Cloud Parallelism and Microservices for Science

Dennis Gannon
Prof. of CS Emeritus, School of Informatics and Computing
Indiana University
&
Microsoft Research (retired)
www.esciencegroup.com
The Cloud Data Center Evolution

• Early days: 2005
  • Very simple servers
  • Network outward facing poor interconnect

• 2008-2016
  • Software defined networks
  • Special InfiniBand sub networks
  • Many different server types
    • 2 cores to 32 cores to GPU accelerations
  • Efficiency experiments
    • Geothermal, wind, wave
    • Prefab clusters in shipping containers

• 2017
  • Azure FPGA accelerated mesh
  • Google Tensor Processing Unit
  • Facebook – Open Compute Project
  • ARM based servers
How to Scale in the Cloud: Models of Parallelism

- Classic HPC
  - SPMD MPI programming
- MapReduce
  - Hadoop style
- Graph Execution
  - Spark and streaming systems
- Microservices
  - Similar to actor model
Classic HPC

- AWS CloudFormation Cluster
  - Fill out CfnCluster template
  - Use aws command line to submit
  - Log into head node

- Azure create a slurm cluster
  - See Azure slurm tutorial

- Or use Azure Batch
  - Similar to AWS batch
Map Reduce

- **Map Reduce**
  - Bulk Synchronous Parallel (BSP)
  - Distribute data over many nodes. (Hadoop Distributed File System)
  - Map Task = an operation applied to blocks of data in parallel
  - Reduce Task - when maps are “done” reduce the results to a single result
The Hadoop- Yarn ecosystem

• Yarn is the name of a project containing many elements
• The runtime system is distributed
• Hadoop, Spark run in distributed mode
• Multiple clients can access the resource manager
• Jupyter and Zeppelin are interactive clients

• HDFS is the Hadoop File system
• Distributed over data node servers
• Files are blocked, distributed and replicated
• Files are write-once.
Graph Parallel Computation

• Graph Parallel
  • The data is in distributed arrays or streams.
  • build a data flow graph of the algorithms functions.
  • The graph is compiled into parallel operators that are applied to the distributed data structures.

• Examples
  • Spark data analytics
  • Stream analytics with Kafka, Storm, Heron, etc.
  • Deep Learning
    • Tensorflow from Google
    • CNTK from Microsoft
Graph computation example: Spark

- A simple map reduce: Compute
- For $n = 10,000,000$
- In Spark on Python is:

```python
import numpy as np
ar = np.arange(n)# an array from 0 to 9999999
numpart = 100
rdd = sc.parallelize(ar,numpart)
x = rdd.map(lambda i: 1.0/(i+1)**2).
.reduce(lambda a,b: a+b)
print(“x=%f”%x)
print(“pi**2/6=%f”%(np.pi**2/6))
```

1.644934
1.644934

Spark Resilient Distributed Dataset (RDD)

Graph executed on distributed cluster

Value returned to python
Microservices

• Cloud-native computation
  • Divide a computation into small, mostly stateless components that can be
    • Easily replicated for scale
    • Communicate with simple protocols
  • Computation is as a swarm of communicating workers.

• Examples
  • Netflix, Google Docs, Azure services, eBay, Amazon, the UK Government Digital Service, Twitter, PayPal, Gilt, Bluemix, Soundcloud, The Guardian
  • JetStream Genomics Docker swarm to spinup container instance of Galaxy for users on demand
Microservices

• Typically run as containers using a service deployment and management service
  • Amazon EC2 Container Service
  • Google Kubernetes
  • DCOS from Berkeley/Mesosphere
  • Docker Swarm

• Major advantage:
  • Resilience – designed for continuous application operation
  • Deployment can be modified on-the-fly (dev-ops)
Demo Example

• Processing Document streams
  • Lots of RSS feeds describing recent scientific documents
  • Let’s classify them by topic
    • Physics, Math, CS, Biology, Finance, ...
    • Then by subtopics
  • By reading the abstracts and using a little machine learning.
    • Abstracts from Cornell Library ArXiv

• Building application steps
  1. create a service cluster in the cloud
  2. define services and interfaces
  3. build each as an individual container
  4. Create task descriptors
Title: Controls for a Pulsed Ion Accelerator Using Apache Cassandra

ArXiv classification: physics.acc-ph

Abstract: We report on updates to the accelerator controls for the Neutral Drift Compression Experiment II, a pulsed accelerator for heavy ions. The control infrastructure is built around a LabVIEW interface combined with an Apache Cassandra (No-SQL) backend for data archiving. Recent upgrades added the storing and retrieving of device settings into the database, as well as adding ZMQ as a message broker that replaces LabVIEW's shared variables. Converting to ZMQ also allows easy access using other programming languages, such as Python.

Predictor returns guesses from 5 different ML algorithms
  • (compsci, compsci, compsci, ??, Physics)
Demo - A simplified version using Amazon AWS and Azure Together

• Create
  • An instance of a message Queue based on AWS SQS
  • An dynamoDB table BookTable
  • An Azure table called BookTable

• Create 3 services
  • Predictor – one parameter (port)
  • TableServiceAWS
  • TableServiceAzure

• 1st step: create a AWS elastic container service cluster
Create a cluster

<table>
<thead>
<tr>
<th>Cluster name*</th>
<th>tutorial-cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC2 instance type*</td>
<td>m4.large</td>
</tr>
<tr>
<td>Number of instances*</td>
<td>3</td>
</tr>
<tr>
<td>EC2 Ami ID*</td>
<td>amzn-ami-2016.09.0-amazon-ecs-optimized [ami-022b9262]</td>
</tr>
<tr>
<td>EBS storage (GiB)*</td>
<td>22</td>
</tr>
<tr>
<td>Key pair</td>
<td>escience1</td>
</tr>
</tbody>
</table>

You will not be able to SSH into your EC2 instances without a key pair. You can create a new key pair in the EC2 console.
The microservice containers

• **Predictor-new**
  • A docker container that
  • takes one parameter at startup
    • The IP port of a service that handles the output
  • Runs a loop that pull abstracts from a queue and applies some machine learning algorithms to classify the abstract
  • Sends the result to the output handling service

• **TableserviceAzure**
  • A webservice that waits for a classified document and saves the result in an Azure table

• **TableserviceAWS**
  • Identical to TableserviceAzure except it has the code to save the result to the AWS dynamoDB

• Each services is a short python program
Code to create a service

```python
response = client.register_task_definition(
    family='predictorAzure',
    networkMode='bridge',
    taskRoleArn= 'arn:aws:iam::066301190734:role/mymicroservices',
    containerDefinitions=[
        {
            'name': 'predictorAzure',
            'image': 'dbgannon/predictor-new',
            'cpu': 20,
            'memoryReservation': 400,
            'essential': True,
            'command': ['8055']
        },
    ],
)

response = client.create_service(
    cluster='tutorial-cluster',
    serviceName='predictorAzure',
    taskDefinition='predictorAzure:1',
    desiredCount=1, deploymentConfiguration={
        'maximumPercent': 100,
        'minimumHealthyPercent': 50
    }
)```
Go to Demo
Microservice Science Applications

• Experiment event stream analysis
  • Astronomy, environmental monitors, particle physics, weather events

• Large scale many-task computations
  • Meta-genomics, protein folding

• Complex workflows
  • Experimental quality control with lots of filters and checks
Parting Thoughts

• The cloud data centers are designed to scale
  • Traditional HPC MPI programming is now possible, but if you need 10,000 cores a Cray is better.

• The cloud excels at distributed interactive computation
  • Spark with Jupyter is a good example

• MapReduce and Graph models are well supported in the cloud

• Microservices provide a means to support very large scale parallelism in continuously running applications.
A new book

Cloud Computing for Science and Engineering

- By Ian Foster and Dennis Gannon
- Published by MIT Press
- Due out in November 2017 (as SC)
Exercises

• If you have Docker installed
  • run dbgannon/tutorial
    run -it --rm -p 8888:8888 dbgannon/tutorial
  • You should see the spark.ipynb in the notebooks. Fire it up. Make sure it is running with kernel python 2 and shutdown other big apps. This needs memory!

• For something different: Signup for https://notebooks.azure.com
  • Do the twitter analysis demo