Cloud Computing – Part 1

Comprehensive Review

DSC 208R – Parallel Data Processing and the Cloud

Contents

1 Motivation

Cloud computing rents compute, storage, memory, and networking resources from remote servers. Key advantages over on-premise clusters are

- **Manageability**: hardware is the provider's problem.
- **Pay as you go**: fine-grained pricing from seconds to years.
- **Elasticity**: capacity can grow or shrink with workload demand.

These points frame the appeal of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).:contentReference[oaicite:0]index=0

2 Layered Model of Cloud Services

Table 1: Typical cloud layers and AWS examples.

Layer	User control	Example AWS services
IaaS	OS, runtime, data	EC2, EBS, VPC, S3
PaaS	Code, data	Aurora, Redshift, EMR, DynamoDB
SaaS	Only usage	SageMaker, Textract, Chime

From a renter's view, control decreases as one moves from IaaS to SaaS, while management burden also decreases.:contentReference[oaicite:1]index=1

3 Evolution of Cloud Infrastructure

1. **Cloud 1.0**: networked servers; users rent whole machines.
2. **Cloud 2.0**: virtualized servers and multi-tenancy; users rent resource *capacity*, enabling load balancing and better elasticity.
3. **Cloud 3.0** (ongoing): serverless and fully disaggregated resources connected by fast networks.

Each generation gives the provider more flexibility to pack workloads efficiently.:contentReference[oaicite:2]index=2

4 Parallelism Paradigms Revisited

Traditional parallel styles carry into the cloud:

- **Shared-nothing**: independent workers with local data (dominant in RDBMS, Hadoop, Spark).
- **Shared-memory**: nodes share a large memory space.
- **Shared-disk**: nodes share a storage layer (now common via S3 plus local EBS).

Modern data-center networks hit 100 Gbps or more, making hybrids common: compute on EC2, data on S3 (shared-disk), and local shuffles in EBS (shared-nothing).:contentReference[oaicite:3]index=3

5 Example IaaS Architecture on AWS

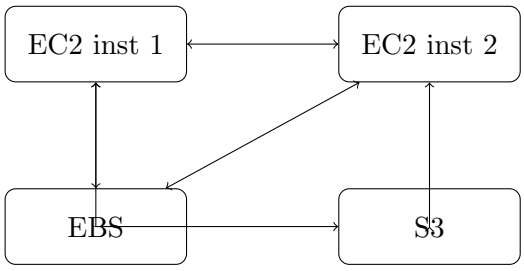

Figure 1: Shared-nothing compute with shared-disk storage in AWS.

6 New Renting Paradigms

6.1 Spot vs On-Demand Instances

- **On-Demand**: fixed hourly price, no interruption risk.
- **Spot**: bid for unused capacity; price fluctuates, instance can be reclaimed with short notice.

Spot can cut costs by up to an order of magnitude but suits only fault-tolerant or checkpointed jobs.:contentReference[oaicite:4]index=4

6.2 Serverless Function as a Service

Users upload a function and pay for milliseconds of execution time; the provider scales instances automatically. This model embodies Cloud 3.0's disaggregated compute vision.

7 Pros and Cons Summary

- **Pros**
  - No hardware maintenance.
  - Fine-grained cost aligned with usage.
  - Rapid elasticity to meet workload spikes.
- **Cons**
  - API and licensing complexity; need CloudOps skills.
  - Long-term spend can exceed on-premise clusters.
  - Vendor lock-in, outage, and security concerns.

8 Future Directions

- Fully disaggregated resources with sub-second elasticity.
- Cross-vendor orchestration to mitigate lock-in.
- Better cost observability and automated budget guards.

Conclusion

Cloud Computing Part 1 highlights why elastic, pay-as-you-go resources have reshaped data analytics. Understanding service layers, evolving infrastructure, and new renting paradigms equips practitioners to balance cost, performance, and operational risk in modern data pipelines.:contentReference[oaicite:5]index=5