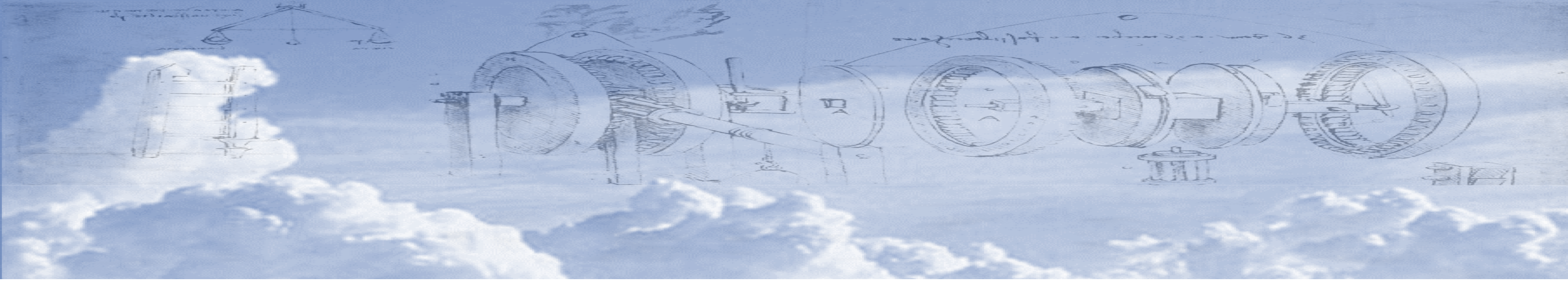
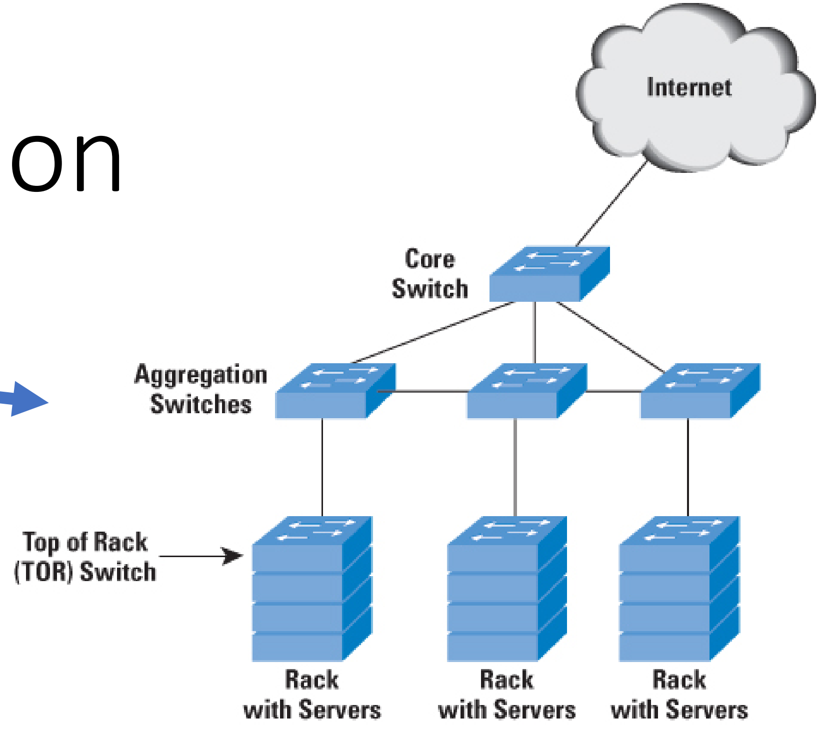


Cloud Parallelism and Microservices for Science

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



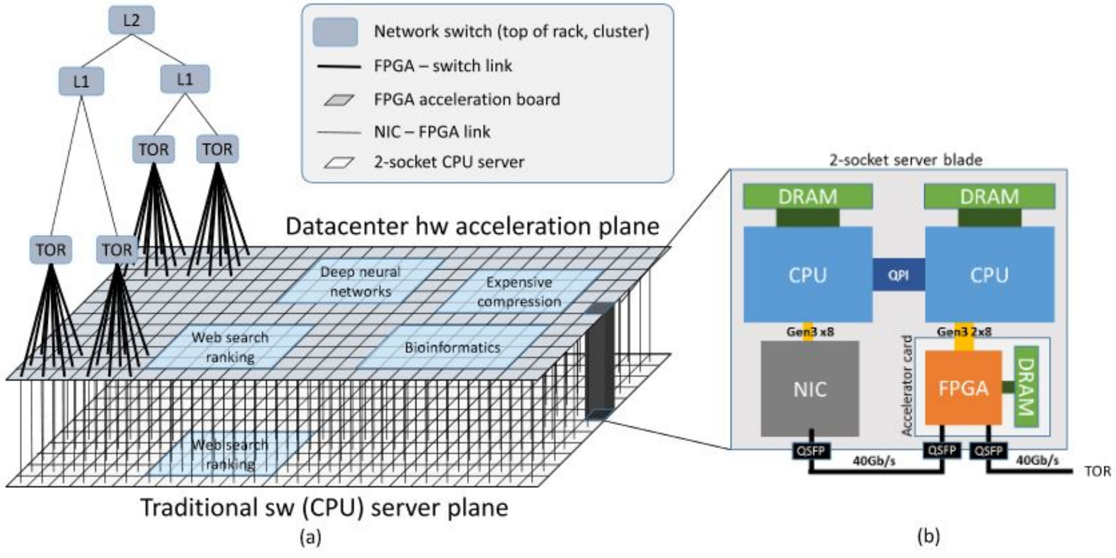


Fig. 1. (a) Decoupled Programmable Hardware Plane, (b) Server + FPGA schematic.

Azure and AWS Global Data Center Network

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

Deploy a slurm cluster

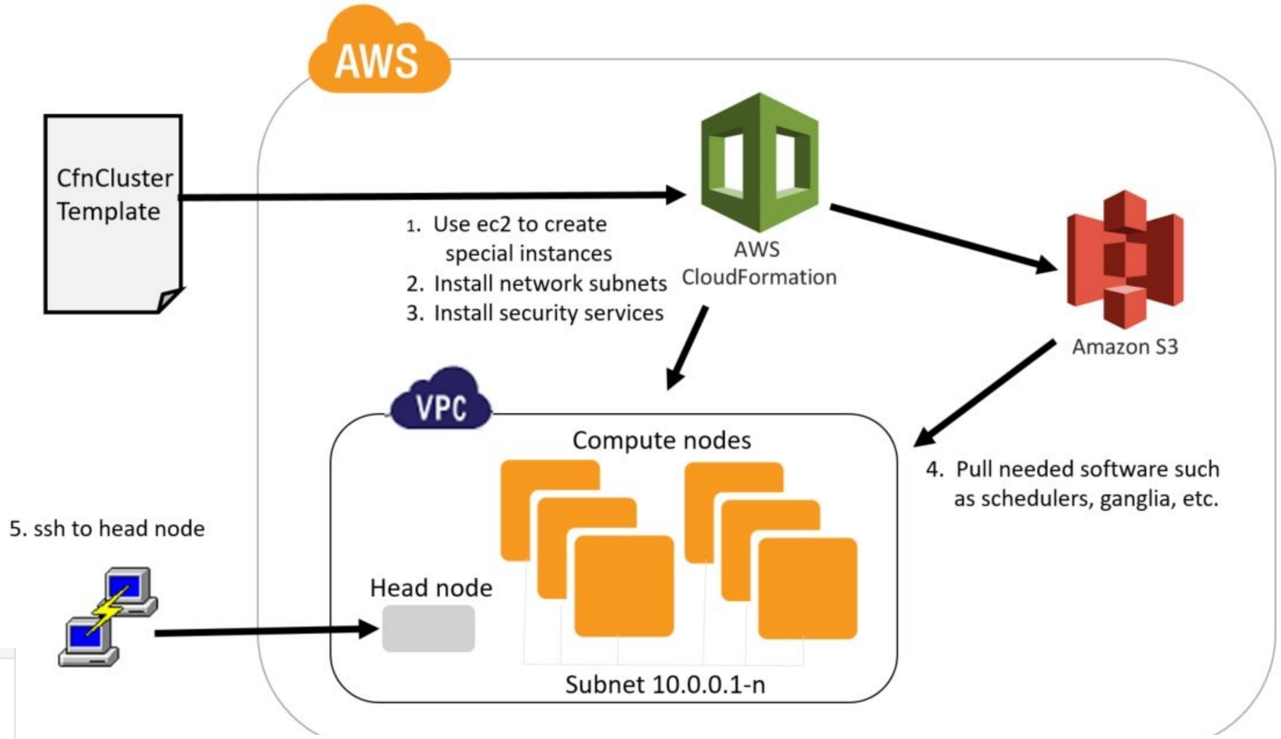
Deploy to Azure | Visualize

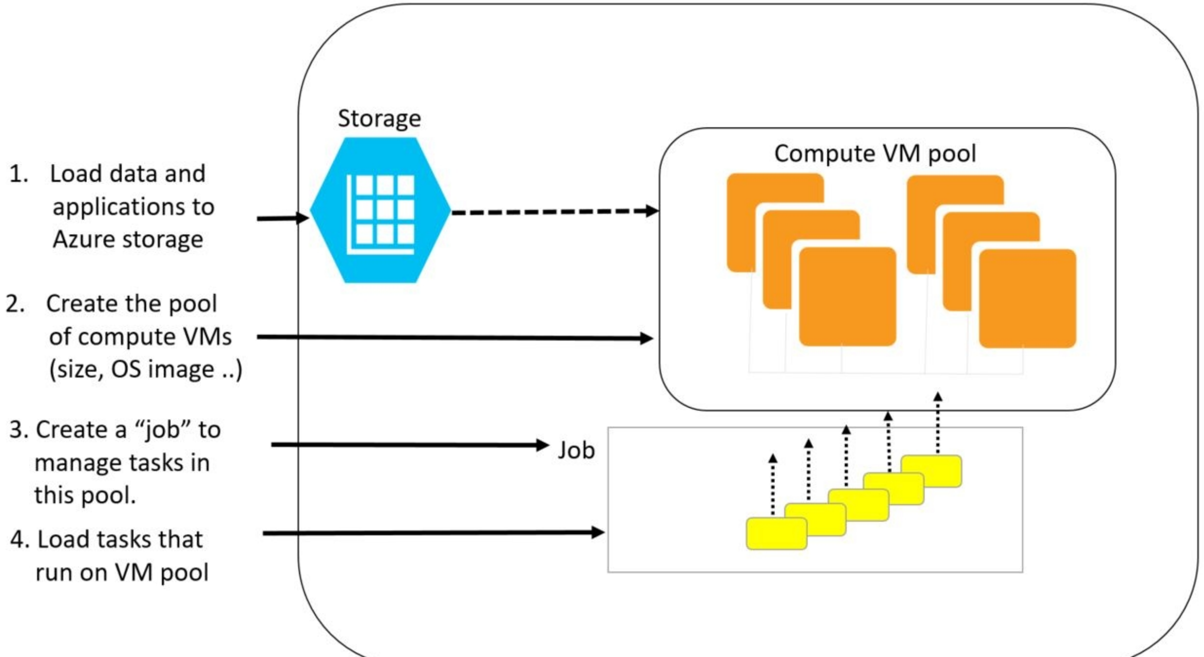
1. Fill in the 3 mandatory parameters - public DNS name, a storage account to hold VM image, and admin user password.
2. Fill in other info and click "OK".

Using the cluster

Simply SSH to the master node and do a srun! The DNS name is ***dnsName.location***.cloudapp.azure.com, for example, yidingslurm.westus.cloudapp.azure.com.

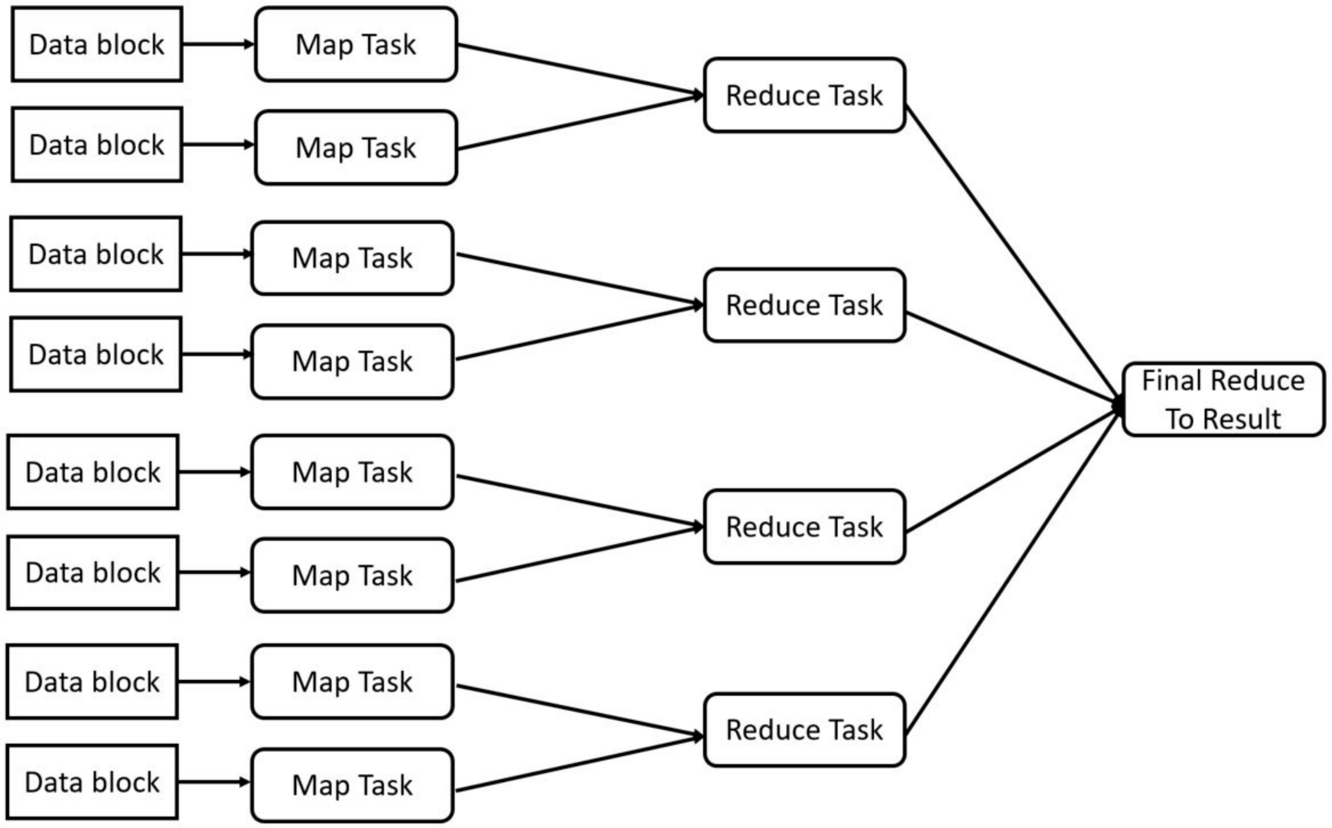
- Or use Azure Batch
  - Similar to AWS batch





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

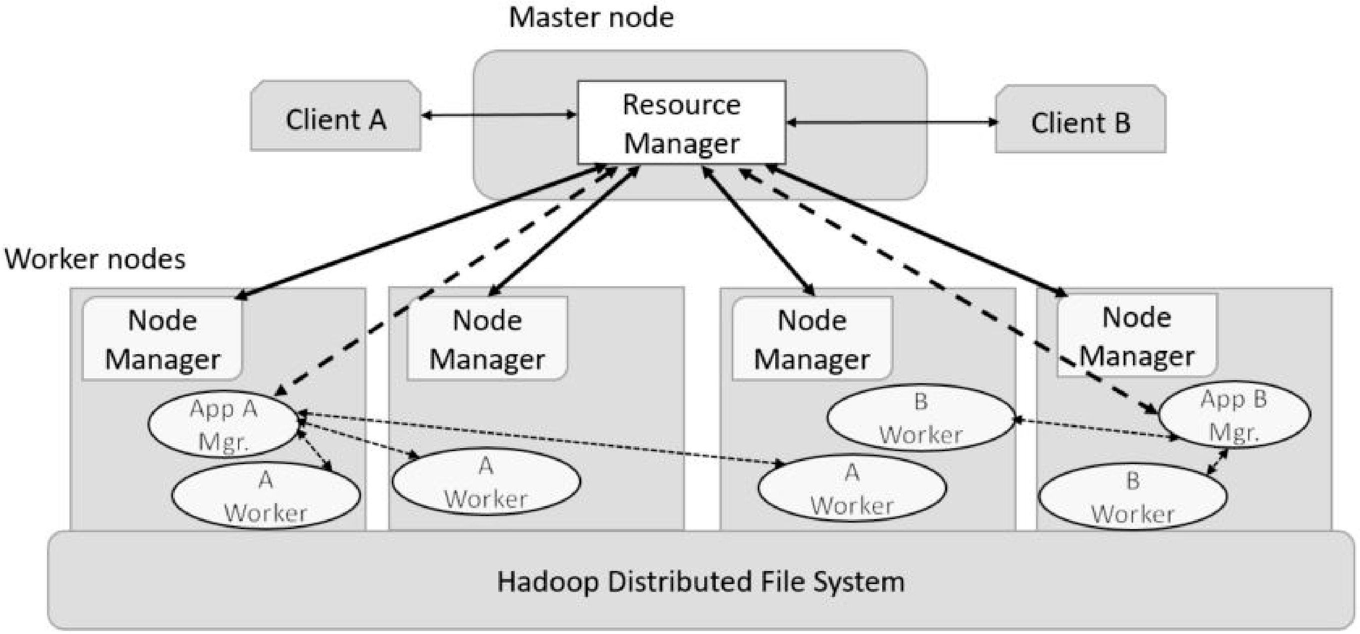


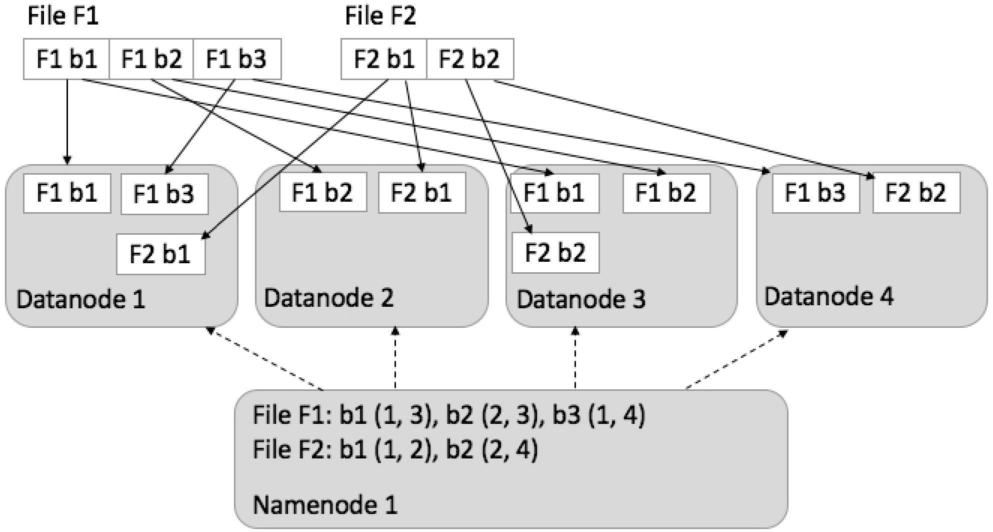
The Hadoop- Yarn ecosystem

- 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

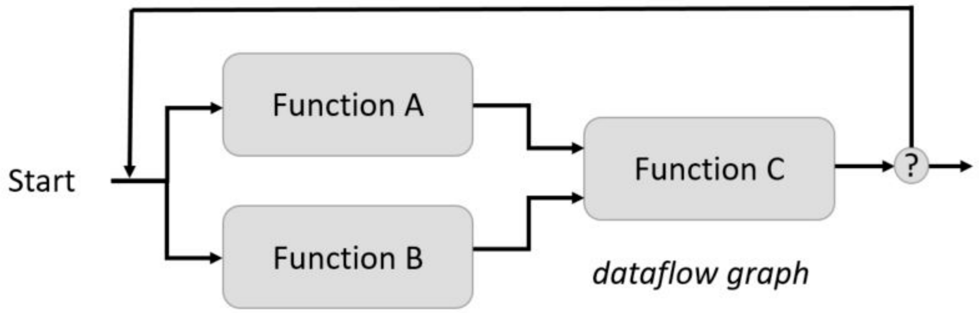
- HDFS is the Hadoop File system
- Distributed over data node servers
- Files are blocked, distributed and replicated
- Files are write-once.

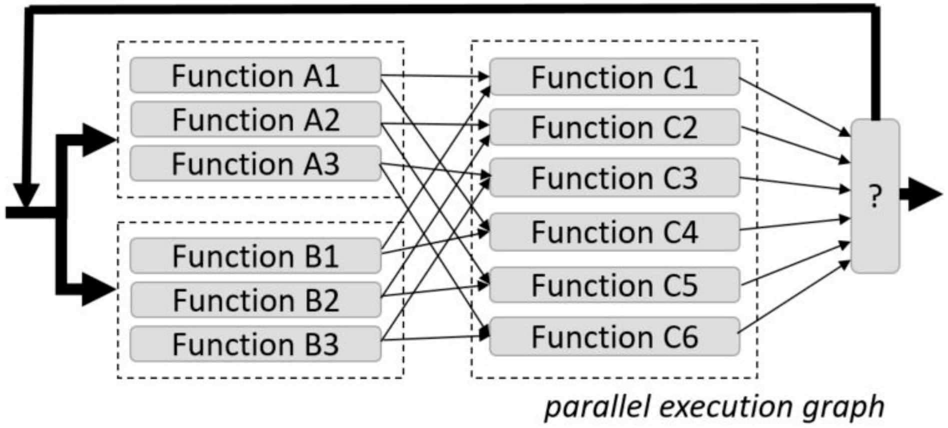




Graph Parallel Computation

- 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

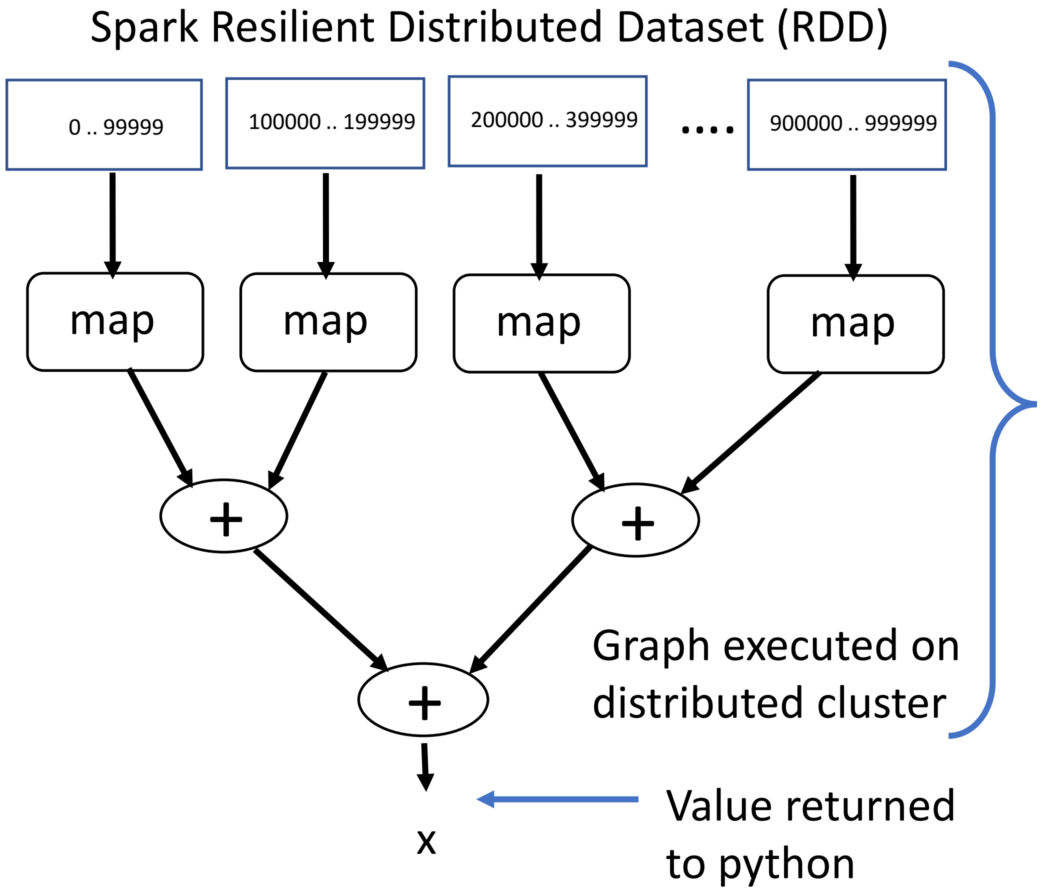




Graph computation example: Spark

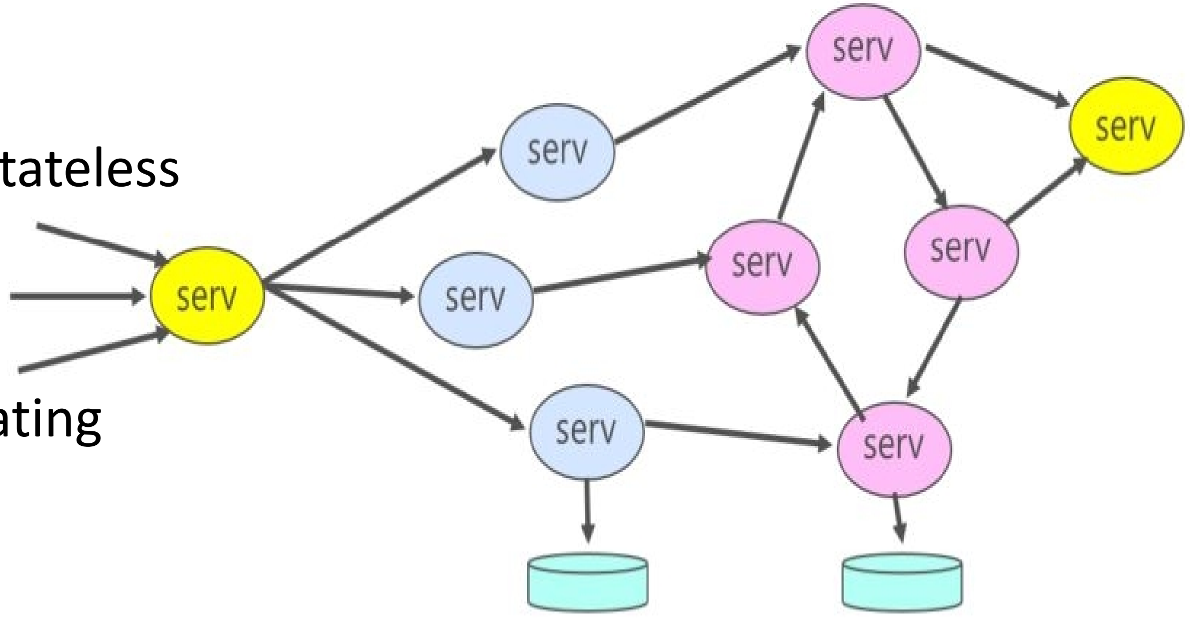
- A simple map reduce: Compute $$\lim_{n->\infty} \sum_{i=1}^{n} \frac{1}{i^2} = \frac{\pi^2}{6}$$
- For n = 10,000,000
- In Spark on Python is:

```
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
```



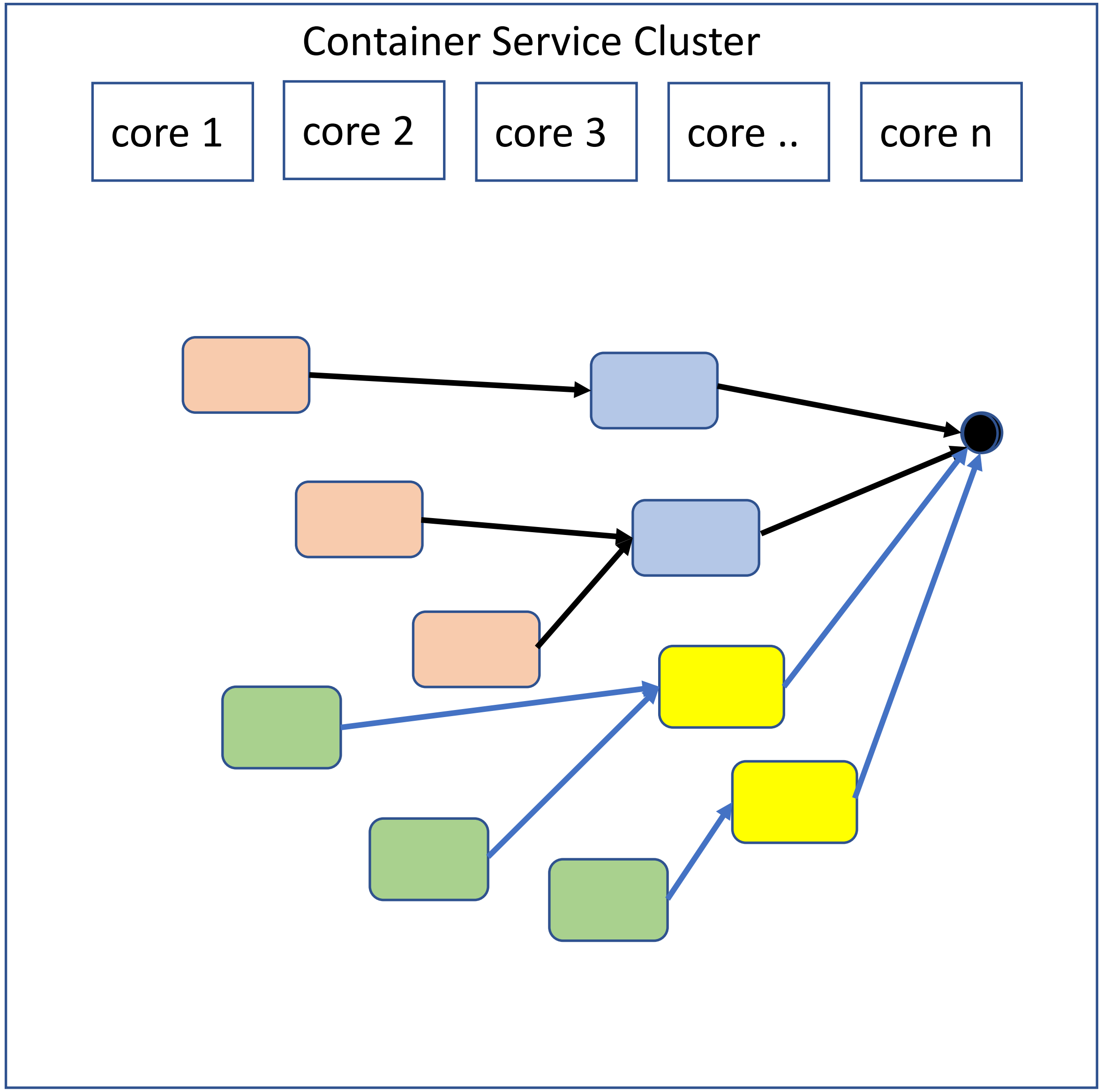
Microservices

- 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

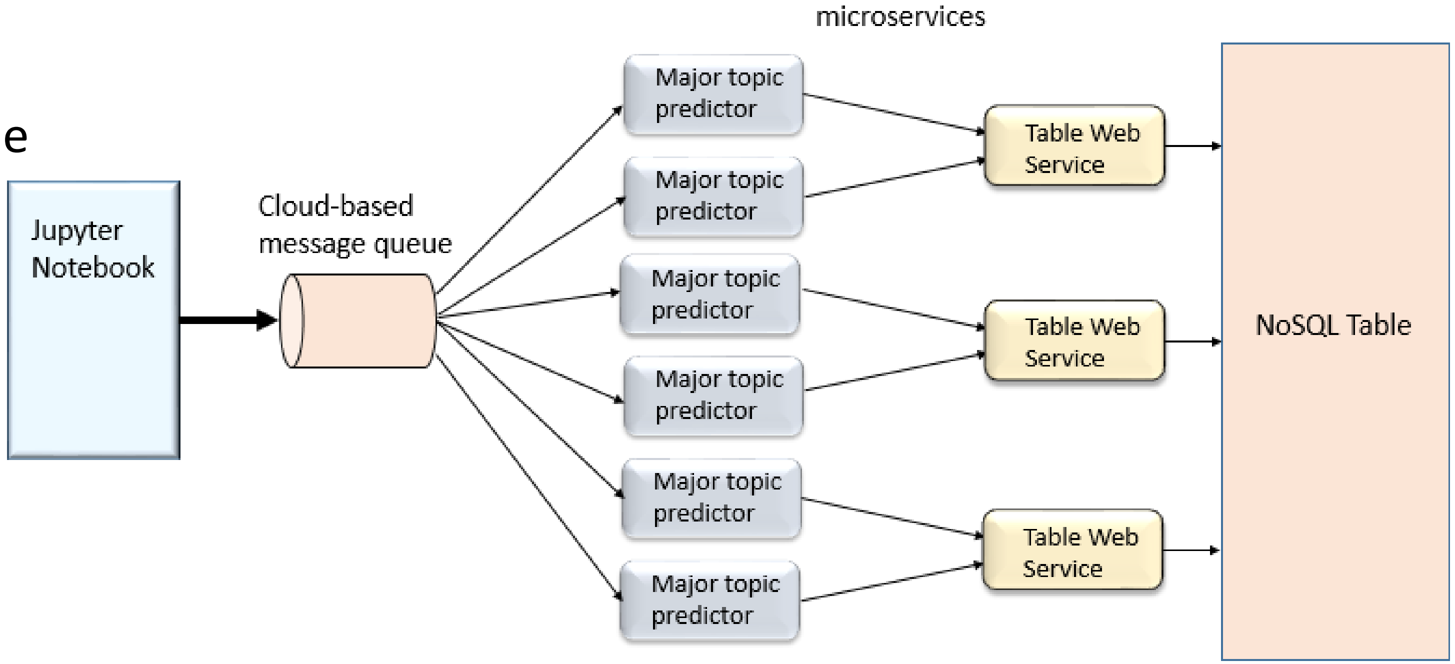
- Typically run as containers using a service deployment and management service
  - Amazon EC2 Container Service
  - Google Kubernetes
  - DCOS from Berkeley/Mesosphere
  - Docker Swarm
- Major advantage:
  - Resilience – designed for continuous application operation
  - Deployment can be modified on-the-fly (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 Cornel Library ArXiv
- Building application steps
  1. create a service cluster in the cloud
  2. define services and interfaces
  3. cuild each as an individual container
  4. Create task descriptors

Document classifier application

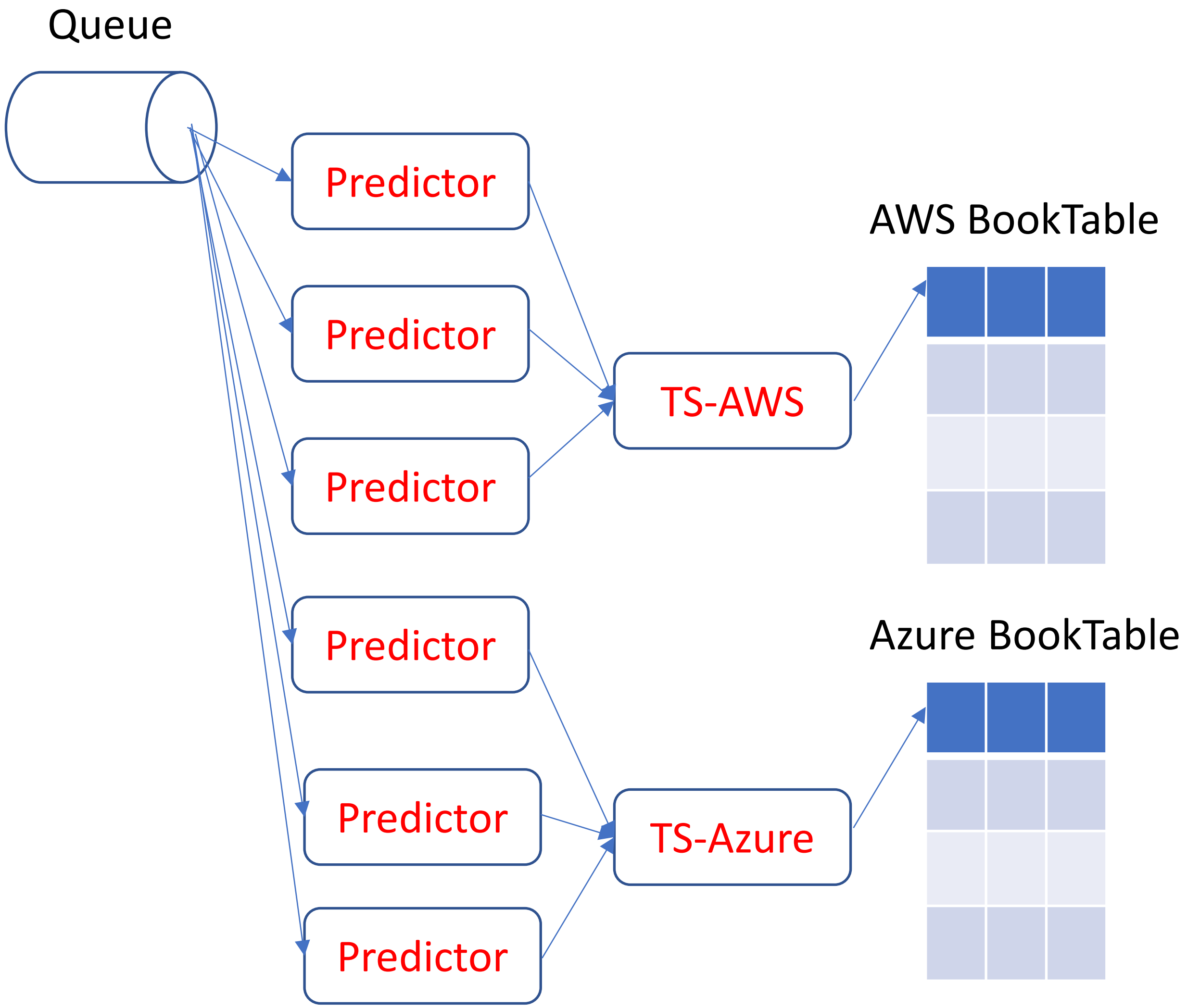


Example document

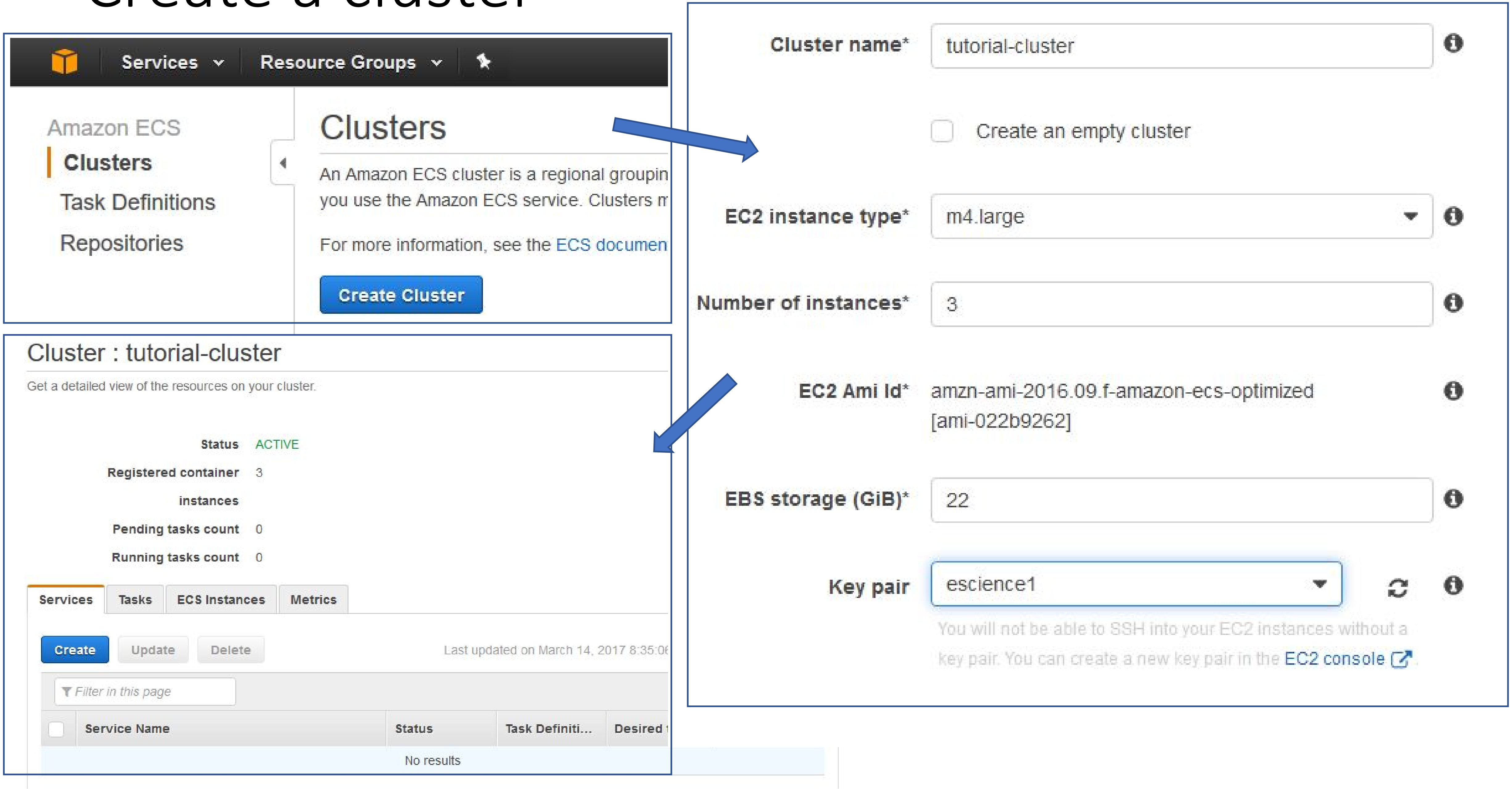
- Title: Controls for a Pulsed Ion Accelerator Using Apache Cassandra
- ArXiv classification: physics.acc-ph
- Abstract: We report on updates to the accelerator controls for the Neutral Drift Compression Experiment II, a pulsed accelerator for heavy ions. The control infrastructure is built around a 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



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

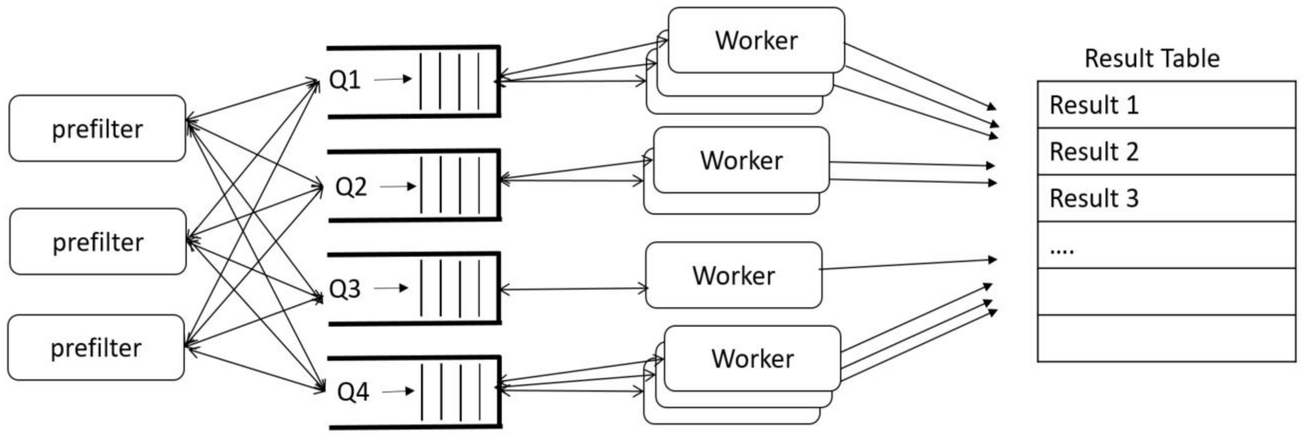
```
response = client.register_task_definition(
    family='predictorAzure',
    networkMode='bridge',
    taskRoleArn= 'arn:aws:iam::066301190734:role/mymicroservices',

    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)
- On line at https://www.Cloud4SciEng.org

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