CS 744: PYWREN

Shivaram Venkataraman

Fall 2022
ADMINISTRIVIA

Project checkins due Nov 23rd
Poster presentation: Dec 13th
Final report: Dec 20th

Project grade breakdown
  Intro: 5%
  Mid-semester checkin: 5%
  Poster: 10%
  Final Report: 10%
Scalable Storage Systems

Datacenter Architecture

Resource Management

Computational Engines

Applications

Machine Learning SQL Streaming Graph
NEW DATA, HARDWARE MODELS
Serverless Computing

Compute Accelerators

Infiniband Networks

Non-Volatile Memory
SERVERLESS COMPUTING
### MOTIVATION: USABILITY

<table>
<thead>
<tr>
<th>Name</th>
<th>API Name</th>
<th>vCPU (Min)</th>
<th>Memory (GB)</th>
<th>Storage (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C5 High-CPU Double Extra Large</td>
<td>c5d.2xlarge</td>
<td>8</td>
<td>16.0</td>
<td>200 NVMe SSD</td>
</tr>
<tr>
<td>C5 High-CPU Extra Large</td>
<td>c5d.xlarge</td>
<td>4</td>
<td>8.0</td>
<td>100 NVMe SSD</td>
</tr>
<tr>
<td>M6a.2xlarge</td>
<td>m6a.2xlarge</td>
<td>96</td>
<td>384.0</td>
<td>EBS only</td>
</tr>
<tr>
<td>MSDN Extra Large</td>
<td>m5dn.xlarge</td>
<td>4</td>
<td>16.0</td>
<td>150 NVMe SSD</td>
</tr>
<tr>
<td>C5 High-CPU Metal</td>
<td>c5.metal</td>
<td>96</td>
<td>192.0</td>
<td>EBS only</td>
</tr>
<tr>
<td>C6A Eight Extra Large</td>
<td>c6a.8xlarge</td>
<td>32</td>
<td>64.0</td>
<td>EBS only</td>
</tr>
<tr>
<td>D3EN 12xlarge</td>
<td>d3en.12xlarge</td>
<td>48</td>
<td>192.0</td>
<td>335520 GB (24 * 13980 GB HDD)</td>
</tr>
<tr>
<td>D3EN Eight Extra Large</td>
<td>d3en.8xlarge</td>
<td>32</td>
<td>128.0</td>
<td>223680 GB (16 * 13980 GB HDD)</td>
</tr>
<tr>
<td>RSAD 16xlarge</td>
<td>r5ad.16xlarge</td>
<td>64</td>
<td>512.0</td>
<td>2400 GB (4 * 600 NVMe SSD)</td>
</tr>
<tr>
<td>MSA Double Extra Large</td>
<td>m5a.2xlarge</td>
<td>8</td>
<td>32.0</td>
<td>EBS only</td>
</tr>
<tr>
<td>MSN Metal</td>
<td>m5n.metal</td>
<td>96</td>
<td>384.0</td>
<td>EBS only</td>
</tr>
<tr>
<td>C6ID Eight Extra Large</td>
<td>c6id.8xlarge</td>
<td>32</td>
<td>64.0</td>
<td>1900 NVMe SSD</td>
</tr>
<tr>
<td>MSAD Double Extra Large</td>
<td>m5ad.2xlarge</td>
<td>8</td>
<td>32.0</td>
<td>300 NVMe SSD</td>
</tr>
<tr>
<td>M6ID Extra Large</td>
<td>m6id.xlarge</td>
<td>4</td>
<td>16.0</td>
<td>237 NVMe SSD</td>
</tr>
</tbody>
</table>
When to use the Cloud?

Data
- Large amounts of data. Can’t store locally
- Shared data across users
- Long term storage

Compute
- Need lots of CPUs for short jobs
- Varying compute needs (Yes)
- No admin costs (Yes)

Why is there no “cloud button”?
ABSTRACTION LEVEL?

Application

Compute Framework

Hardware

Logistic Regression

Spark

Amazon EC2
CloudLab
Private Cluster

…

Application

Compute Framework
“SERVERLESS” COMPUTING

300-900 seconds single-core

512-10240 MB in /tmp

3-10GB RAM

Python, Java, node.js, Ruby, Go etc.

Support for containers
import pywren
import numpy as np

def addone(x):
    return x + 1

wrenexec = pywren.default_executor()
xlist = np.arange(10)
futures = wrenexec.map(addone, xlist)

print [f.result() for f in futures]

The output is as expected:

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
PYWREN: HOW IT WORKS

future = runner.map(fn, data)

future.result()

your laptop  the cloud
future = runner.map(fn, data)

Serialize func and data
Put on S3
Invoke Lambda

pull job from s3
download anaconda runtime
python to run code
pickle result
stick in S3

result

poll S3
unpickle and return

your laptop

the cloud
STATELESS FUNCTIONS: WHY NOW?

What are the trade-offs?

<table>
<thead>
<tr>
<th>Storage Medium</th>
<th>Write Speed (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSD on c3.8xlarge</td>
<td>208.73</td>
</tr>
<tr>
<td>SSD on i2.8xlarge</td>
<td>460.36</td>
</tr>
<tr>
<td>4 SSDs on i2.8xlarge</td>
<td>1768.04</td>
</tr>
<tr>
<td>S3</td>
<td>501.13</td>
</tr>
</tbody>
</table>
MAP AND REDUCE?

Input Data

Output Data

\( \lambda \)

\( \lambda \)
Use lambdas to run “workers”

Parameter server as a service?
WHEN SHOULD WE USE SERVERLESS?

Yes!

Maybe not?
Motivation: Usability of big data analytics
Approach: Language-integrated cloud computing

Features
- Breakdown computation into stateless functions
- Schedule on serverless containers
- Use external storage for state management

Open question on scheduling, overheads
DISCUSSION

https://forms.gle/GFIkkME52tvDRdDC8
Consider you are a cloud provider (e.g., AWS) implementing support for serverless. What could be some of the new challenges in scheduling these workloads compared to schedulers we have studied in this class? How would you go about addressing them?
Happy Thanksgiving!

Next steps:
- Mid-semester project check-in, Nov 23rd
- After break: Owl, TPU papers
- Midterm 2, Dec 6th