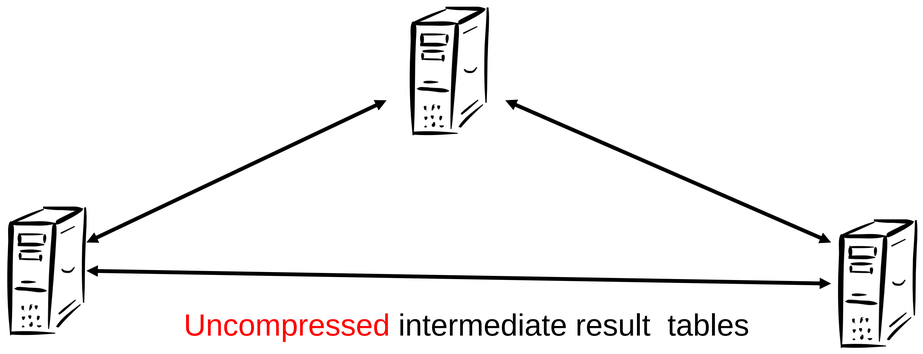
Using compression to reduce the query latency in a distributed system

Ligia Nistor
ligia.nistor@oracle.com
Oracle Corporation, Cambridge, MA, USA

Problem: query latency is too high

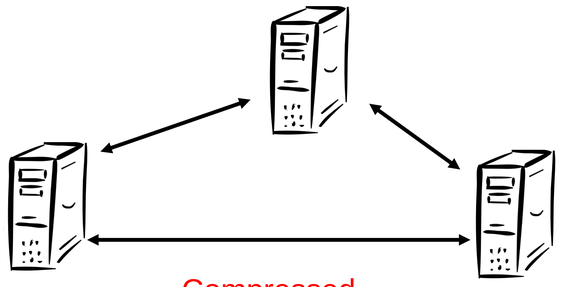
- The latency of running queries in distributed mode is too high.
- Sending uncompressed result tables across the network, in the cluster of nodes, influences the latency of a query run in distributed mode in our Big Data Discovery Query engine.
- The result tables containing partial results of evaluating a query written in EQL (Endeca Query Language) are sent between nodes at various times in the query execution, at stage boundaries.



Uncompressed intermediate result tables

Solution: send compressed result tables

- We reduce the size of result tables by using compression, thus we shorten the stage result wait times.



Compress tables here

Compressed intermediate result tables

Uncompress tables here

Implementation

- The compression algorithm that we use is LZ4. The two main functions are
  int LZ4_compress (const char* source, char* dest, int inputSize)
  int LZ4_uncompress (const char* source, char* dest, int outputSize).
- Each result table is compressed on the sending node and uncompressed on the receiving node.
- Ran the query below twice on a cluster of two nodes, on a dataset of 1 billion records.

```
RETURN result AS
SELECT MIN(passenger_count) as MIN FROM taxi GROUP
```

Results

	NO/WITH Compression	Evaluator	Code Generation Latency (ms)	Execution Latency (ms)	Execution CPU (ms)	Peak Memory Usage (MB)
First run	NO	0	58.7	12536.1	100712.9	12354.5
	WITH	0	55.2	8666.8	98535.7	12354.5
	NO	1	59.2	12537.3	97810.2	6180.9
	WITH	1	61.1	8667.5	96957.2	6181.9
Flushed cache	NO	0	56.3	12876.4	107207.3	12354.5
	WITH	0	61.9	7496.1	90132.5	12354.5
Second run	NO	1	56.6	12877.1	109234.7	6181.4
	WITH	1	58.2	7497.7	95545.1	6183.4