



# **Global SQL platform on inter-cloud computing environment**

**Sep., 2016**

Ikuo Nakagawa, Hiroki Kashiwazaki, Shinji Shimojo,  
Kohei Ichikawa, Tohru Kondo, Yoshiaki Kitaguchi





# Background



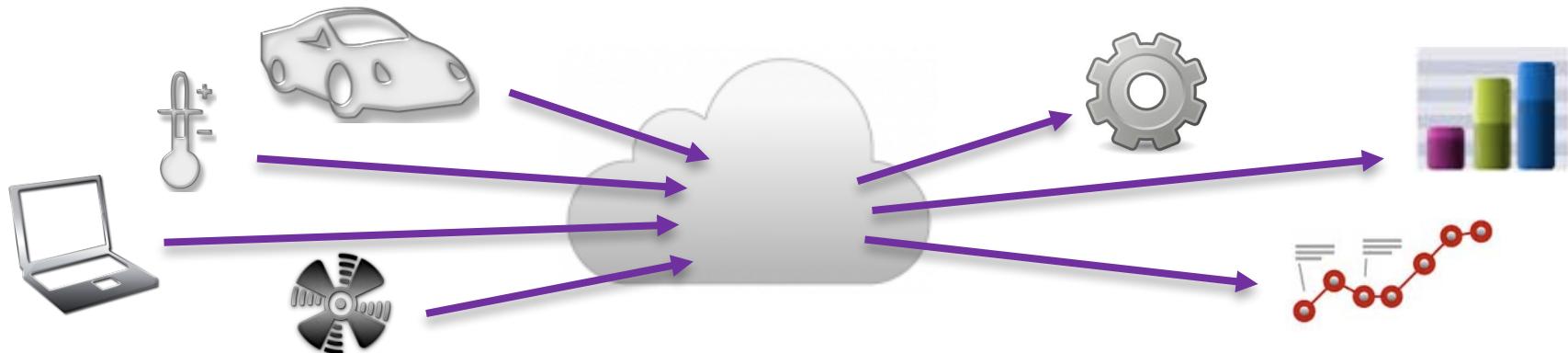
# Background : IoT and its data

## ▶ IoT (Internet of Things)

- a very large number of devices (billions or more)
- huge amount of data (unlimited scale)
- into cloud, to everywhere

## ▶ Accessing data

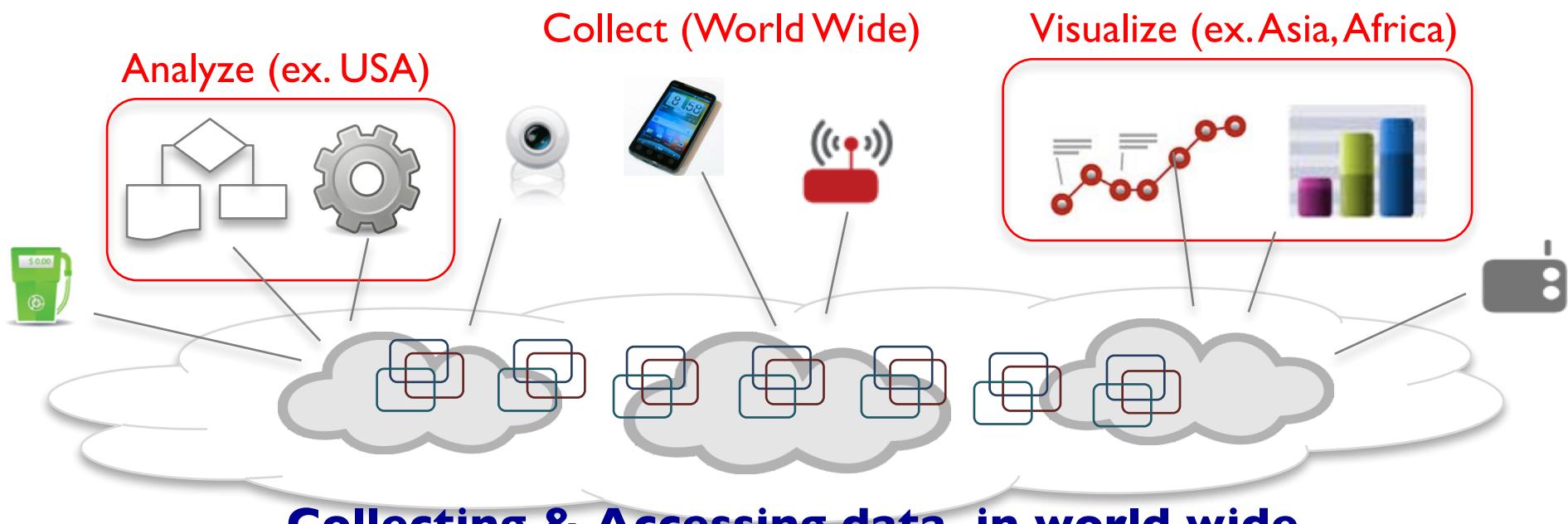
- for visualization, for analyzation, for statistical computation, ...
- for academic research purposes or application services
- from cloud, at anywhere



# Goal

# Goal : World Scale IoT Data Platform

- ▶ Collecting data at anywhere
  - Devices are connecting to the Net, everywhere
- ▶ Accessing data at anywhere
  - Processing, Analyzing, Visualizing, in various locations



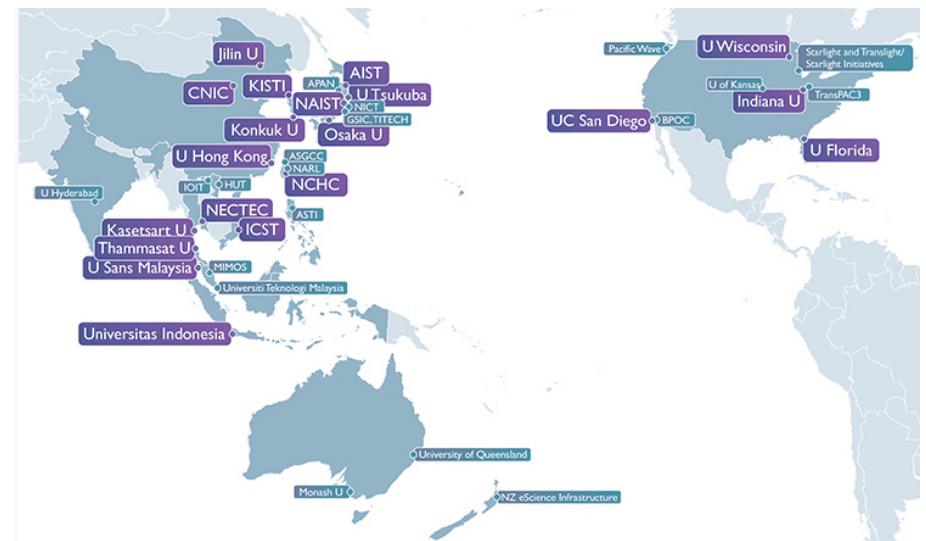


# Challenges

# Challenge-2 : Inter-cloud

## Many organizations in PRAGMA

distributed, multiple, independent computing resources  
collaboration is the key



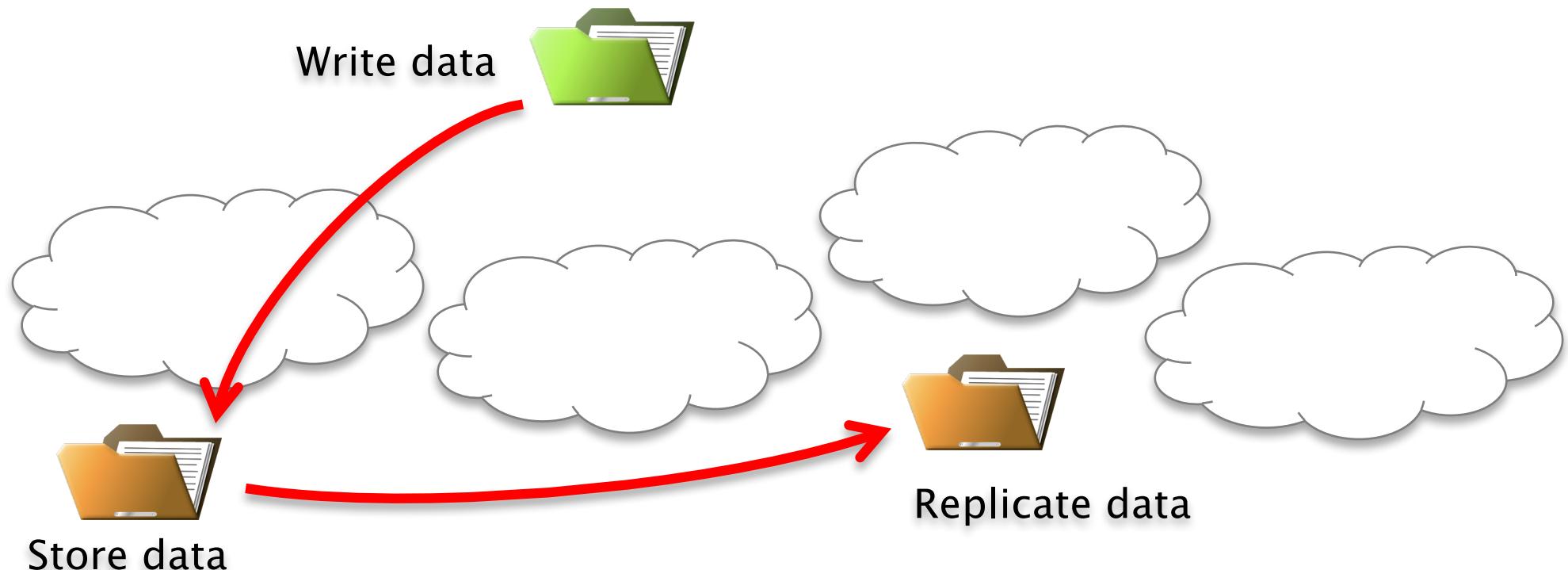
## Many leading-edge services

various leading-edge and useful cloud services  
multiple-cloud operation is a big trend



## Challenge-2 : Disaster Aware

- ▶ DR / BCP are very important, of course
  - replicate data in transpacific-environment, in PRAGMA
  - data would be available even a case of continental-class disaster



## **Challenge-3 : Simplicity**

Making it simple, awesomely simple. That's creativity.

**“Simplicity”**

Biggest challenge for simplicity is,

**“Nothing new, for users”**



# **Design and implementation**



## Design concept

Simplest way for collecting data  
**“Write into a file”**



→ Distcloud file system

analytic tools, applications, may want

**“Read via SQL”**



→ Apache Drill (on Distcloud)

# **Distcloud File System**

## new Software Defined Storage

with

### Inter-cloud computing environment

not only academic, but also on AWS, Azure, GCP, SL or even private

### fully distributed architecture

no centralized server, no SPoF, distributed in world scale, multiple-site, all-active (NOT primary-backup)

### DR (Disaster Recovery) feature, by nature

even for continent class disaster

### POSIX like file system interfaces

intuitive and easy to use for existing applications, nothing new for users

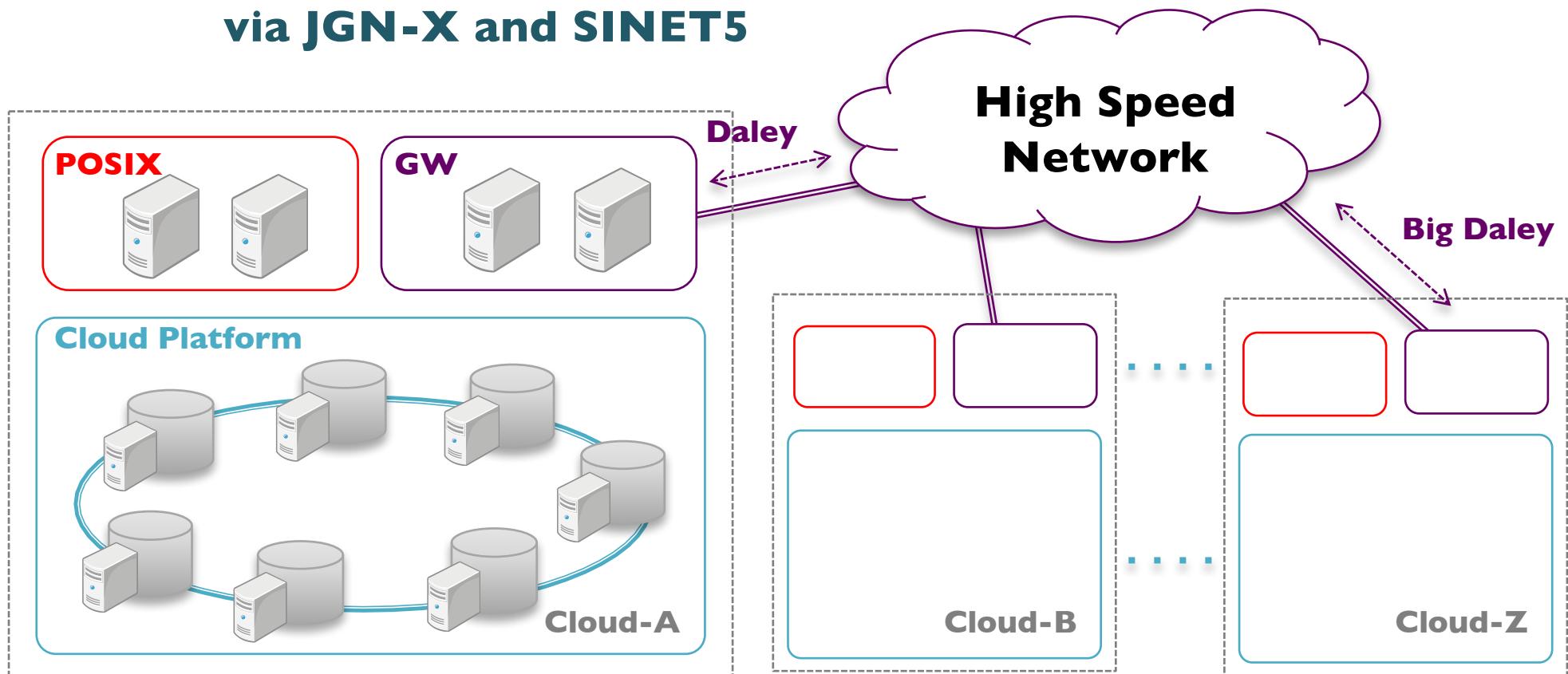
### native Java implementations

since we like Java ☺

# Distcloud Project

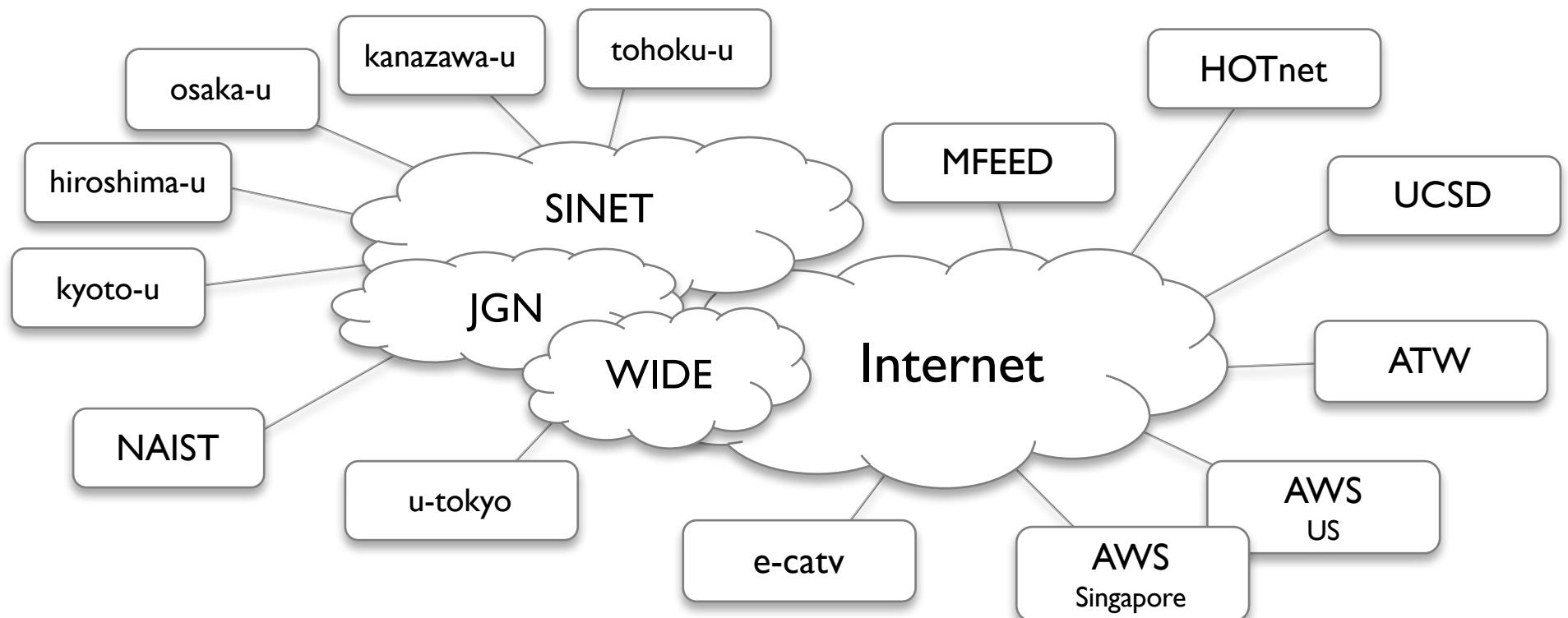
**world scale distributed file system  
Fully distributed, No RDB, No centralized system**

**Interconnect universities, labs, etc.  
via JGN-X and SINET5**



# Distcloud - Experimental Testbed

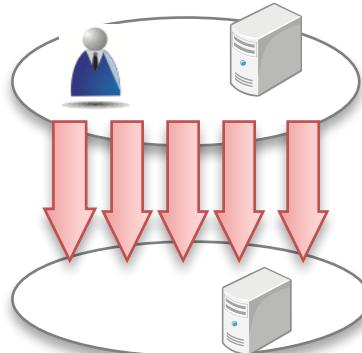
- ▶ We are expanding inter-cloud environment
  - Universities, Research Organizations, Service Providers, ...



# Challenges in Distcloud File System

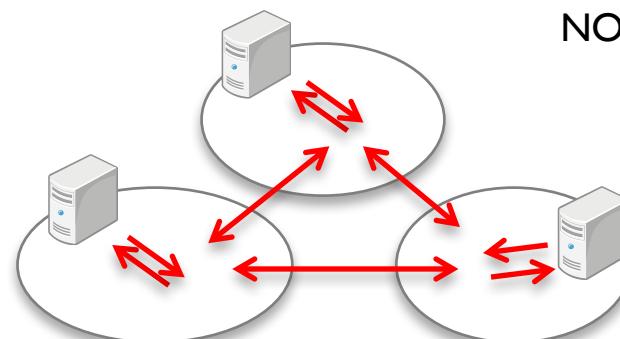
**One big challenges is achieving inter-cloud feature, with globally distributed & strict consistency**

## Improve throughput



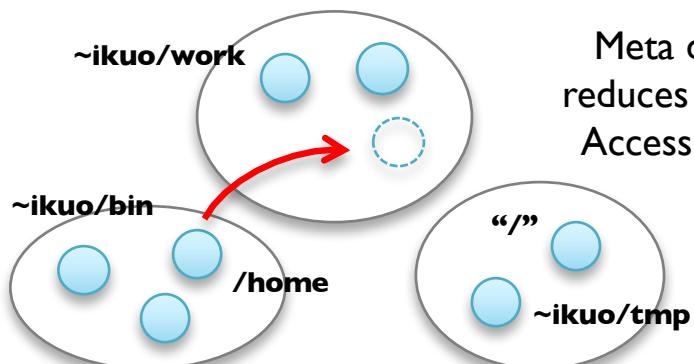
Solution for global env.  
“TCP throughput” issues.  
Large number of  
parallel TCP sessions  
improve total performance

## Multiple site ( $N>2$ ) & All Active



NOT primary/backup.  
Support all-active

## Optimize BIG delay



Meta data migration  
reduces latency issues  
Access locality is key

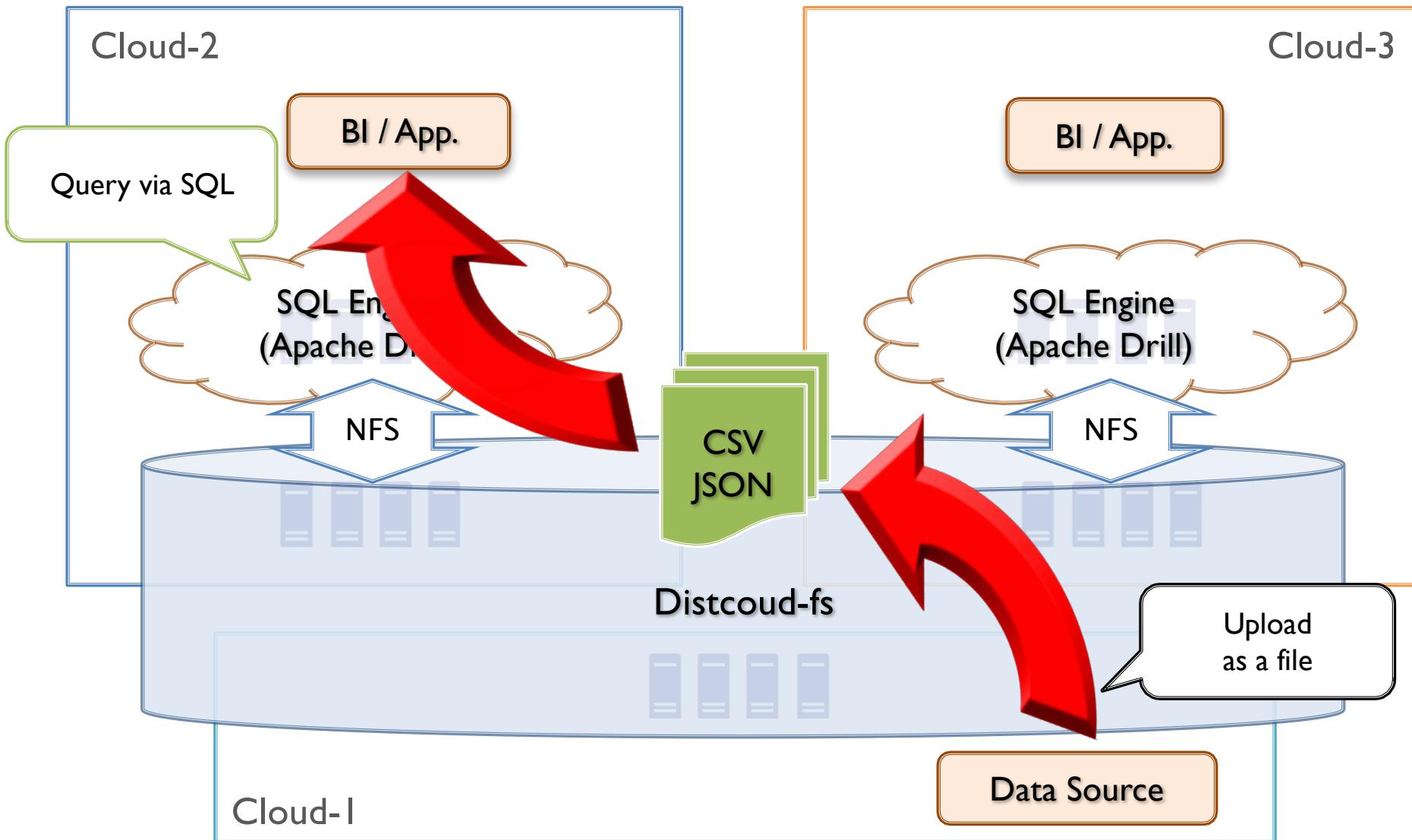
## Strict consistent platform for VMs



Even VM live migration works on distcloud  
since we provides strict consistent POSIX.

# Apache Drill on Distcloud File System

Apache Drill (SQL frontend) over the Distcloud File System



# Benefit of the architecture

- ▶ **Inter-cloud** – World scale
  - Works with many, independent cloud services
- ▶ **DR/BCP** – Resilient, by nature
  - Always replicates data, over trans-ocean
- ▶ **Very Simple** – Nothing new, for users
  - Write into a file (into POSIX file system, by Distcloud)
  - Read via SQL (by Apache Drill)

(\*) Other benefits also supported,  
high scalability, global distribution (world scale), parallel processing, ...

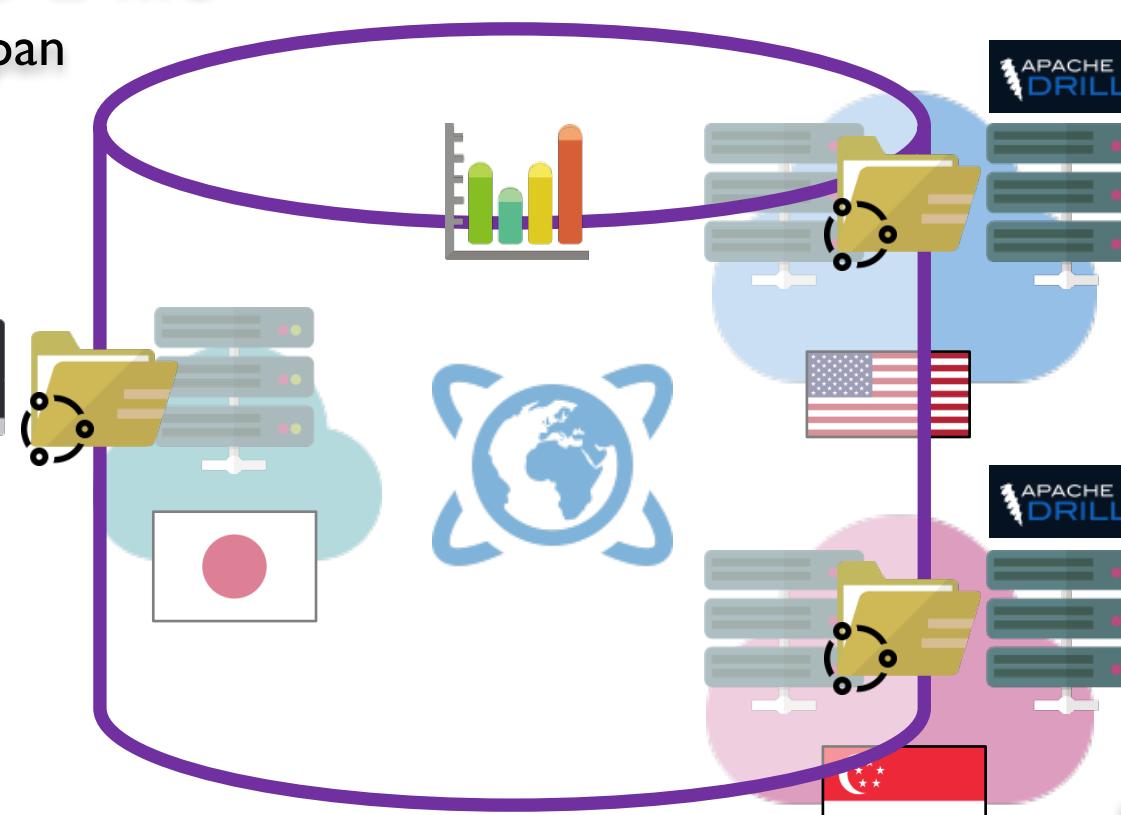
# Demonstration

- ▶ Global file system on inter-cloud environment
  - Our Private Cloud in Japan
  - Public Cloud in US
  - Public Cloud in Singapore
- ▶ Write into a file, into global file system
  - from Japan
- ▶ Read via SQL (Apache Drill) on various cloud
  - on US and on Singapore, in the demonstration

# Demonstration scenario

Write into a file

on cloud-1 in Japan



Read via SQL

on cloud-2 in US



Read via SQL

on cloud-3 in Singapore

POSIX global file system  
on inter-cloud computing environment

# Demonstration

[See Demonstration Video](#)

# Conclusion – Our data platform is...

- ▶ **Inter-cloud** – World scale
  - Works with many, independent cloud services
- ▶ **DR/BCP** – Resilient, by nature
  - Always replicates data, over trans-ocean
- ▶ **Very Simple** – Nothing new, for users
  - Write into a file (into POSIX file system, by Distcloud)
  - Read via SQL (by Apache Drill)

# Future Researches

- ▶ Performance tuning
  - Reading via Apache Drill is the key
- ▶ Upload data via SQL
  - Apache Drill is only for READ data via SQL
- ▶ Encryption
  - Encrypt when data is out of your hands
  - Decrypt when Engine reads your data
- ▶ Operation
  - Single operation ?
  - Manage multiple independent cloud services ?

# *Thank you !!*