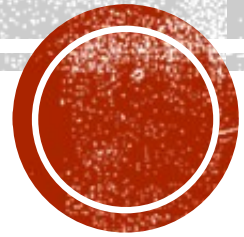
PARALLEL COMPUTING PARADIGMS IN THE CLOUD

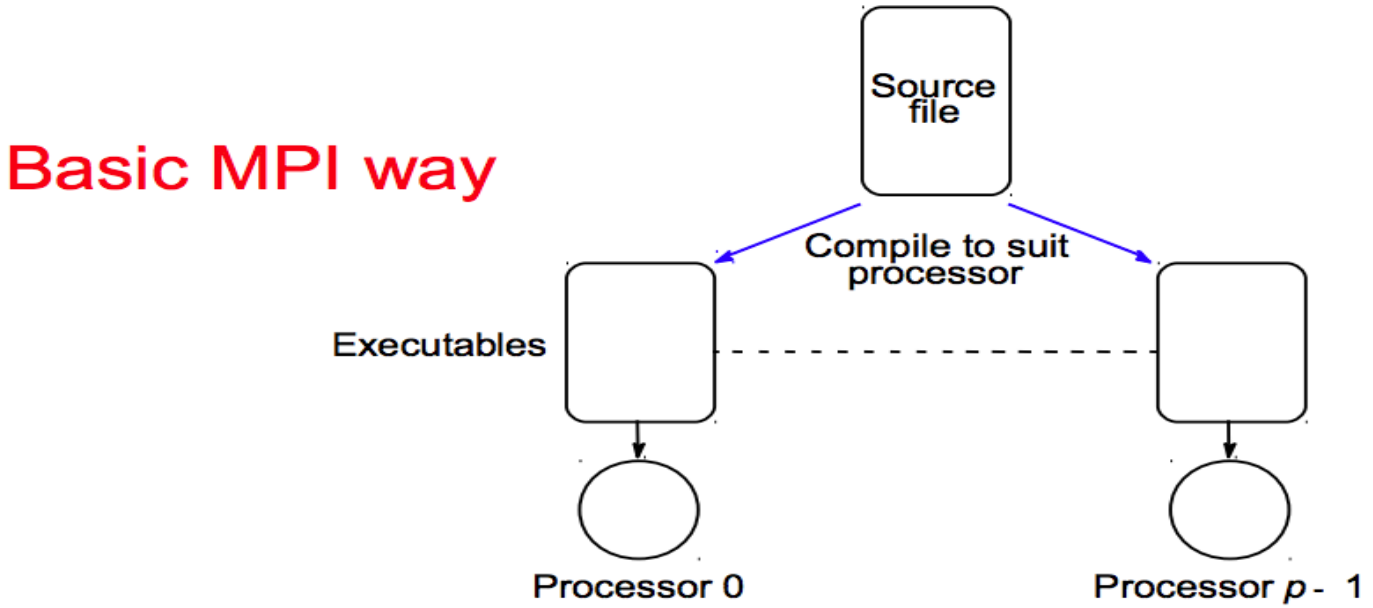


PARADIGMS OF PARALLEL COMPUTING IN THE CLOUD

- Single Program Multiple Data (SPMD)
  - Classic High Performance Computing (HPC)
- Many Task Parallelism
  - A large queue or queues of tasks may be executed in any order
- Bulk Synchronous Parallelism (BSP)
  - Map Reduce
- Graph Execution
  - Spark and streaming systems
- Microservices
  - Computing is perform by one or more actors which communicate via message
- Serverless
  - Focus on application, not the infrastructure

SPMD

- Message Passing Interface (MPI) Programming Model
  - A computation comprises one or more processes that communicate by calling library routines to send and receive messages to other processes
- Same program executed by each processor
- Control statements select different parts for each processor to execute



A SIMPLE SPMD

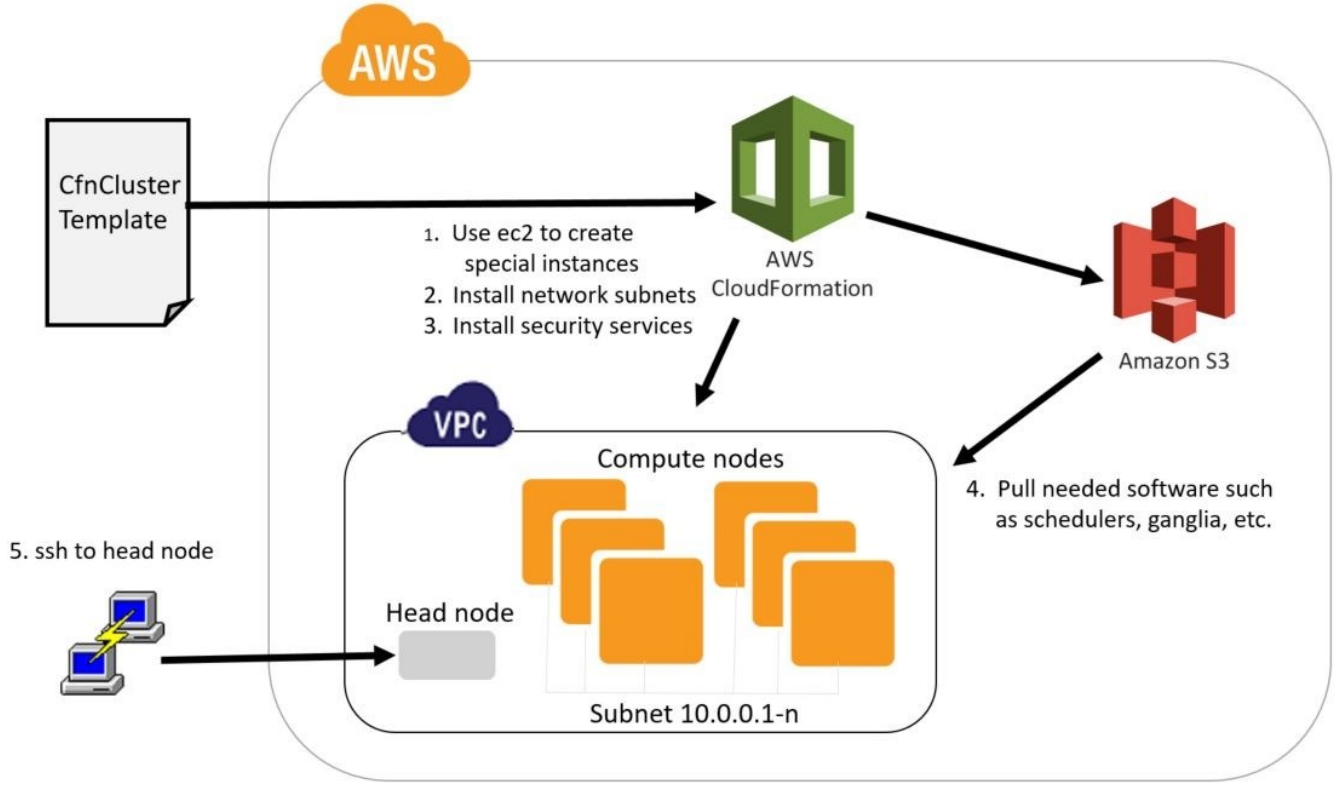
```c
#include <stdio.h>
#include <mpi.h>
#include <stdlib.h>

main(int argc, char **argv)
{
    char hostname[1024];
    gethostname(hostname, 1024);
    printf("%s\n", hostname);
}
```

```
mpicc ip-print.c
srun -n 16 /home/ec2-user/a.out > machines
```

SPMD USING AWS

- Amazon CloudFormation service
  - Enables automated deployment of complex collections of related services, ie.
    - Multiple EC2 instances
    - Load-balancers
    - Special network connecting them
    - Security groups
- AWS CloudFormation Cluster
  - Fill out CfnCluster template
  - Use aws command line to submit
  - Log into head node



SPMD USING AZURE

- Azure's Slurm Cluster service
  - Fill out the template similar to AWS
  - Setup a Slurm Cluster

Deploy a slurm cluster

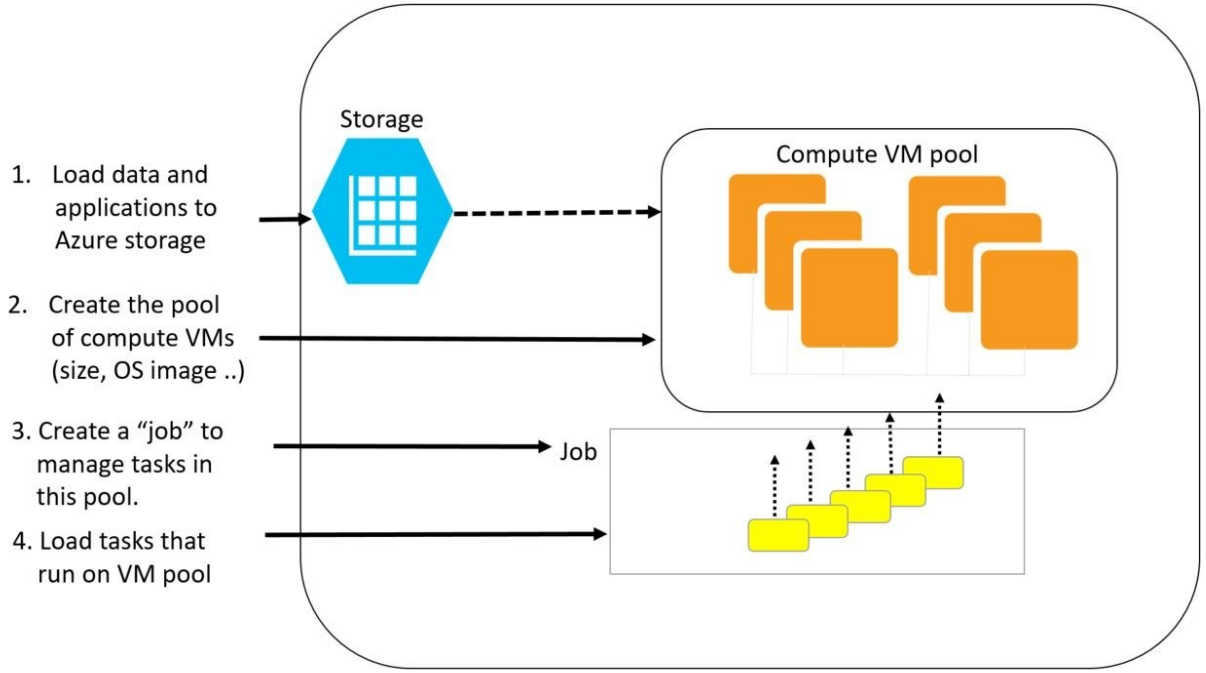
Deploy to Azure | Visualize

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

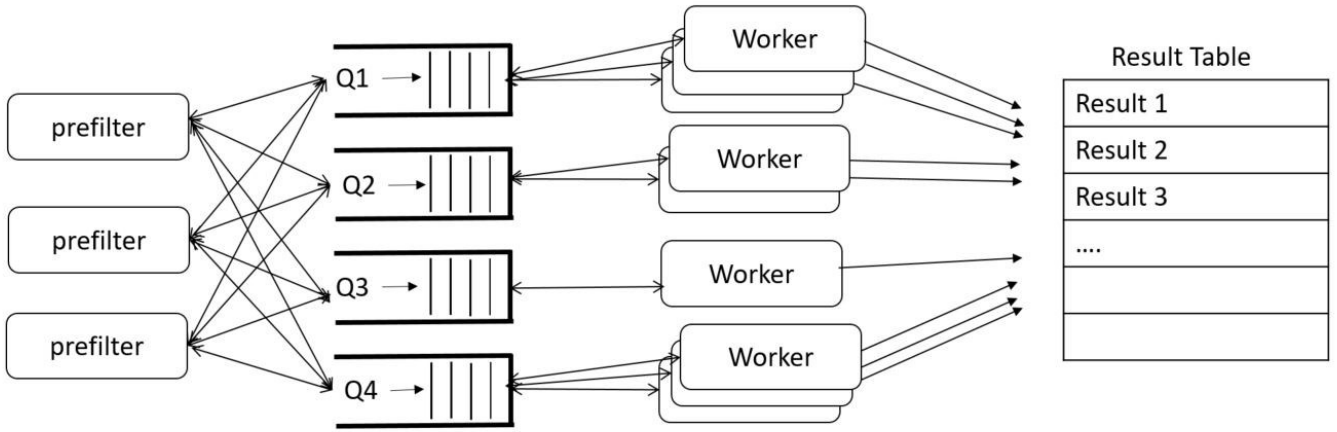
Simply SSH to the master node and do a srun! The DNS name is ***dnsName.location***.cloudapp.azure.com, for example, yidingslurm.westus.cloudapp.azure.com.

- Use Azure Batch
  - Similar to AWS batch



MANY TASK PARALLELISM

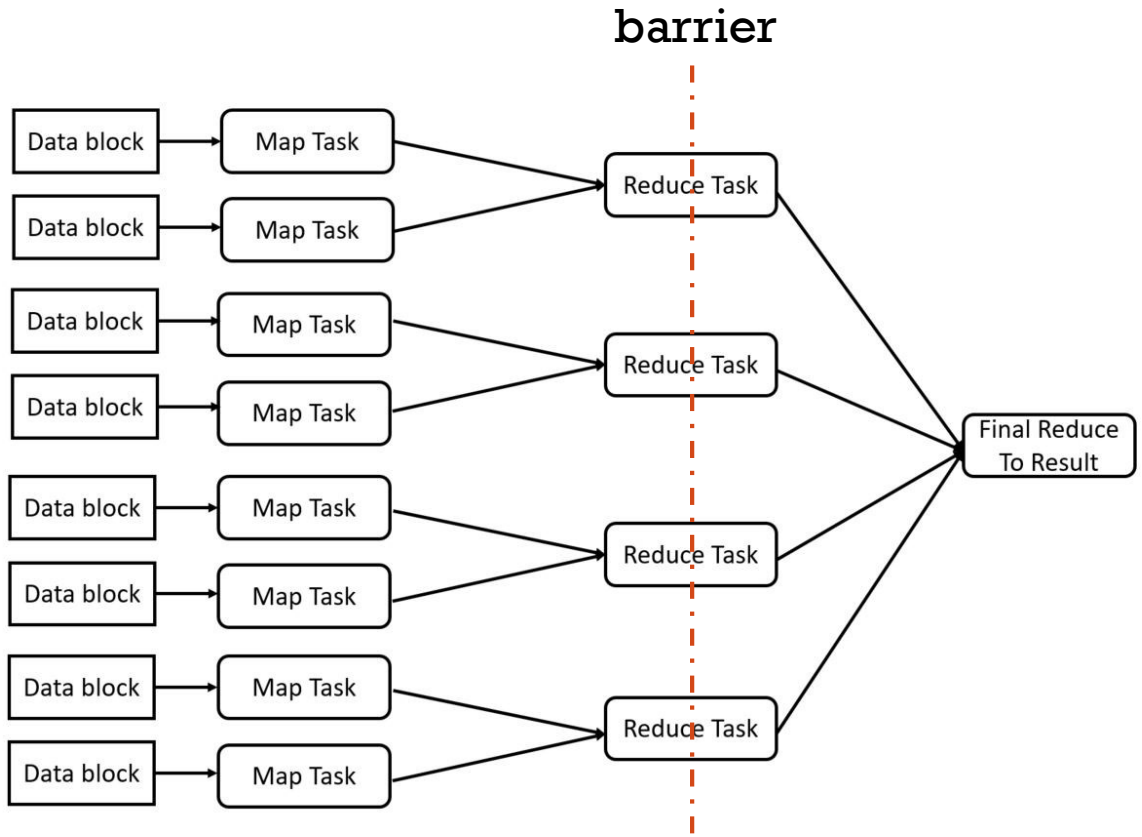
- Task parallel model is great for solving problems that involve doing many independent computations



Each worker repeatedly pulls a sample from a queue, processes the data,
and stores the result in a shared table

BSP – MAPREDUCE

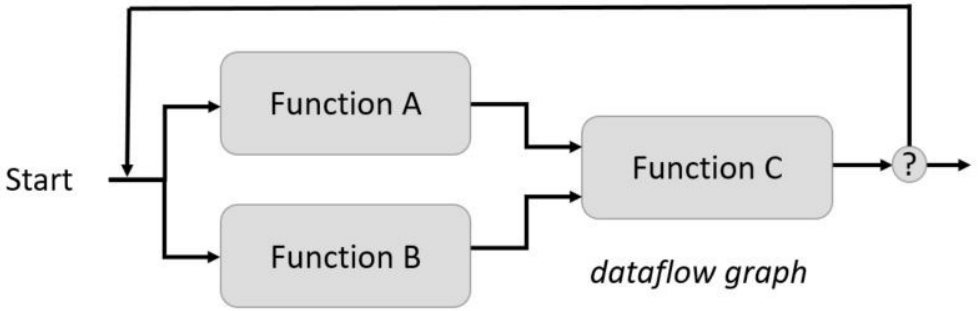
- Worker tasks periodically synchronize and exchange data with each other
- *Barrier* = the point of synchronization
- MapReduce – a special case of BSP
  - Map Task = an operation applied to blocks of data in parallel
  - Reduce Task- when maps are "done" reduce the results to a single result
- MapReduce → open source: Hadoop

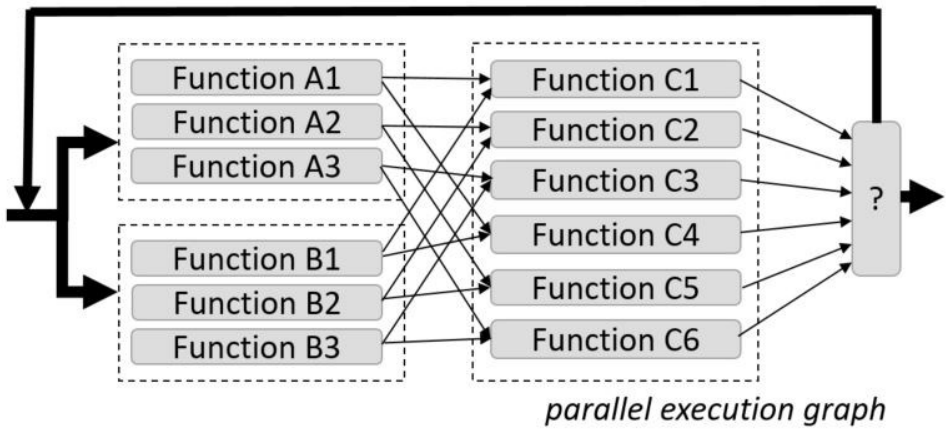


GRAPH PARALLEL

- Each function node represents a parallel invocation of the function on the distributed data structure
- 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

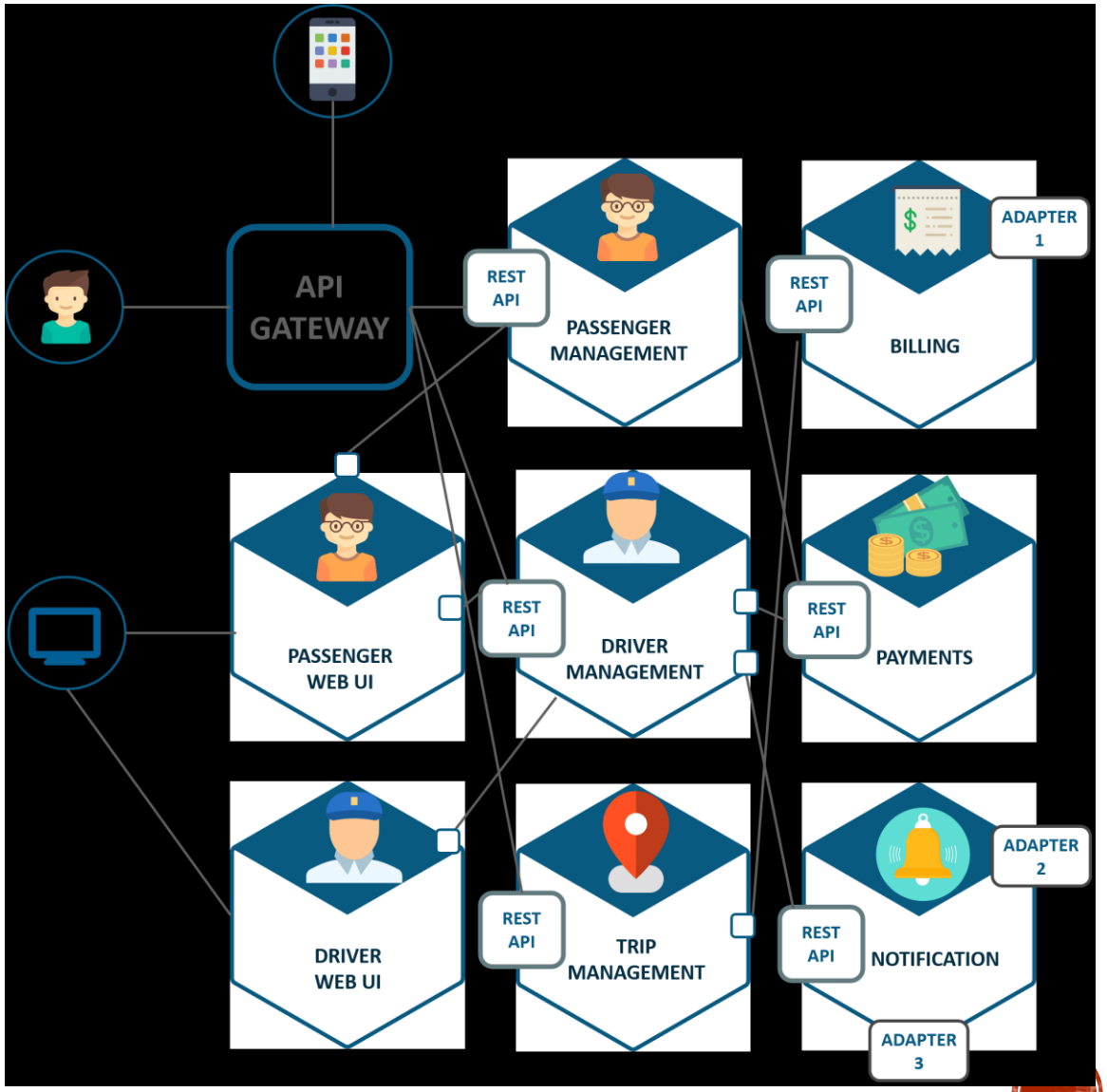
- Data Analytic: Spark, Spark Streaming, Apache Flink, Storm, Google DataFlow
- Machine Learning: Google TensorFlow, MS Cognitive Toolkit





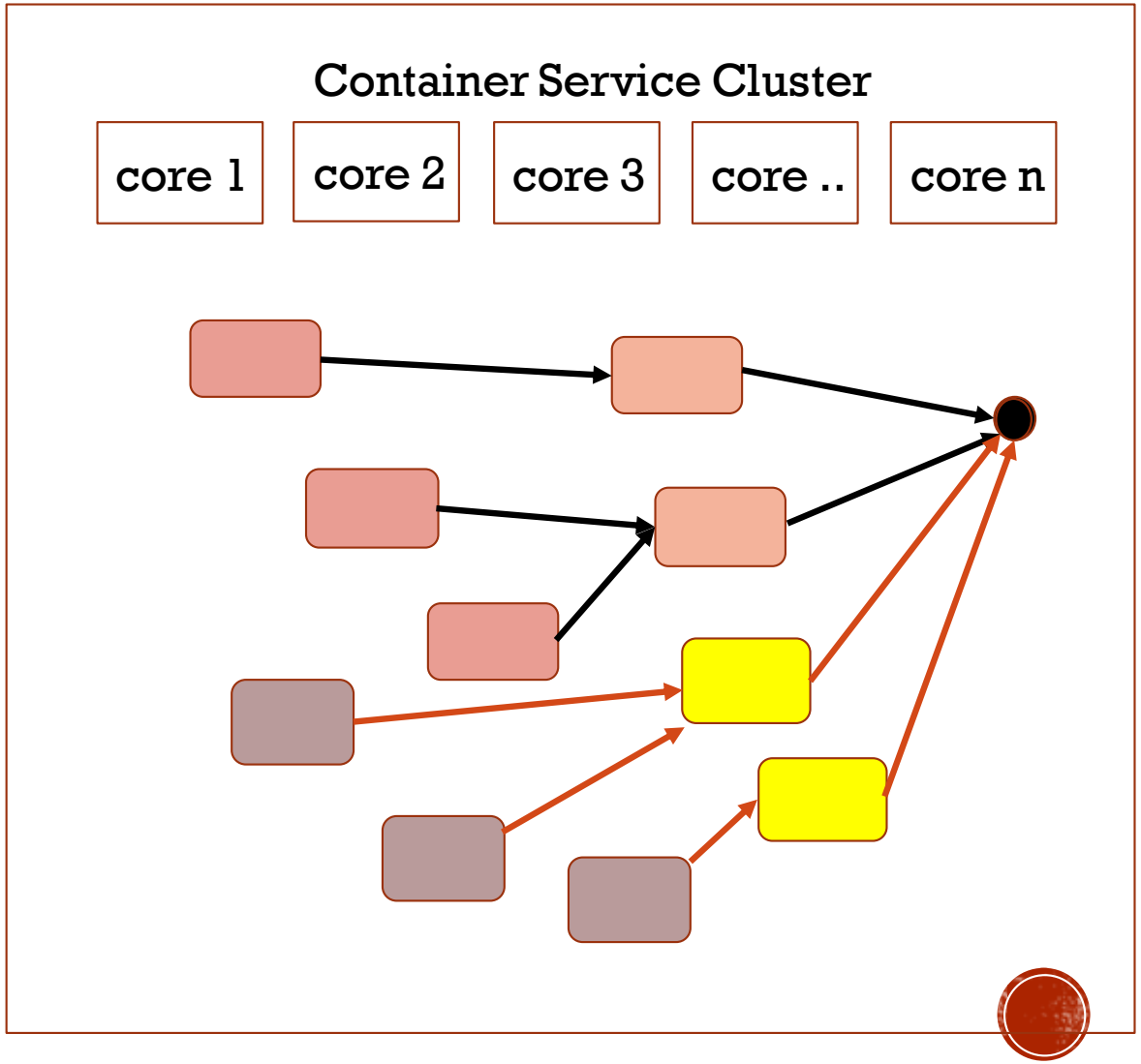
MICROSERVICES

- 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.



MICROSERVICES IN THE CLOUD

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



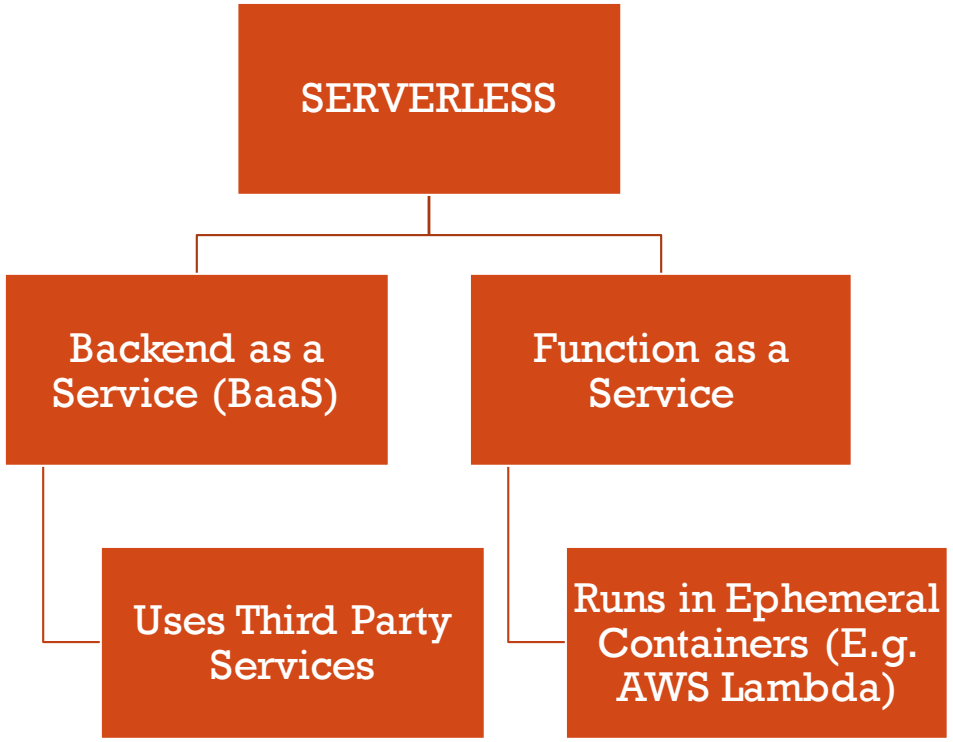
WHAT IS REST API?

- Interface for **distributed hypermedia systems**, created by Roy Fielding in 2000
- Guiding Principles of REST
  - Client – Server
  - Stateless
  - Cacheable
  - Uniform Interface
  - Layered System
  - Code on demand (optional)
- The key abstraction of information in REST is a **resource**
  - Identified by a resource identifier, ie URI
- Resources can be retrieved or transformed to another state by a set of methods
  - GET/PUT/POST/DELETE/PATCH
- The clients and servers exchange representations of resources by using a standardized interface and protocol typically HTTP

Good tutorial - https://www.smashingmagazine.com/2018/01/understanding-using-rest-api/

SERVERLESS

- New paradigm for service delivery
  - User provides a simple function to execute, against under certain conditions
- Very lightweight process, similar to daemon waiting for event to occur
- Managed by cloud infrastructure
  - No server management
  - Pay only while your code runs
  - Scale automatically
  - Highly available and fault tolerant
- Examples: AWS Lambda, Google Cloud Function, Azure Function



A SIMPLE SERVERLESS EXAMPLE - GCP

```
/**
 * HTTP Cloud Function.
 * This function is exported by index.js, and is executed when
 * you make an HTTP request to the deployed function's endpoint.
 *
 * @param {Object} req Cloud Function request context.
 *                     More info: https://expressjs.com/en/api.html#req
 * @param {Object} res Cloud Function response context.
 *                     More info: https://expressjs.com/en/api.html#res
 */
exports.helloGET = (req, res) => {
  res.send('Hello World!');
};
```

Have a simple function that return "Hello World"

```
gcloud functions deploy helloGET --runtime nodejs6 --trigger-http
```

Deploy the function with HTTP trigger

```
gcloud functions describe helloGET
```

```
https://GCP_REGION-PROJECT_ID.cloudfunctions.net/helloGET
```

Visit this URL in your browser. You should see a Hello World!

SERVERLESS

- Pros
  - No need to write or manage backend code
  - Achieve Event-based programming without the complexity of building and maintaining the infrastructure.

- Cons
  - Vendor Lock-in
  - Architectural Complexity
    - AWS Lambda limits the number of concurrent executions you can be running of all your lambdas.
  - Startup Latency in FaaS
    - It takes time to initialize an instance of a function before each event.