Heterogeneity-Aware Resource Allocation and Scheduling in the Cloud

Gunho Lee (UC Berkeley)
Byung-Gon Chun (Yahoo! Research)
Randy H. Katz (UC Berkeley)
We have resources and jobs

Resource

Job/Task
Then schedule jobs/tasks on them
Goal 1. Minimize the cluster size while providing good performance
Goal 2. Provide each job with “fair share” of resources
Heterogeneity makes the problem more complex
Our Approach

• Consider *Job Affinity* to match more suitable resources to jobs

• Redefine a share metric to provide fairness

• Allocation
  – Core Nodes + Accelerator Nodes

• Scheduling
  – Progress Share
Fair Share Metric

• The scheduler try to equalize “share” of all jobs
  – SlotShare: Number of slots owned
    • Does not work well in heterogeneous environments
  – ProgressShare: Progress being made with owned slots / all slots
    • Contribution of a slot to a job’s progress rate
Progress Share

Progress without sharing (1 job)
Progress Share

Just good progress with sharing (2 jobs)

Progress without sharing (1 job)
Progress Share

- Progress without sharing (1 job)
- Just good progress with sharing (2 jobs)
- (Even better)
- (Under-served)
Progress Share of Job A =
Ratio of progress slope \( \frac{b}{a} \)
Homogeneous case

Progress

0

1

Time

Job A

Job B

Slot

Share

0

1

0

1

Progress

Share
Heterogeneous case

Job A runs faster on gray slots

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
A & A \\
\hline
A & A \\
\hline
A & A \\
\hline
A & A \\
\hline
A & A \\
\hline
A & A \\
\hline
A & A \\
\hline
A & A \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
B & B & B & B & B & B \\
\hline
B & B & B & B & B & B \\
\hline
B & B & B & B & B & B \\
\hline
B & B & B & B & B & B \\
\hline
B & B & B & B & B & B \\
\hline
B & B & B & B & B & B \\
\hline
B & B & B & B & B & B \\
\hline
B & B & B & B & B & B \\
\hline
\end{tabular}
\end{table}

\begin{figure}
\centering
\begin{tikzpicture}
\begin{axis}[
    xlabel=Time,
    ylabel=Progress,
    axis lines=left,
]
\addplot[blue,mark=*,mark options=solid] coordinates {
    (0,0)
    (1,1)
};
\node at (axis cs:0.5,0.5) {Job A};
\end{axis}
\end{tikzpicture}
\end{figure}

\begin{figure}
\centering
\begin{tikzpicture}
\begin{axis}[
    xlabel=Time,
    ylabel=Progress,
    axis lines=left,
]
\addplot[red,mark=*,mark options=solid] coordinates {
    (0,0)
    (1,1)
};
\node at (axis cs:0.5,0.5) {Job B};
\end{axis}
\end{tikzpicture}
\end{figure}
Heterogeneous case 1

Using SlotShare

Progress

Slot Share

Time

Progress Share

Time

Job A

Job B
Heterogeneous case 1

Using SlotShare

<table>
<thead>
<tr>
<th>B</th>
<th>B</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

![Graph showing progress and slot share over time for jobs A and B.]

Job A

Progress Share

Job B

Progress Share
**Heterogeneous case 1**

Using SlotShare

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>B</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>B</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>B</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

Job A is making less progress, with the same number of slots.

**Progress vs Time**

- **Job A**
- **Job B**
### Heterogeneous case 2

#### Using ProgressShare

<table>
<thead>
<tr>
<th>Time</th>
<th>Slot Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The diagram illustrates the progress and share of two jobs, Job A and Job B, over time. The progress and share are represented on the vertical and horizontal axes, respectively. The shaded area indicates a specific state or condition.
Heterogeneous case 2

Using ProgressShare

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>

![Graph showing progress and time for Jobs A and B]
Heterogeneous case 2

Using ProgressShare

Both jobs making progress $\geq 0.5$
Performance Gain of Using Progress Share

![Bar chart showing competition time in minutes for Slot Share and Progress Share with 'Crypt' and 'Sort' categories.](chart.png)
Summary

• Heterogeneity should be taken account at both levels of two-level scheduling
  – Resource Allocation and Job Scheduling

• Need to redefine “share” to provide performance and fairness simultaneously in heterogeneous environments
  – Propose “progress share”

• Future Work
  – Combine with sub-linear performance model
  – Consider inference of co-located jobs