Janus
User-Defined
Cloud-backed
Storage

Ricardo Mendes, Alysson Bessani, Tiago Oliveira
Introduction

• Tons of data are generated everyday that need to be stored

• Users and companies are moving their data to the clouds

• Cloud storage services
  – store their data in a single cloud provider (e.g. Amazon S3) which can fail
  – lack the capabilities to meet some user requirements (Security & Dependability, Cost, Data Locality, etc.)
Introduction

• Different **types of data** have **different requirements**
  – Durability, access patterns, location, etc.

• Users are willing to spend different amounts of resources to store them
Introduction

• There are many **cloud storage providers** with **different characteristics**
  – Price, latency, location, security standards, etc.
## Clouds’ Characteristics

<table>
<thead>
<tr>
<th>Cloud</th>
<th>Cost</th>
<th>Multi-Account Sharability</th>
<th>SLA (#9s)</th>
<th>Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Storage (¢$/GB/mon)</td>
<td>in (¢$)</td>
<td>out (¢$)</td>
<td></td>
</tr>
<tr>
<td>BlackBlaze</td>
<td>0,5</td>
<td>0</td>
<td>0,5</td>
<td>4</td>
</tr>
<tr>
<td>RunAbove</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Seagate EVault</td>
<td>1,5</td>
<td>0</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Google Cloud</td>
<td>2</td>
<td>0</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>Amazon S3</td>
<td>2,4</td>
<td>0</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>Microsoft Azure</td>
<td>2,4</td>
<td>0</td>
<td>8,7</td>
<td>3</td>
</tr>
<tr>
<td>DreamHost</td>
<td>2,5</td>
<td>0</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>LunaCloud</td>
<td>3,8</td>
<td>0</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>IBM – SoftLayer</td>
<td>4</td>
<td>0</td>
<td>10</td>
<td>∞</td>
</tr>
<tr>
<td>Joyent Manta</td>
<td>4,3</td>
<td>0</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>4,7</td>
<td>0</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>HP Cloud</td>
<td>9</td>
<td>0</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>Rackspace</td>
<td>10</td>
<td>0</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>CenturyLink</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>
Janus Storage Service

• Provides Secure multi-cloud storage

Tolerates Failure Events

• Allows users to define their storage requirements
• Generates requirement compliant storage volumes

Choose the best Clouds for the User Requirements
Janus Platform Architecture

1. Volume specs.
2. Janus
3. driver
4. Cloud services: Windows Azure, AT&T, Amazon S3, Google, Rackspace
# Volume Specifications

## Available Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Workload</th>
<th>Sharability</th>
<th>Computable</th>
<th>Data Location</th>
<th>Consistency Guarantees</th>
<th>Provider Fault Tolerance</th>
<th>Security Standards Compliance</th>
<th>Budget</th>
<th>SLA</th>
<th>Available Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>● Num writes</td>
<td>● No</td>
<td>● Single-Account</td>
<td>● NOT IN</td>
<td>● Eventual consistency</td>
<td>● CFT</td>
<td>● COMPLY</td>
<td>● Max price</td>
<td>● Latency</td>
<td>● Disk space</td>
</tr>
<tr>
<td></td>
<td>● Num Reads</td>
<td></td>
<td>● Multi-account</td>
<td>● ONLY IN</td>
<td>● Strong consistency</td>
<td>● BFT</td>
<td>● NOT COMPLY</td>
<td>● Availability SLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Num Files</td>
<td></td>
<td></td>
<td>● PREFER IN</td>
<td></td>
<td></td>
<td>● PREF COMPLY</td>
<td>● Durability SLA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Average File Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Virtual Disk Driver

• **Dependable & Secure** Multi-cloud storage
  – Ciphers Data
  – Tolerates cloud failures
• **Storage Efficient**
  – Caching, Data Compression, Erasure Codes
• **Exports various interfaces:**
  – Filesystem, NFS, CIFS, FTP and OpenStack SWIFT

Original file $\rightarrow$ Compressed file $\rightarrow$ Encrypted file

Coded blocks – all the blocks are different

- Amazon S3
- Windows Azure
- Google
Janus Server

- **Front-end** of the platform
- **Provides the user-interface** in which users specify their storage requirements
- **Generates the best volume configurations** for the given requirements
Billing Manager

• Automatizes the **creation and management** of the **cloud providers accounts**

• Maintains the **billing information** of each client
  – e.g. read/storage costs for each cloud used

• Credentials for accessing clouds are sent to the client together with the disk driver installation file
Cloud info Collector

- Responsible for **periodically obtaining information** about all cloud services
  - 13 providers, 24 services, 68 geographical locations
  - e.g. costs, SLAs, security standards

- This information is used to feed the *Requirements Solver* module

- Reduces the management effort the system needs
Requirements Solver

• **Finds the best storage configurations** to be used by the virtual disk driver
  – e.g. cloud services, locations, storage techniques, etc.

• Implemented in Prolog

• Uses the storage requirements given by the client the clouds’ info provided by *Cloud info Collector module*
Requirements Solver

- Clouds’ info are in the **knowledge base** in the form of a set of Prolog **facts** and **relations**
  - service(s3).
  - provider(s3,aws).

- Knowledge base also define several static **rules** to relate the clouds info
  - same_provider(X,Y) :- provider(X,Z), provider(Y,Z).
Query Solving Strategy

• Find the most adequate cloud services to use is hard!
  – Search space is huge!
  – e.g. 68 service locations, BFT (3f+1), f=3
    • $C(68,10) \approx 300 \text{ Billion } \rightarrow 1 \mu s \approx 3 \text{ days}$

• Minimize this by separate this task in three steps:
  – 1\textsuperscript{st}: \textbf{Filter} the cloud services according to the requirements
  – 2\textsuperscript{nd}: \textbf{Sort} the clouds according to a linear function $f(x)$ which quantifies the interest of cloud $x$ to the configuration
  – 3\textsuperscript{rd}: \textbf{Combine} the services to obtain the $m$ best configurations
Query Solving Strategy

- \( n=2f+1, \ f = 1 \rightarrow n=3 \) clouds
- Only EU
- \( m = 4 \)

Query Solving Strategy

- \( n=2f+1, \ f = 1 \rightarrow n=3 \) clouds
- **Only EU**
- \( m = 4 \)

\[
\begin{array}{lllllll}
B \ (EU) & C \ (US) & F \ (EU) & D \ (EU) & E \ (EU) & A \ (EU) & G \ (US) & H \ (US) \\
\end{array}
\]

1st: Filter

\[
\begin{array}{llll}
B \ (EU) & F \ (EU) & D \ (EU) & E \ (EU) & A \ (EU) \\
\end{array}
\]
Query Solving Strategy

- \( n = 2f + 1, \ f = 1 \rightarrow n = 3 \text{ clouds} \)
- Only EU
- \( m = 4 \)

\[
\begin{array}{cccccc}
B & F & D & E & A \\
\end{array}
\]

2nd: Sort

\[
\begin{array}{cccccc}
A & B & D & E & F \\
\end{array}
\]
Query Solving Strategy

- $n = 2f + 1$, $f = 1 \rightarrow n = 3$ clouds
- Only EU
- $m = 4$

A B D E F

Mark an **index** (starting at $n$)

3rd: Combine
Query Solving Strategy

- $n=2f+1$, $f = 1 \rightarrow n=3$ **clouds**
- Only EU
- $m = 4$

- $\langle A, B \rangle$

Generate all combinations of $n-1$ elements before the index

3rd: Combine
Query Solving Strategy

- \( n = 2f + 1, \ f = 1 \rightarrow n = 3 \text{ clouds} \)
- Only EU
- \( m = 4 \)
- \( \langle A, B, D \rangle \)

3rd: Combine

Concatenate value on index
Query Solving Strategy

- \( n=2f+1, f = 1 \rightarrow n=3 \text{ clouds} \)
- Only EU
- \( m = 4 \)

- \(<A, B, D>\)

Increase index if the number of combinations is smaller than \( m \)

3rd: Combine
Query Solving Strategy

- $n = 2f + 1$, $f = 1 \rightarrow n = 3\text{ clouds}$
- Only EU
- $m = 4$
- $<A,B,D>$
Query Solving Strategy

- $n = 2f + 1$, $f = 1 \rightarrow n = 3$ clouds
- Only EU
- $m = 4$

- $<A, B, D>$
- $<A, B, E>$
- $<A, D, E>$
- $<B, D, E>$
Query Solving Strategy

- $n=2f+1$, $f = 1 \rightarrow n=3$ clouds
- Only EU
- $m = 4$

- $\langle A, B, D \rangle$
- $\langle A, B, E \rangle$
- $\langle A, D, E \rangle$
- $\langle B, D, E \rangle$

We have more than $m$ solutions
Challenges

• What is the best way to define the function $f(x)$ used to sort the cloud services in the solver?

• Volume reconfiguration
Thanks!

Questions?

LaSIGE
Large-Scale Informatics Systems Laboratory

Ciências ULisboa

Navigators

SUPER CLOUD