CS 744: GANDIVA

Shivaram Venkataraman

Fall 2019
ADMINISTRIVIA

- Course project proposal
- Midterm
**Bismarck**
- Supervised learning, Unified Interface
- Shared memory, Model fits in memory

**Parameter Server**
- Large datasets, large models (PB scale)
- Consistency model, Fault tolerance

**Tensorflow**
- Need for flexible programming model
- Dataflow graph, Heterogeneous accelerators

**Ray**
- Reinforcement learning applications
- Actors and tasks, Local and global scheduler

---

*Inference*

*Chipper*
Scalable Storage Systems

Datacenter Architecture

Resource Management

Computational Engines

Applications

Machine Learning

SQL

Streaming

Graph

Applications

SQL

Streaming

Graph

Computational Engines

Scalable Storage Systems

Resource Management

Datacenter Architecture

Applications

SQL

Streaming

Graph

Computational Engines

Scalable Storage Systems

Resource Management

Datacenter Architecture
MACHINE LEARNING WORKFLOW?

```python
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 10, kernel_size=5)
        self.conv2 = nn.Conv2d(10, 20, kernel_size=5)
        self.conv2_drop = nn.Dropout2d()
        self.fc1 = nn.Linear(320, 50)
        self.fc2 = nn.Linear(50, 10)
```
SHARED ML CLUSTERS

- Maintenance
- Power/Cooling

Rack

$\text{PS}_0 \rightarrow \text{exclusive}

\rightarrow \text{multi-tenant}

\rightarrow \text{enqueue}

\rightarrow \text{if } J_1 \text{ takes 1 day to run}

\rightarrow \text{and } J_2 \text{ waits for 1 day}

\rightarrow \text{W}_0 \rightarrow \text{W}_1

\rightarrow \text{W}_2 \rightarrow \text{Efficiency of cluster hardware is low}

\rightarrow \text{W}_3 \rightarrow \text{PS}_0$
Feedback-driven exploration

Hyper parameter search

\[ \begin{align*}
  h_0 & \text{ min } 0, \text{ max } 100 \\
  h_2 & \text{ min } 2, \text{ max } 10 \\
  h_k & \text{ min } K, \text{ max } K
\end{align*} \]

\[ \text{min } f(x) + \| \nabla \text{all}_2 \]\n
\[ \downarrow \text{ value}, \text{ regularization} \]

\[ \rightarrow \text{ learning rate} \]

\[ \text{Early stopping} \]

\[ \text{truncates execution} \]

\[ \text{of suboptimal hyper parameters} \]
Figure 1: Intra-server locality. Figure 2: Inter-server locality.
INTRA JOB PREDICTABILITY

while
  Sample
  Compute \( \forall f \)
  Aggregate Sync/Async
  end
  Predictability → useful
  Heterogeneity

(a) ResNet50/Imagenet
(b) GNMT/WMT’14 En-De
MECHANISMS (1)

1. Suspend-Resume

- $J_2 \rightarrow$ suspend job 1, place job 2 on GPUs for time quantum \(= 1\) min
- \(\rightarrow\) suspend job 2, resume Job 1
- \(\rightarrow\) the intra-job predictability to suspend MB finishes

2. Migration

- \(\rightarrow\) more tasks to improve locality
- or improve utilization
- \(\rightarrow\) copy state old mc and new mc
- \(\rightarrow\) copy state 1GB = longer time period
MECHANISMS (2)

3. Grow-shrink

→ If GPUs idle, can we give them to existing tasks

\[ b = 256, \ 2 \text{ GPUs} \]

→ 2 GPUs when they become free

4. Profiling

→ Mem usage

→ Mini batch time taken

\[ \text{MB latency (ms)} \]

\[ \text{Time} \]

look up table of profiles

locality
SCHEDULING POLICY

 Goals

  early feedback
  cluster efficiency
  cluster-level fairness? \[\rightarrow \text{Non-goal}\]

 Two modes

  Reactive
  Introspective
**REACTIVE MODE**

React to events
- Job arrivals, departures, failures

Hierarchical Preference
- Nodes with same “affinity”
- Nodes with “different affinity”
- Nodes with “no affinity”
- Suspend-resume …
INTROSPECTIVE MODE

Monitor and optimize placement of jobs periodically

Actions

Packing  

Migration

Grow-shrink

Over subscribe resources by using predictability

Priorities: High, 2 time quantum

undo mis
DISCUSSION

https://forms.gle/aHYbNcTFdGjtXefj9
What are some guarantees provided by Mesos that are not provided by Gandiva? Explain with an example.

Mesos - "resource guarantee" - strategy proof, while Gandiva. Jobs can inflate mini batch size - Fairness → Making incentive
Are mechanisms in Gandiva also useful in a cluster running Apache Spark jobs? Provide one example either for or against

- Migration is not that useful because of backup tasks?
- Need to transfer data, RDD or shuffle data
- Suspend/Resume may not be useful because of lack of predictability
- PageRank, Logistic Regression do have predictability within the DAG
- Naturally happening task boundaries

Can be useful for large shuffles
NEXT STEPS

New module on SQL!
Course project introductions
Midterm