The Power of Choice in Data-Aware Cluster Scheduling

Shivaram Venkataraman\textsuperscript{1}, Aurojit Panda\textsuperscript{1}
Ganesh Ananthanarayanan\textsuperscript{2}, Michael Franklin\textsuperscript{1}, Ion Stoica\textsuperscript{1}

\textsuperscript{1}UC Berkeley, \textsuperscript{2}Microsoft Research
Data grows faster than Moore’s Law

[Kathy Yelick LBNL, VLDBJ 2012, Dhruba Borthakur]
Trends: Big Data

Facebook Hive cluster
Last 4 years: data growth 2500x queries/day 60x

Microsoft Scope Cluster
“The number of daily jobs has doubled every six months for the past two years.”

[Kathy Yelick LBNL, VLDBJ 2012, Dhruba Borthakur]
Trends: Low Latency

- 2004: MapReduce batch job
- 2009: Hive
- 2010: Dremel
- 2012: In-memory Spark

Latencies:
- 10 min
- 1 min
- 10s
- 2s
Big Data or Low Latency?

SQL Query: 2.5 TB on 100 machines

> 15 minutes  1 - 5 Minutes  < 10s
Sampling
Applications

Approximate Query Processing
  blinkdb, presto, minitable

Machine learning algorithms
  stochastic gradient, coordinate descent
Choices

N

Any K

N

Any K
Sampling $\rightarrow$ Smaller Inputs + Choice
Example

\[ N = 4 \]

\[ K = 2 \]
Available (N) = 2

Required (K) = 2

Available Data

Running

Unavailable Data

Busy
Available (N) = 4

Available Data

Required (K) = 2

Running

Busy

Choice-Aware
Choice-Aware

Available (N) = 4
Launched (M) = 3
Required (K) = 2

1
2
3
4

Available Data
Running
Busy

Rack

Time
KMN Scheduler

- How much can KMN improve locality
- Propagate benefits across stages
- Handling stragglers
Job → DAG

KMN Scheduler

One-to-One

Many-to-One
One-to-One Stages

Locality

Disk ~ 100MB/s
Network ~ 10 Gbps (~1GB/s)
Memory ~ 50GB/s
KMN Locality

\( \binom{N}{K} \) Choices

Any K

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
2 & 3 & 4 & 1 \\
3 & 4 & 1 & 2 \\
4 & 1 & 2 & 3 \\
\end{array}
\]
Locality, $K=100$

K – Number of blocks chosen
N – Number of blocks available

KMN significantly improves locality
Many-to-One Stages
Many-to-One Stage

15 transfers
Many-To-One Transfers

![Diagram showing many-to-one transfers]

- Core
  - M1
    - R2
  - M1
    - R3
  - M2
    - R2
  - M3
    - R3
  - M3
    - R3

M1, M2, M3, M4, M5

R1, R2, R3
Bottleneck Link

Link with Max. transfers

Cross Rack Data Skew

Maximum transfers

Minimum transfers

\[ \frac{6}{2} = 3 \]
Facebook Trace

Cross Rack Data Skew

CDF

Cross Rack Data Skew

Maximum transfers

Minimum transfers

<50 tasks
50-150 tasks
>150 tasks
Power of Choice

Load balancing: balls and bins

Insight: Run extra tasks ($M > K$)

Cross Rack Data Skew = 3
Power of Choice

Technique:
Spread out choice of K tasks to reduce skew

M = 7, K = 5
Cross Rack Data Skew = 2
Handling Stragglers

Rack

Stragglers vs. Cross-Rack Data Skew

Time

M_1
M_2
M_3
M_4
M_5
M_6
M_7
Using KMN

// Create Spark RDD
file = sc.textFile("tpc-h.data")

// Select a 10% sample using KMN
sample = file.blockSample(0.1)

// RDD operations
sample.map { li =>
    (li.linestatus, li.quantity)
}.collect()
Also in the paper

User-defined sampling functions

Placing reduce tasks

Killing extra tasks
Evaluation

Facebook traces replay
Long DAGs (Stochastic Gradient Descent)
SQL queries from Conviva
Reducer placement
Varying Utilization

Baseline: Use a pre-selected random sample
Setup: 100 m2.4xlarge EC2 machines, 60GB RAM/mc
Facebook Overall

Job Completion Time (s) vs. Job Size

- Baseline
- KMN-M/K=1.05

- >100
- 11-100
- 0-10
How many extra tasks?

Cross-Rack Skew

M/K=1.0
M/K=1.1
M/K=2.0

50 - 150 tasks

> 150 tasks
KMN: How many stages?

Gradient

Aggregate 1

Aggregate 2

Aggregate 3

Stochastic Gradient Descent
KMN: How many stages?

<table>
<thead>
<tr>
<th>KMN Stages</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient</td>
<td>15.27</td>
</tr>
</tbody>
</table>

Diagram:
- Gradient
- Aggregate1
- Aggregate2
- Aggregate3

Table:
- Gradient: 15.27 seconds
KMN: How many stages?

<table>
<thead>
<tr>
<th>KMN Stages</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient</td>
<td>15.27</td>
</tr>
<tr>
<td>Gradient + Agg1</td>
<td>12.72</td>
</tr>
</tbody>
</table>
KMN: How many stages?

<table>
<thead>
<tr>
<th>KMN Stages</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient</td>
<td>15.27</td>
</tr>
<tr>
<td>Gradient + Agg1</td>
<td>12.72</td>
</tr>
<tr>
<td>Gradient + Agg2</td>
<td>11.79</td>
</tr>
</tbody>
</table>
KMN: How many stages?

<table>
<thead>
<tr>
<th>KMN Stages</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient</td>
<td>15.27</td>
</tr>
<tr>
<td>Gradient + Agg1</td>
<td>12.72</td>
</tr>
<tr>
<td>Gradient + Agg2</td>
<td>11.79</td>
</tr>
<tr>
<td>Gradient + Agg3</td>
<td>12.09</td>
</tr>
</tbody>
</table>
Related Work

Power of Choice
  Power-of-Two choices [TPDS’01]
  Sparrow [SOSP’13]

Improving Cluster Scheduling
  Quincy [SOSP’09]
  alsched [SOCC’12]
  Dolly [NSDI’13]
KMN Scheduler

Emerging applications: ML algorithms, AQP
Improves locality, Balances network transfers