UNIVERSITY OF LISBON
FACULTY OF SCIENCES

Janus

User-Defined Cloud-backed Storage

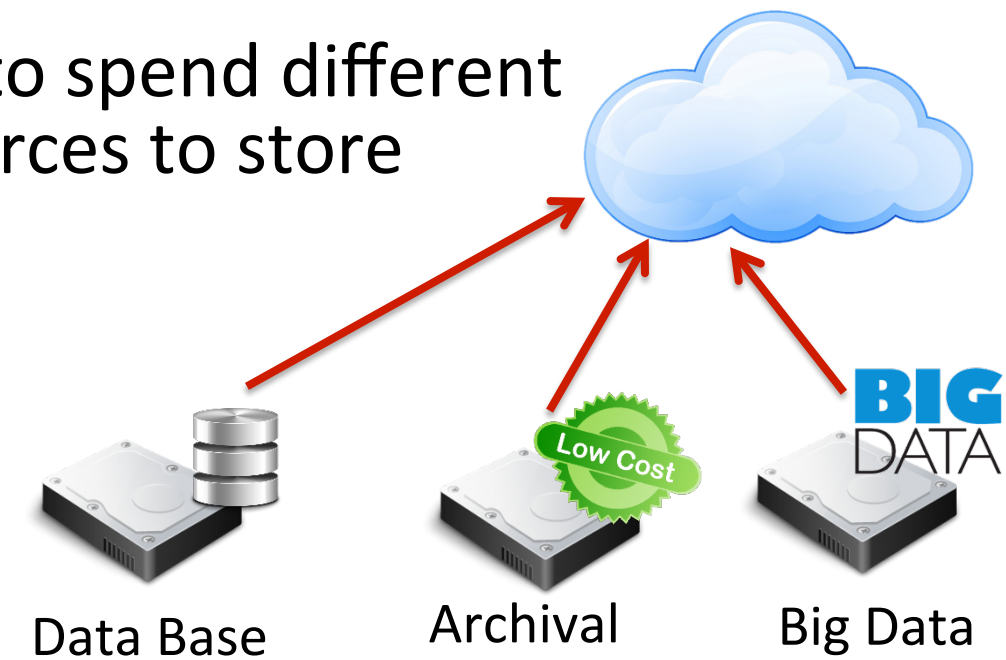
Ricardo Mendes, Alysson Bessani, Tiago Oliveira

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

Cloud	Cost			Multi-Account Sharability	SLA (#9s)	Locations
	Storage (¢$/GB/mon)	in (¢$)	out (¢$)			
BlackBlaze	0,5	0	0,5			
RunAbove	1	0	1		4	2 - CAN, EU
Seagate EVault	1,5	0	7		3	1 - US
Google Cloud	2	0	12	ACL	3	3 - AS, EU, US
Amazon S3	2,4	0	9	ACL	4	8 - US, EU, AS, AUS, BR
Microsoft Azure	2,4	0	8,7	TempUrl	3	12 - US, EU, AS, BR
DreamHost	2,5	0	5		3	2 - US
LunaCloud	3,8	0	10	none	4	1 - EU
IBM – SoftLayer	4	0	10		∞	15 - EU, US, CAN, AUS, AS
Joyent Manta	4,3	0	12			4 - EU, US
AT&T	4,7	0	10		4	4 - US, EU
HP Cloud	9	0	12	ACL, TempUrl		2 - US
Rackspace	10	0	12	ACL, TempUrl	3	6 - US, EU, AS, AUS
CenturyLink	15	5	5		4	1 - CAN

Janus Storage Service

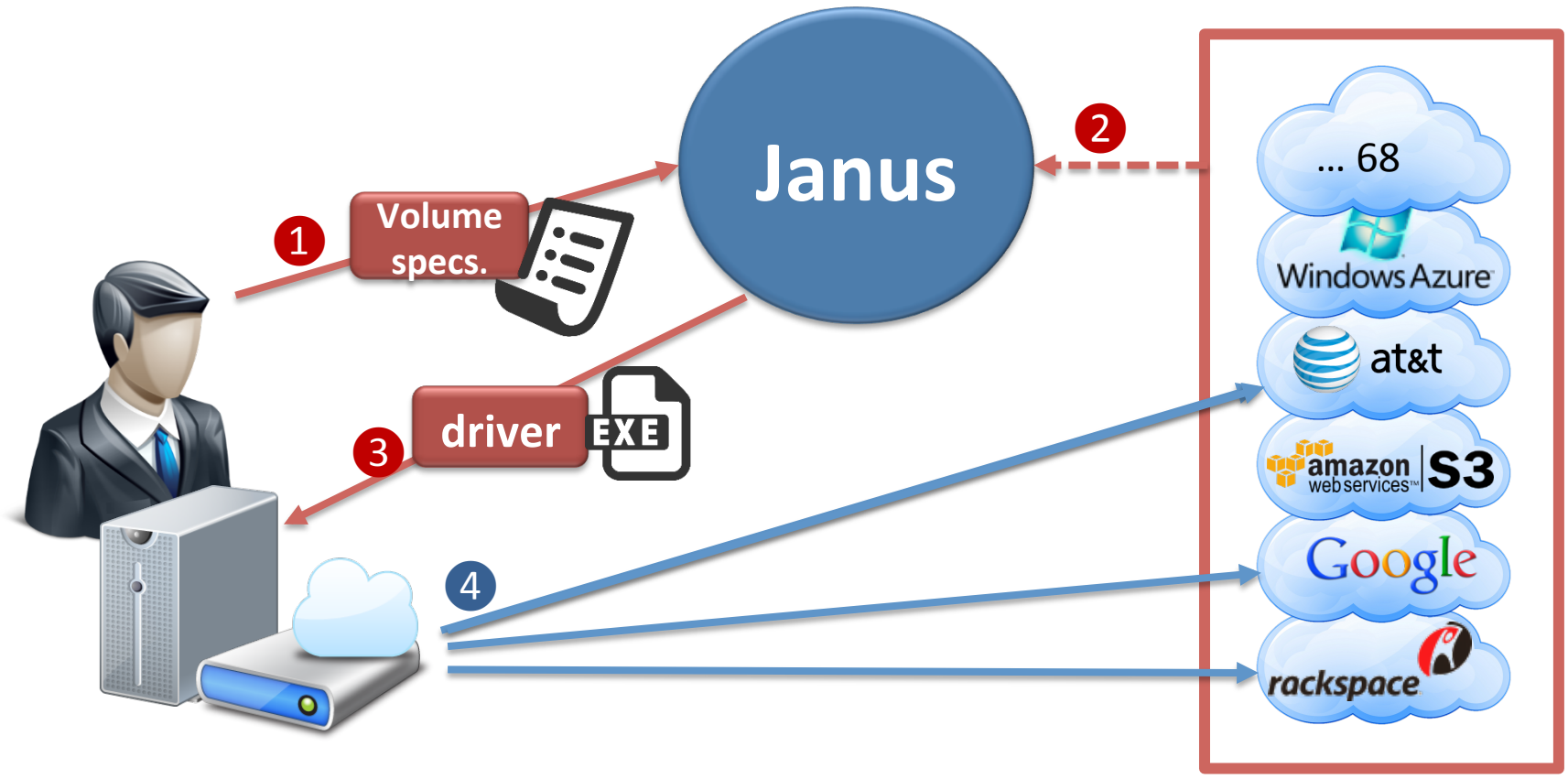
- Provides Secure multi-cloud storage

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



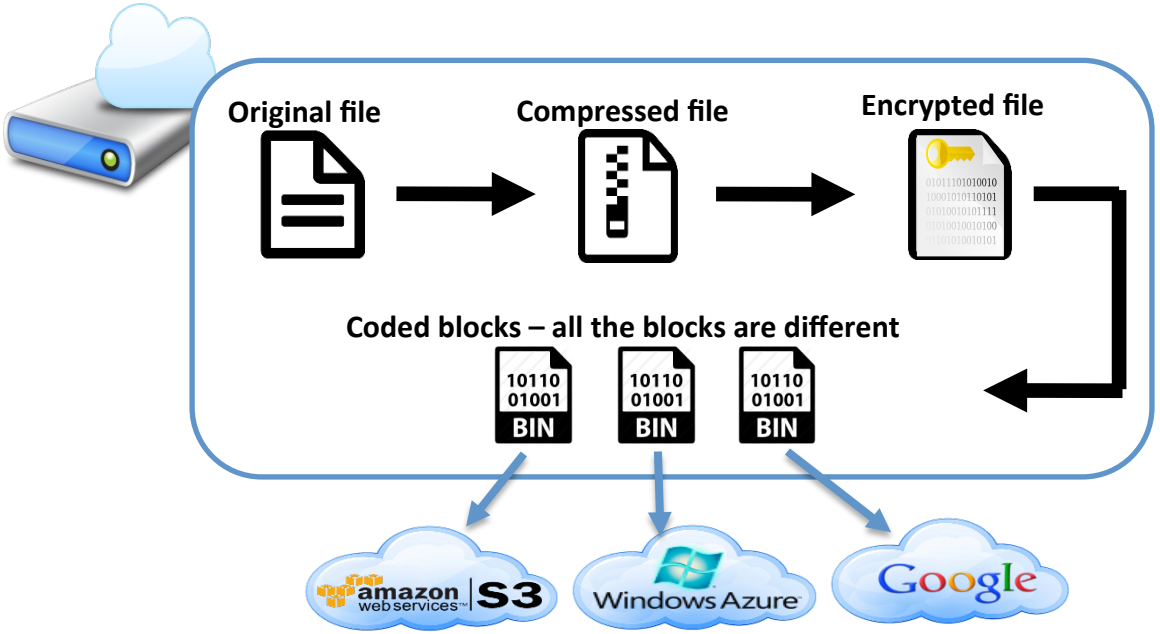
Volume Specifications

Available Options

	Options	
Workload	• Num writes • Num Reads • Num Files • Average File Size	
Sharability	No	
	Single-Account	
	Multi-account	
Computable	Yes	
	No	
Data Location	NOT IN	• Continent • Country
	ONLY IN	
	PREFER IN	
Consistency Guarantees	Eventual consistency	
	Strong consistency	
Provider Fault Tolerance	No	
	CFT	• Number of faulty clouds
	BFT	
Security Standards Compliance	COMPLY	• witch standards
	NOT COMPLY	
	PREF COMPLY	
Budget	• Max price	
SLA	• Latency • Availability SLA • Durability SLA	
Available Cache	• Disk space	

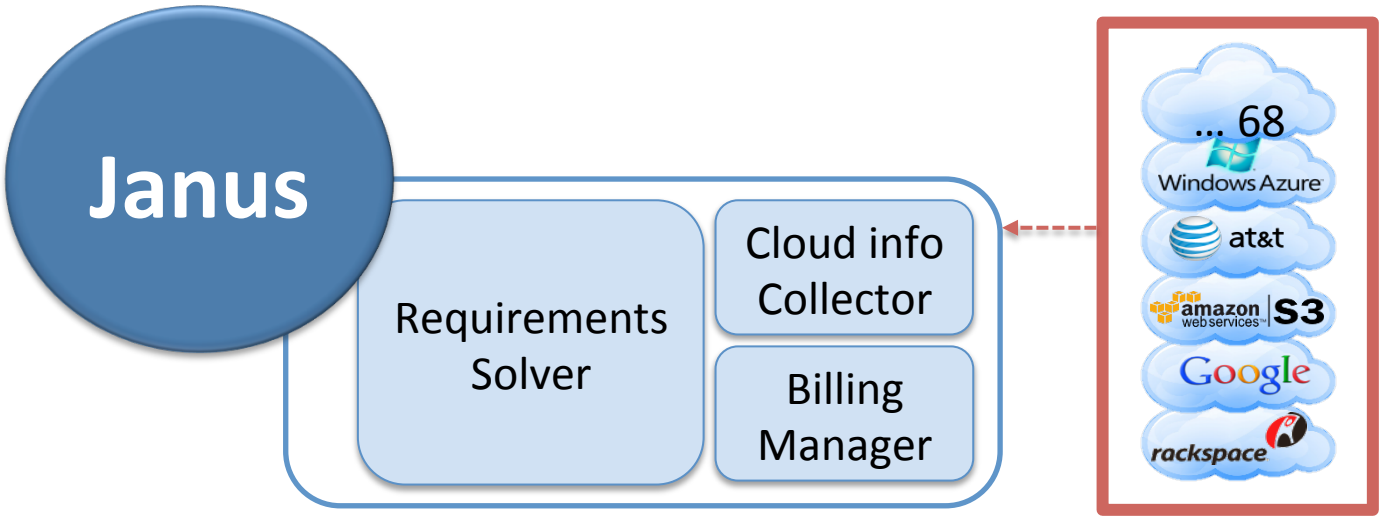
Virtual Disk Driver

- **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



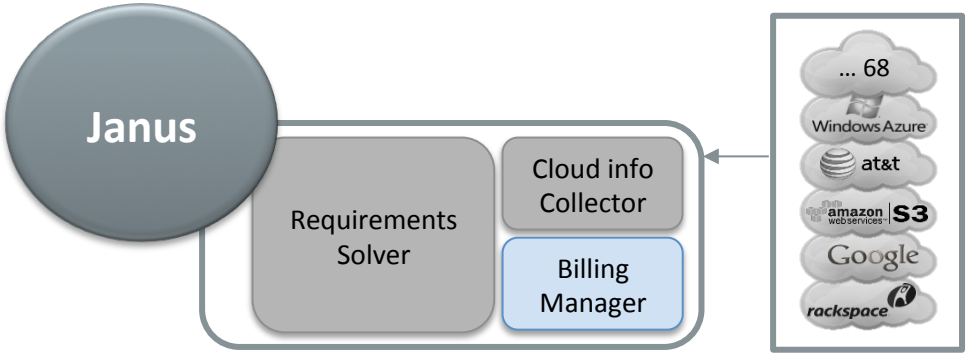
Janus Server

- **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



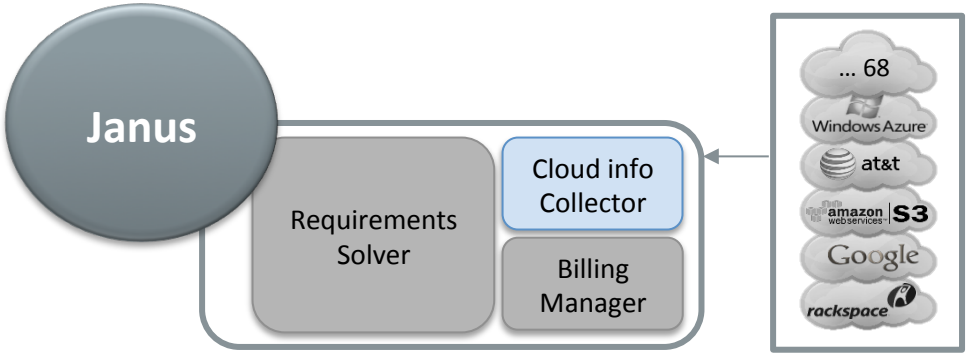
Billing Manager

- 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



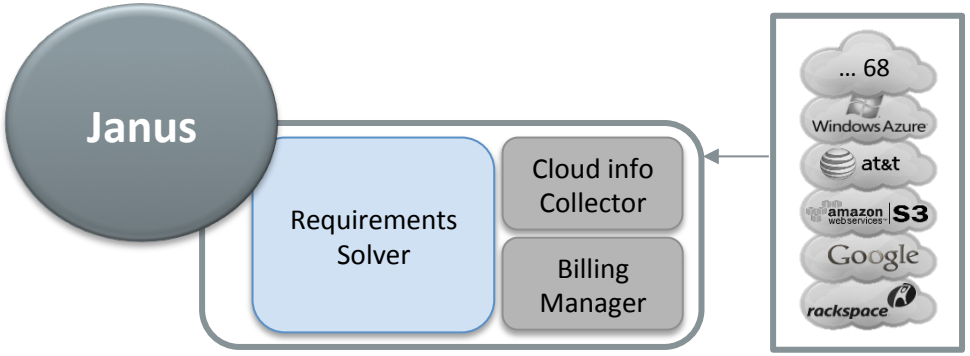
Cloud info Collector

- 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 need



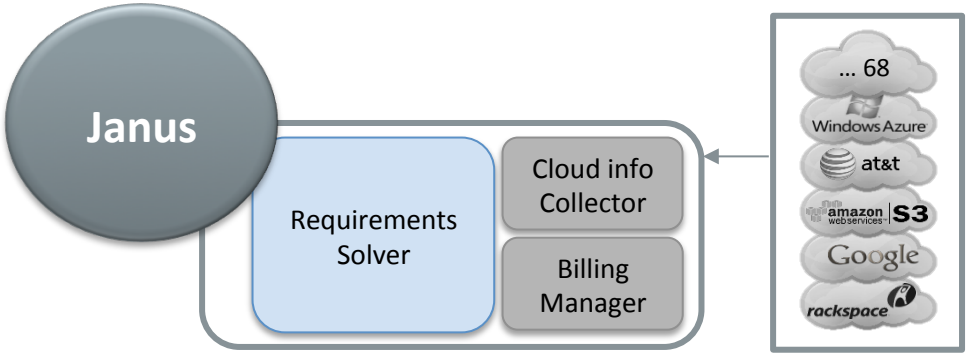
Requirements Solver

- **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) ≈ 300 Billion → x 1μs ≈ 3 days

- Minimize this by separate this task in three steps:
  - 1st: **Filter** the cloud services according to the requirements
  - 2nd: **Sort** the clouds according to a linear function $f(x)$ which quantifies the interest of cloud x to the configuration
  - 3rd: **Combine** the services to obtain the m best configurations

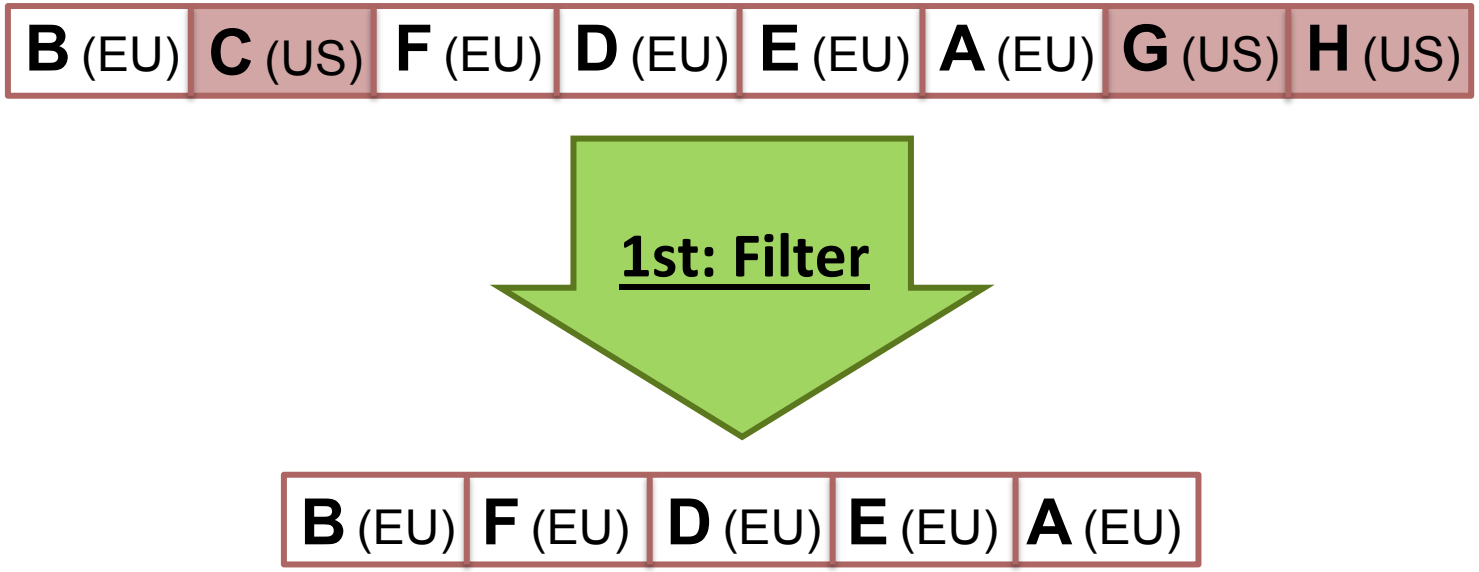
Query Solving Strategy

- n=2f+1, f = 1 → n=**<u>3 clouds</u>**
- Only EU
- m = 4

B (EU)	**C** (US)	**F** (EU)	**D** (EU)	**E** (EU)	**A** (EU)	**G** (US)	**H** (US)

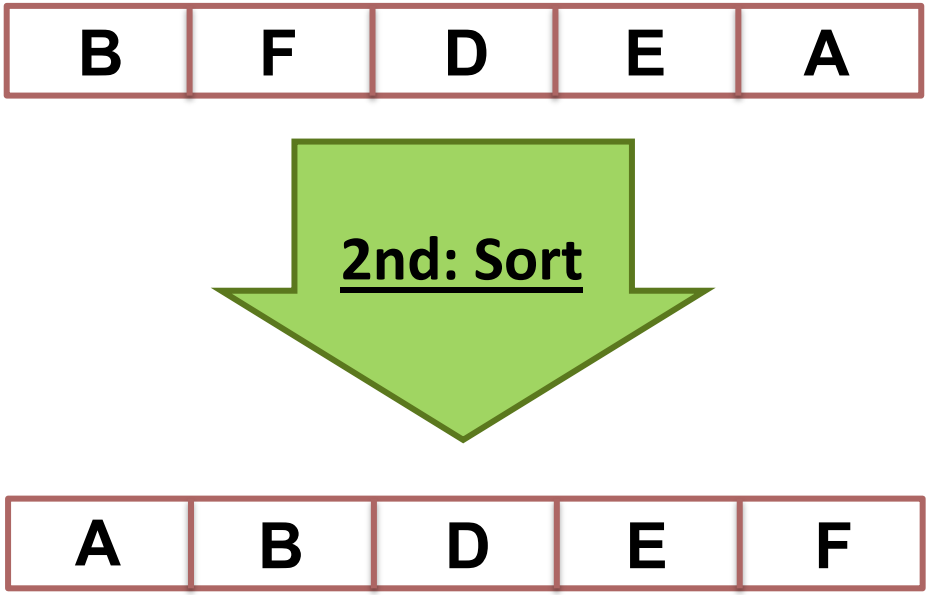
Query Solving Strategy

- n=2f+1, f = 1 → n=**<u>3 clouds</u>**
- **<u>Only EU</u>**
- m = 4



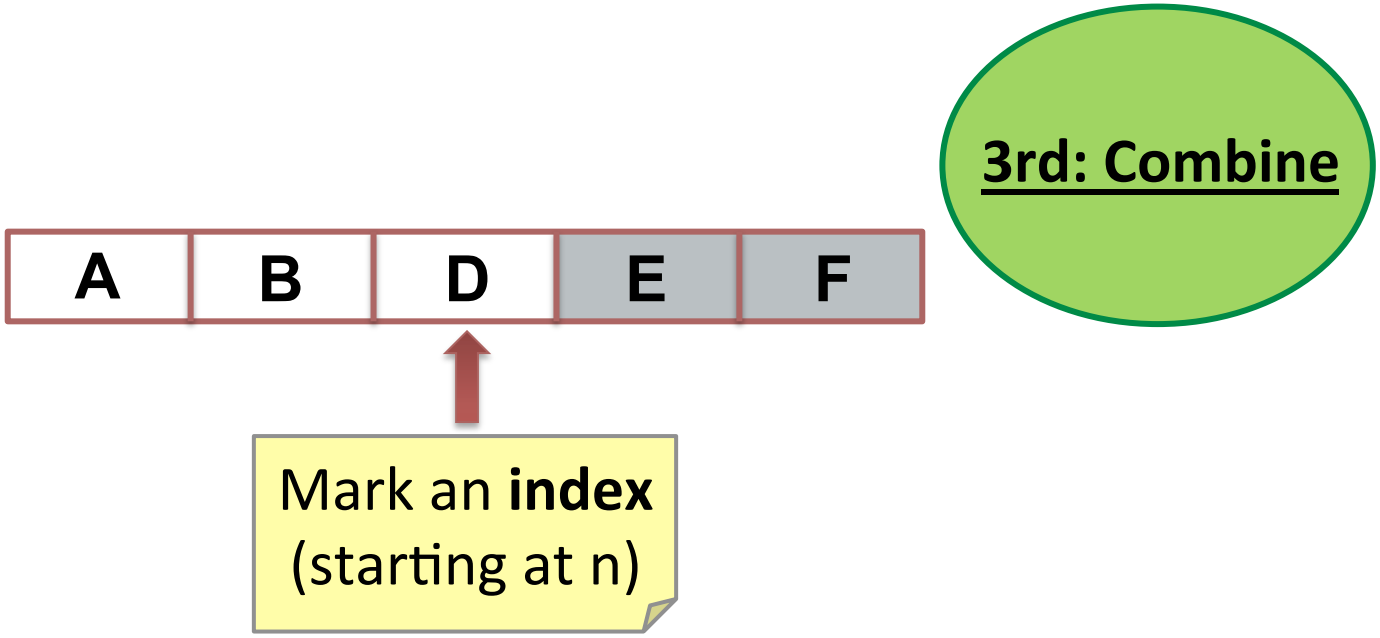
Query Solving Strategy

- n=2f+1, f = 1 → n=**<u>3 clouds</u>**
- Only EU
- m = 4



Query Solving Strategy

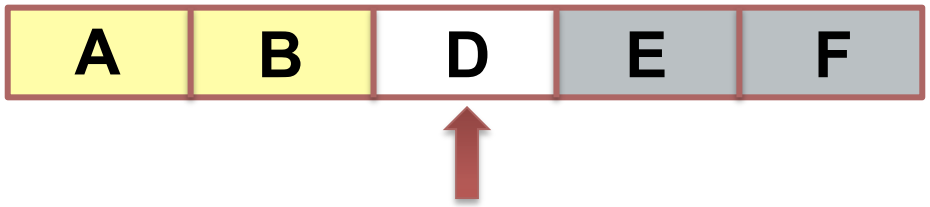
- n=2f+1, f = 1 → n=**<u>3 clouds</u>**
- Only EU
- m = 4



Query Solving Strategy

- n=2f+1, f = 1 → n=**<u>3 clouds</u>**
- Only EU
- m = 4

<u>3rd: Combine</u>



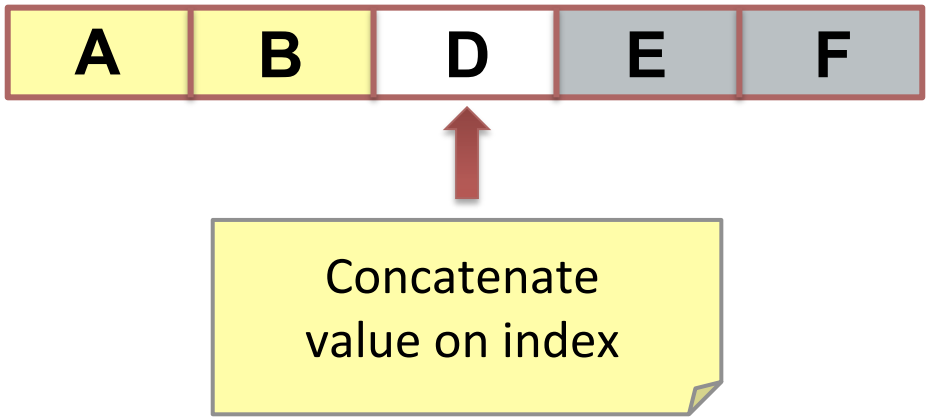
- <**<u>A,B</u>**>

Generate all combinations of n-1 elements before the index

Query Solving Strategy

- n=2f+1, f = 1 → n=**<u>3 clouds</u>**
- Only EU
- m = 4

<u>3rd: Combine</u>

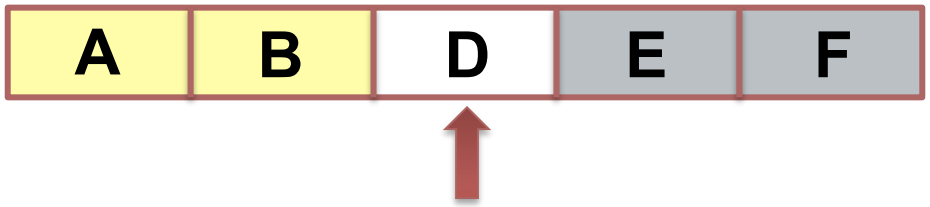


- <A,B,**<u>D</u>**>

Query Solving Strategy

- n=2f+1, f = 1 → n=**<u>3 clouds</u>**
- Only EU
- m = 4

<u>3rd: Combine</u>

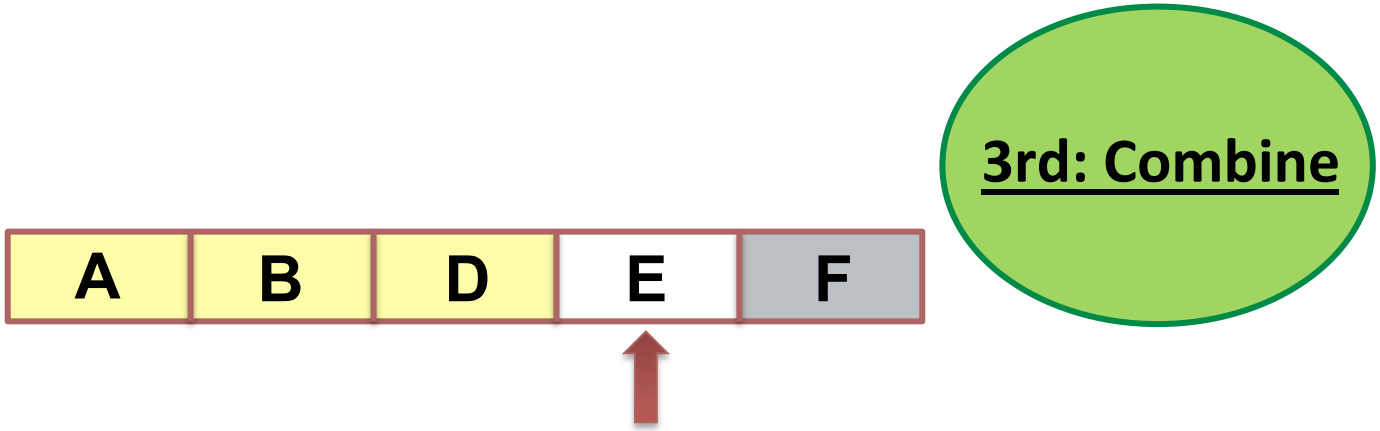


- <A,B,D>

Increase **index** if the number of combinations is smaller than m

Query Solving Strategy

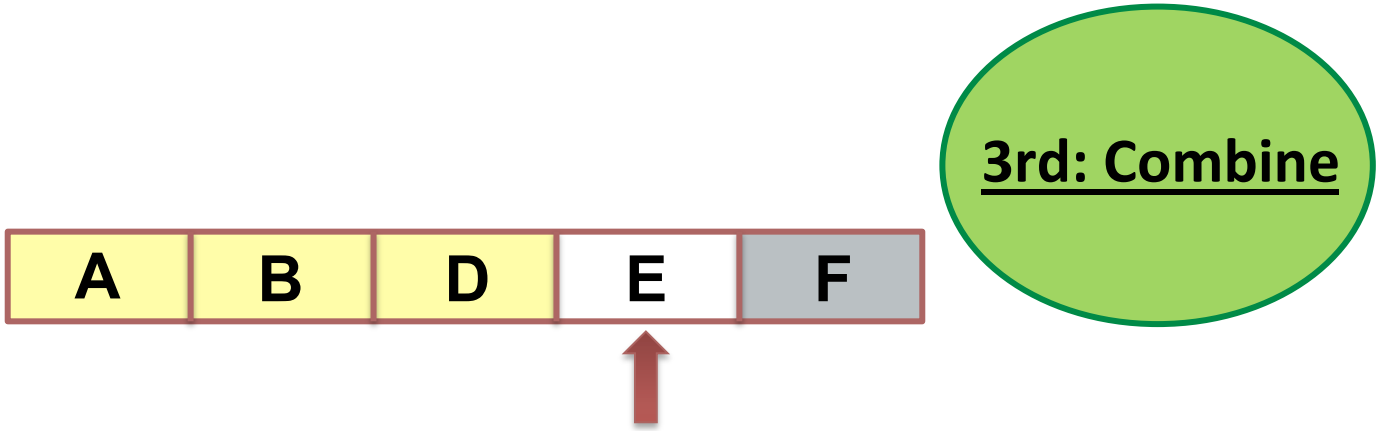
- n=2f+1, f = 1 → n=**<u>3 clouds</u>**
- Only EU
- m = 4



- <A,B,D>

Query Solving Strategy

- n=2f+1, f = 1 → n=**<u>3 clouds</u>**
- Only EU
- m = 4

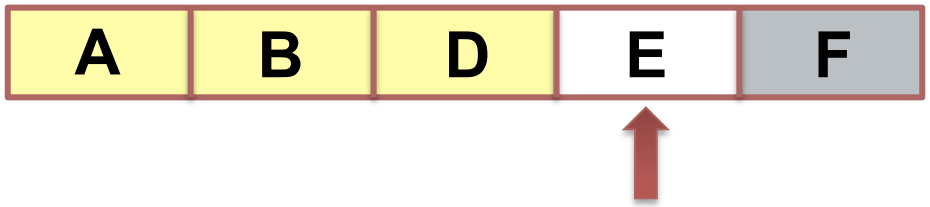


- <A,B,D>
- <**<u>A,B,E</u>**>
- <**<u>A,D,E</u>**>
- <**<u>B,D,E</u>**>

Query Solving Strategy

- n=2f+1, f = 1 → n=**<u>3 clouds</u>**
- Only EU
- m = 4

<u>3rd: Combine</u>



- <A,B,D>
- <**<u>A,B,E</u>**>
- <**<u>A,D,E</u>**>
- <**<u>B,D,E</u>**>

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