AlloX: Compute Allocation in Hybrid Clusters

Tan N. Le    Xiao Sun    Mosharaf Chowdhury    Zhenhua Liu

tnle@cs.stonybrook.edu
Resource Allocation in Clusters

Performance

Fairness

Utilization

99 %
Resource Allocation Design Space

Single Resource

Time

Space

Multiple Resources

Traditional

CPU Sharing

Memory Sharing

DRF (nsdi’11),
Carbyne (osdi’16)
HUG (nsdi’16)

Interchangeable

Deadlines

Performance & Fairness

TetriSched
(eurosys’16)

AlloX
Interchangeability in Resources

Same applications run on different resource types

- Tensorflow, Caffe, Pytorch, Matlab
  - CPU
  - GPU

- CNNLab, PaddlePaddle
  - FPGA
  - GPU

- Tensorflow
  - CPU
  - GPU
  - TPU

Modern Frameworks support Interchangeability

https://github.com/PaddlePaddle/Paddle
https://github.com/cnnlabs
Heterogeneity in hybrid CPU/GPU Clusters

Traditional nodes

Expensive GPUs

Speed-up rates are distinct

Intel E5 2.4Ghz CPU vs. Nvidia K80 GPU
Overload if most users prefer GPUs

Expensive GPUs are overloaded while CPUs are under-utilized

Let’s explore some solutions
Join the Shortest Queue (JSQ)

Processing times (GPU, CPU)

<table>
<thead>
<tr>
<th></th>
<th>GPU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>J2</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>J3</td>
<td>35</td>
<td>150</td>
</tr>
<tr>
<td>J4</td>
<td>50</td>
<td>160</td>
</tr>
</tbody>
</table>

JSQ does not consider processing times

Optimal

-69% makespan
-54% avg. compl. time
Shortest Job First (SJF)

Processing times (GPU, CPU)

- **J1**: (10, 20)
- **J2**: (15, 25)
- **J3**: (20, 100)
- **J4**: (20, 90)

SJF does not consider speed-up rates

Optimal

- 75% makespan
- 60% avg. compl. time

SJF does not consider speed-up rates
AlloX – Minimize Avg. Completion Time

Convert the scheduling & placement

Jobs: J1, J2, J3

into min-cost bipartite matching

solved in polynomial time
AlloX – Maintains Fairness for interchangeable resources

User A may not be happy if we keep putting him on CPU.

Idea: Prioritize users with low fairness scores $F$ who run jobs on the unfavorable resources

<table>
<thead>
<tr>
<th>User</th>
<th>$F$</th>
<th>GPU</th>
<th>CPU</th>
<th>$F-1 &lt; F-2$</th>
<th>$F-1 &gt; F-2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>$F-1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 2</td>
<td>$F-2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AlloX System

Estimation Tool
- Sample the jobs
- Estimate the processing times

Processing times
- CPU configuration
- GPU configuration

Scheduler
- **Fairness**: Pick the set of users with least fair scores
- **Scheduling**: Decide to place jobs on CPUs or GPUs.

Resource Placer
- Configure a job to run on CPU or GPU
- kubelet
- GPUs
- CPUs

Jobs

Placement constraints
Performance of AlloX

**DRF**: Dominant Resource Fairness + FIFO
Resource configurations are fixed

**ES**: Equal Share + SJF
Keep filling the available resources

**SRPT**: Shortest Remaining Processing Time
Impractical switching between CPU&GPU

AlloX reduces up to 95% avg. completion time

TensorFlow CNN benchmarks
AlloX: Compute Allocation in Hybrid Clusters

Tan N. Le
Xiao Sun
Mosharaf Chowdhury
Zhenhua Liu

tnle@cs.stonybrook.edu