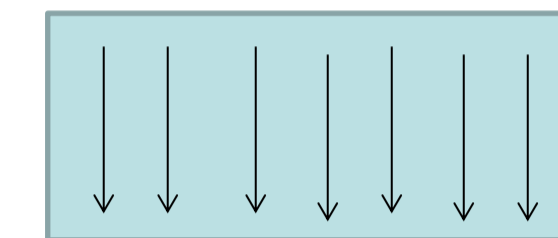
Solution

- A query has the **same priority** on all machines.



- All machines can be masters and receive queries.
- One machine is the **Prioritizer** that keeps track of priorities and delivers them to nodes.
- Planning of a query happens on a single node, evaluation is distributed on all nodes.

Motivating Scenario



- **32 threads** used for running queries, which are held in ExchangeScheduler
- **Policy:** the first query in the queue is given **16 threads**, the next query is given **8 threads**, the next one is given **4**, then **2** and finally **1**
- **In practice:** all queries do single threaded work at first → each query will get 1 thread
- **Problem:** 32 queries were run in the system at the same time, equal to the number of threads
- A query can be **anceled** in some cases when the same query **would have succeeded** if there were fewer queries being executed concurrently. ✘
- **Cancellation** of a query because of using too much memory can lead to the crash of the engine. ✘

Implementation

- Previous



- New Implementation



Admission policy: only admit a job if there are at least 1% available threads.

Results

- Ran a **test** having **more than 40000 concurrent queries**.
- While the total number of queries might go down slightly when using our proposed admission policy (or a similar one), there will be cases when queries **will not be canceled** and complete successfully → no crashes of the engine.

	32 core machine	40 core machine
No WLM	Test failed, query cancelled	42849
With WLM (1% policy)	15990	41313