Lecture 8:

- Services, Serverless
- Cloud systems building-blocks
- Intro to Borg

Announcements:

- R-1 due Mon
- HW-1 out

Additional Concepts Related to Cloud Computing

**Service**: functionality, used by clients, well-defined interface

- e.g. *service-oriented architecture* divides system into distinct services

"**As a Service**": a service offered by a third party in the cloud

- don’t have to purchase, manage, use hardware/software from on-premise
- instead, pay for service – via subscription or on demand (based on use)

**Various types of As a Service**:

- **Infrastructure (IaaS)**: Physical compute, storage, networking, virtualization
- **Platform (PaaS)**: For developing apps (DBMS, Web Servers, ML tech.)
- **Software (SaaS)**: Entire applications for customers (lots of examples)
**Additional Concepts Related to Cloud Computing**

**Fully Managed:** *Automated provisioning, user server management*

- choose number of (virtual) machines, resources, etc.
- machines automatically provisioned, set up, maintained, etc.
- pay for machine and resource usage (e.g., Compute Engine)
- similar to IaaS

**Serverless:** *Automated provisioning and server management*

- fully managed and don’t have to manage servers or scale them
- pay for what you use (not on provisioned resources)
- many examples (Cloud Run, BigQuery query processing, etc.)
- similar to PaaS

---

**Cloud Building Blocks**

Google internal & cloud services run on top of 3 main services:

- **Borg:** Cluster-scale job scheduling
- **Colossus:** Distributed file storage (next-gen GFS)
- **Spanner:** Distributed transactional database system

Similar patterns in these and systems built on them:

- **Scalability:** many machines, data partitioning (sharding), parallel exec.
- **Replication & Recovery:** for fault tolerance, high availability
- **Coordination:** centralized components to “manage” parallel “workers”
- **Consistency:** concurrency control to deal with replica differences
- **Placement:** manage communication cost and data movement
Borg designed to support two types of jobs:

1. **Long Running Services**: that never shut down
   - involve short-lived latency-sensitive requests
   - end-user products (e.g., GMail), internal tools (e.g., BigTable)

2. **Batch Jobs**: take a few seconds to a few days to run

Jobs consist of one or more tasks: where each task runs the same program

- tasks run in containers (isolation, not VMs)
- jobs can be sequenced (one starts after another finishes)
- each job runs in a single cell – one cell per cluster
- each cell can have a mix of job types

Jobs and tasks can have resource constraints:

- processor architecture, OS version, external IP address, etc. (jobs)
- CPU cores, RAM, disk space, etc. (tasks)

Jobs assigned priorities:

- in paper, only production (high) and non-production (low)
- actually 100s of priorities by “tier”: free, best-effort, mid, production, etc.
Borg Architecture Basics

**Borg’s main functions:**
- figure out where (machine) in a cell to run tasks
- allocate the resources for tasks
- install programs and dependencies for running tasks
- monitor task health
- restart tasks on failure

**BorgMaster:** centralized controller
- each cell has a replicated borgmaster (5 replicas)
- handles client requests (e.g., create/lookup job)
- each replica stores state information of the cell (in distributed database)
- cell has a leader borgmaster that handles all operations
- if leader fails, another leader is chosen (elected)
- when a job is submitted, records info in DB and adds to pending queue
Summary – Things to Know

- Services, “as a Service”, IaaS, Paas, SaaS
- Managed, Serverless
- 3 main GCP building blocks (services)
- Borg’s purpose, basic operation