DSC 102
Systems for Scalable Analytics

Arun Kumar

Topic 2: Basics of Cloud Computing
Cloud Computing

- Compute, storage, memory, networking, etc. are virtualized and exist on remote servers; rented by application users.
- Main pros of cloud vs on-premise clusters:
  - **Manageability**: Managing hardware is not user’s problem.
  - **Pay-as-you-go**: Fine-grained pricing economics based on actual usage (granularity: seconds to years!)
  - **Elasticity**: Can dynamically add or reduce capacity based on actual workload’s demand.
- Infrastructure-as-a-Service (IaaS); Platform-as-a-Service (PaaS); Software-as-a-Service (SaaS)
Cloud Computing

Resources Managed at each Layer:

- **Application Layer**
  - Applications / Software
  - Software as a Service (SaaS)
- **Platform Layer**
  - Software Frameworks (Java/Python/.NET)
  - Platform as a Service (PaaS)
  - Storage (Database / File)
- **Infrastructure Layer**
  - Virtual Machines
  - Infrastructure as a Service (IaaS)
  - CPU, Memory, Disk
- **Hardware Layer**
  - Network / System Administrators
Examples of AWS Cloud Services

❖ IaaS:
  ❖ **Compute**: EC2, ECS, Fargate, Lambda
  ❖ **Storage**: S3, EBS, EFS, Glacier
  ❖ **Networking**: CloudFront, VPC

❖ PaaS:
  ❖ **Database/Analytics Systems**: Aurora, Redshift, Neptune, ElastiCache, DynamoDB, Timestream, EMR, Athena
  ❖ **Blockchain**: QLDB; **IoT**: Greengrass

❖ SaaS:
  ❖ **ML/AI**: SageMaker, Elastic Inference, Lex, Polly, Translate, Transcribe, Textract, Rekognition, Ground Truth
  ❖ **Business Apps**: Chime, WorkDocs, WorkMail
Evolution of Cloud Infrastructure

- **Data Center**: Physical space from which a cloud is operated
- **3 generations of data centers/clouds:**
  - **Cloud 1.0 (Past)**: Networked servers; user rents servers (time-sliced access) needed for data/software
  - **Cloud 2.0 (Current)**: “Virtualization” of networked servers; user rents amount of resource capacity; cloud provider has a lot more flexibility on provisioning (multi-tenancy, load balancing, more elasticity, etc.)
  - **Cloud 3.0 (Ongoing Research)**: “Serverless” and disaggregated resources all connected to fast networks
3 Paradigms of Multi-Node Parallelism

- **Shared-Nothing Parallelism**
  - Independent Workers
  - Interconnect
  - Contention

- **Shared-Disk Parallelism**
  - Interconnect

- **Shared-Memory Parallelism**
  - Interconnect

Most parallel RDBMSs (Teradata, Greenplum, Redshift), Hadoop, and Spark use shared-nothing parallelism.
Modern networks in data centers have become much faster: 100GbE to even TbE!

- **Decoupling** of compute+memory from storage is common in cloud
  - *Hybrids* of shared-disk parallelism + shared-nothing parallelism
  - E.g, store datasets on S3 and read as needed to local EBS
Example: AWS Services for PA1

Elastic Compute Cloud (EC2)
Elastic Block Storage (EBS)

Machine Instance 1
Machine Instance 2

Simple Storage Service (S3)

AWS-internal Interconnect

Internet

You

AWS Data Center(s)
Example: AWS Database/Analytics

Example: AWS services for ML app.

https://docs.aws.amazon.com/sagemaker/latest/dg/how-it-works-training.html
New Cloud Renting Paradigms

❖ Cloud 2.0’s flexibility enables radically different paradigms
❖ AWS example (Azure and GCP have similar gradations):

AWS EC2 Consumption Models

<table>
<thead>
<tr>
<th>On-Demand</th>
<th>Reserved</th>
<th>Spot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pay for compute capacity by the second or hour with no long-term commitments</td>
<td>Significant discount compared to On-Demand instance pricing</td>
<td>Spare EC2 capacity for up to 90% off the On-Demand price.</td>
</tr>
<tr>
<td>For spiky workloads or to define needs initially</td>
<td>Steady state applications or predictable usage, databases</td>
<td>For fault tolerant, instance flexible or time-insensitive workloads</td>
</tr>
</tbody>
</table>

https://www.slideshare.net/AWSUsersGroupBengalu/amazon-ec2-spot-instances
## More on Spot vs On-Demand

<table>
<thead>
<tr>
<th>Spot Instances</th>
<th>On-Demand Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Launch time</strong></td>
<td>Can only be launched immediately if you make a manual launch request and capacity is available.</td>
</tr>
<tr>
<td>Can only be launched immediately if the Spot Request is active and capacity is available.</td>
<td></td>
</tr>
<tr>
<td><strong>Available capacity</strong></td>
<td>If capacity is not available when you make a launch request, you get an insufficient capacity error (ICE).</td>
</tr>
<tr>
<td>If capacity is not available, the Spot Request continues to automatically make the launch request until capacity becomes available.</td>
<td></td>
</tr>
<tr>
<td><strong>Hourly price</strong></td>
<td>The hourly price for On-Demand Instances is static.</td>
</tr>
<tr>
<td>The hourly price for Spot Instances varies based on demand.</td>
<td></td>
</tr>
<tr>
<td><strong>Rebalance recommendation</strong></td>
<td>You determine when an On-Demand Instance is interrupted (stopped, hibernated, or terminated).</td>
</tr>
<tr>
<td>The signal that Amazon EC2 emits for a running Spot Instance when the instance is at an elevated risk of interruption.</td>
<td></td>
</tr>
<tr>
<td><strong>Instance interruption</strong></td>
<td></td>
</tr>
<tr>
<td>You can stop and start an Amazon EBS-backed Spot Instance. In addition, the Amazon EC2 Spot service can interrupt an individual Spot Instance if capacity is no longer available, the Spot price exceeds your maximum price, or demand for Spot Instances increases.</td>
<td>You determine when an On-Demand Instance is interrupted (stopped, hibernated, or terminated).</td>
</tr>
</tbody>
</table>
New Cloud Renting Paradigms

Such bundling means some applications might under-utilize some resources!

❖ **Serverless** paradigm gaining traction for some applications, e.g., online ML prediction serving on websites
❖ User gives a program (function) to run and specifies CPU and DRAM needed
❖ Cloud provider abstracts away all resource provisioning entirely
❖ Higher resource efficiency; much cheaper, often by 10x vs Spot instances
❖ Aka *Function-as-a-Service* (FaaS)
Car Analogy for Serverless Cloud

Own a car
(Bare metal servers)

Rent a car
(VPS)

City car-sharing
(Serverless)

Cars are parked 95% of the time (loige.link/car-parked-95)

How much do you use the car?

Example: Serverless RDBMS on AWS

- Remote read of data from S3
- Schema-on-read
- Many data formats
- Simple interactive queries

https://www.xenonstack.com/blog/amazon-athena-quick sight/
Example: Serverless ML app. on AWS

Disaggregation: Glimpse into the Future?

- Logical next step in serverless direction: full **resource disaggregation**! That is, compute, memory, storage, etc. are all network-attached and elastically added/removed.

Ongoing Research: Fulfill this promise with low latency!
Example: AWS services for IoT app.

OMG, is all this complexity worth it?!

- Depends on user’s/application’s Pareto tradeoffs! :)
- **On-premise** cluster are still common in large enterprises, healthcare, and academia; “hybrid clouds” too
- Recall main pros of cloud: manageability, cost, and elasticity
- Some “cons” of cloud (vs on-premise):
  - **Complexity** of composing cloud APIs and licenses; data scientists must keep relearning; “CloudOps” teams
  - **Cost** over time can crossover and make it costlier!
  - “Lock-in” by cloud vendor
  - **Privacy, security, and governance** concerns
  - **Internet disruption** or **unplanned downtime**, e.g., AWS outage in 2015 made Netflix, Tinder, etc. unavailable! :)
OMG, is all this complexity worth it?!

Immediate Release

DOD Announces Enterprise General Purpose Cloud Contract Award

Oct. 25, 2019

Over the last two years the Department of Defense has awarded more than $11 billion across 10 separate cloud contracts. As we continue to execute the DOD Cloud Strategy, additional contracts are planned for both cloud services and complementary migration and integration solutions necessary to achieve effective cloud adoption.
The State of the Cloud Survey

Public Cloud Adoption

% of All Respondents

- AWS: 61% Currently use, 16% Experimenting, 7% Plan to use
- Azure: 52% Currently use, 16% Experimenting, 9% Plan to use
- Google Cloud: 19% Currently use, 22% Experimenting, 14% Plan to use
- VMware Cloud on AWS: 12% Currently use, 16% Experimenting, 13% Plan to use
- IBM Cloud: 12% Currently use, 9% Experimenting, 6% Plan to use
- Oracle Cloud: 11% Currently use, 8% Experimenting, 7% Plan to use
- Alibaba Cloud: 3% Currently use, 6% Experimenting, 8% Plan to use

Source: RightScale 2019 State of the Cloud Report from Flexera

The State of the Cloud Survey

Public Cloud Services Used – All Respondents

<table>
<thead>
<tr>
<th>Service</th>
<th>2019</th>
<th>2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBaaS (Relational)</td>
<td>53%</td>
<td></td>
</tr>
<tr>
<td>Push notifications</td>
<td>43%</td>
<td>41%</td>
</tr>
<tr>
<td>Data warehouse</td>
<td>40%</td>
<td></td>
</tr>
<tr>
<td>Queueing</td>
<td>39%</td>
<td>35%</td>
</tr>
<tr>
<td>DBaaS (NoSQL)</td>
<td>39%</td>
<td>33%</td>
</tr>
<tr>
<td>Container-as-a-service</td>
<td>37%</td>
<td>26%</td>
</tr>
<tr>
<td>Mobile services</td>
<td>37%</td>
<td>31%</td>
</tr>
<tr>
<td>Severless</td>
<td>36%</td>
<td>29%</td>
</tr>
<tr>
<td>Caching</td>
<td>36%</td>
<td>32%</td>
</tr>
<tr>
<td>Batch processing</td>
<td>36%</td>
<td>24%</td>
</tr>
<tr>
<td>Search</td>
<td>36%</td>
<td>26%</td>
</tr>
<tr>
<td>Hadoop</td>
<td>30%</td>
<td>29%</td>
</tr>
<tr>
<td>Stream processing</td>
<td>20%</td>
<td>24%</td>
</tr>
<tr>
<td>Machine learning</td>
<td>26%</td>
<td>18%</td>
</tr>
<tr>
<td>DRaaS</td>
<td>26%</td>
<td>21%</td>
</tr>
<tr>
<td>IoT</td>
<td>21%</td>
<td>15%</td>
</tr>
</tbody>
</table>

Source: RightScale 2019 State of the Cloud Report from Flexera

The State of the Cloud Survey

### Top Cloud Initiatives in 2019

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimize existing use of cloud (cost savings)</td>
<td>64%</td>
</tr>
<tr>
<td>Move more workloads to cloud</td>
<td>58%</td>
</tr>
<tr>
<td>Expand use of containers</td>
<td>39%</td>
</tr>
<tr>
<td>Implement a cloud-first strategy</td>
<td>39%</td>
</tr>
<tr>
<td>Automated policies for governance</td>
<td>35%</td>
</tr>
<tr>
<td>Better financial reporting on cloud costs</td>
<td>35%</td>
</tr>
<tr>
<td>Expand public clouds we use</td>
<td>33%</td>
</tr>
<tr>
<td>Implement CI/CD in the cloud</td>
<td>33%</td>
</tr>
<tr>
<td>Move on-prem software to SaaS</td>
<td>29%</td>
</tr>
<tr>
<td>Manage software licenses in the cloud</td>
<td>24%</td>
</tr>
<tr>
<td>Enable IT to broker cloud services</td>
<td>20%</td>
</tr>
</tbody>
</table>

*Source: RightScale 2019 State of the Cloud Report from Flexera*

The State of the Cloud Survey

Policies to Optimize Cloud Costs

<table>
<thead>
<tr>
<th>Policy</th>
<th>Automated policies</th>
<th>Manual policies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shutdown workloads after hours</td>
<td>35%</td>
<td>36%</td>
</tr>
<tr>
<td>Rightsize instances</td>
<td>31%</td>
<td>49%</td>
</tr>
<tr>
<td>Required tags</td>
<td>32%</td>
<td>38%</td>
</tr>
<tr>
<td>Specify expiration dates</td>
<td>29%</td>
<td>38%</td>
</tr>
<tr>
<td>Eliminate inactive storage</td>
<td>24%</td>
<td>49%</td>
</tr>
<tr>
<td>Software license compliance</td>
<td>22%</td>
<td>53%</td>
</tr>
<tr>
<td>Allowed instance sizes/types</td>
<td>21%</td>
<td>50%</td>
</tr>
<tr>
<td>Underutilized discounts</td>
<td>21%</td>
<td>44%</td>
</tr>
<tr>
<td>Use lowest-cost cloud</td>
<td>15%</td>
<td>42%</td>
</tr>
<tr>
<td>Use lowest-cost regions</td>
<td>15%</td>
<td>47%</td>
</tr>
</tbody>
</table>

Source: RightScale 2019 State of the Cloud Report from Flexera