Big Data in OpenStack Storage

Ivan Tomašić, Aleksandra Rashkovska, Matjaž Depolli, Roman Trobec

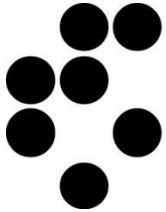
Department of Communication Systems

Jožef Stefan Institute, Ljubljana, Slovenia

KC Class

openstack
CLOUD SOFTWARE

Outline



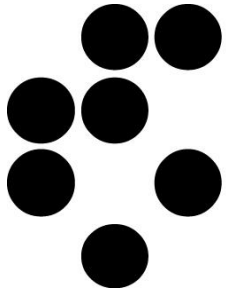
- Introduction
- Swift in the CLASS project
- Experience with installation of OpenStack Storage
- Hybrid Cloud Storage

  (OpenStack Storage and AmazonS3)
- Conclusion

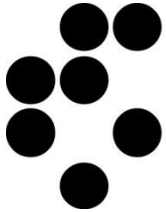
Experience with installation of OpenStack Storage

Ivan Tomašić

Ivan.Tomasic@ijs.si

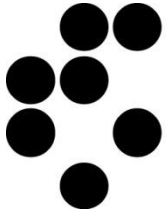


Introduction



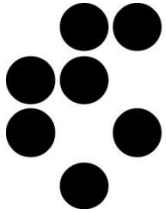
- OpenStack Object Storage in the CLASS project
- OpenStack Object Storage (Swift) installation:
  - Hardware and software platform
  - Current Swift installation on IJS
  - Installation details
  - Work on progress
- Benchmarks
- Conclusion

Swift in the CLASS project



- IJS group is responsible for CLASS/Petabyte storage
- Until M6 we identified requirements for IaaS needed for CLASS/Petabyte system and recommendations for its users
- We determined 12 main criteria for comparison of the available systems: scalability, data model, failure handling, compatibility, security, …
- Three types of storage systems were distinguished regarding their data model:
  - relational (SQL databases),
  - NoSQL databases and
  - Systems that store unstructured data objects.

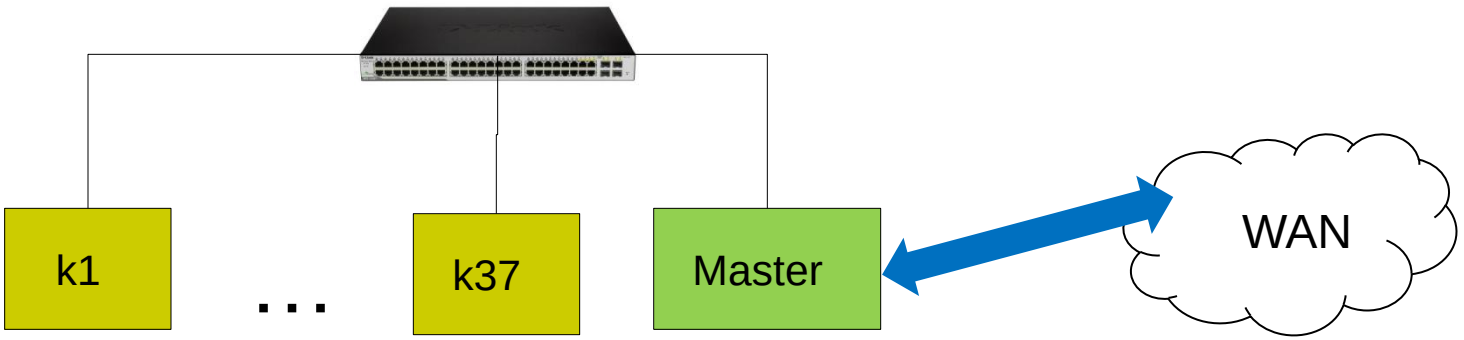
Swift in the CLASS project – cont.



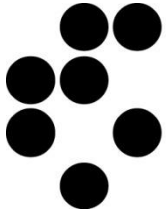
- We analyzed open source systems: WALRUS (Eucaliptus), SWIFT, LUSTRE, TWISTED STORAGE, TASHI, SECTORE/SPHERE, HADOOP, MYSQL CLUSTER, MONGO DB
- And commercial cloud-based storage: MICROSOFT AZURE, ORACLE, AMAZON S3, GOOGLE
- We selected SWIFT (Object storage) and HADOOP (DFS) for test implementation and testing.
- We foresee different application scenarios:
  - A single user with high storage requirements (private)
  - Multiple users of shared data
  - A single user with dynamic amount of data (need for extension of private storage with public storage)
  - Same as above, but for shared data
  - Some other scenarios could appear?

Hardware and software platform

- Cluster on Jožef Stefan Institute (JSI):
  - 37 nodes:
    - Master node
    - Working nodes
    - Each node is independent machine
  - Nodes are connected with Gigabit Ethernet

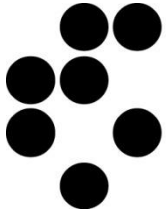


Hardware specifics and operating system



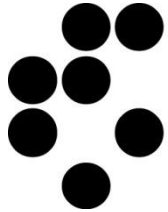
- The base is the HP DL160 G6 server:
  - up to 2 processors
  - up to 18 x 16 GB of RAM
- Processor Intel Xeon 5520 (4 cores and hyperthreading
- 6 GB RAM (DDR3, ECC, 1060 MHz, 3 channels)
- 500 GB hard disk (3.5 in, SATA, 7.2 K rotations/min)
- **RAID - only the master node** has 4 HDs:
  - RAID 1+0 with 917 GB of storage space altogether
- All the nodes have 64-bit Ubuntu Server 11.04

Swift installation on IJS



- Current Swift installation on IJS has the following configuration:
  - Master node is running:
    - Proxy server
    - Authorization server
- Working nodes k10, k11, k12, k13, k14 are each running:
  - Object services
  - Container service
  - Account services

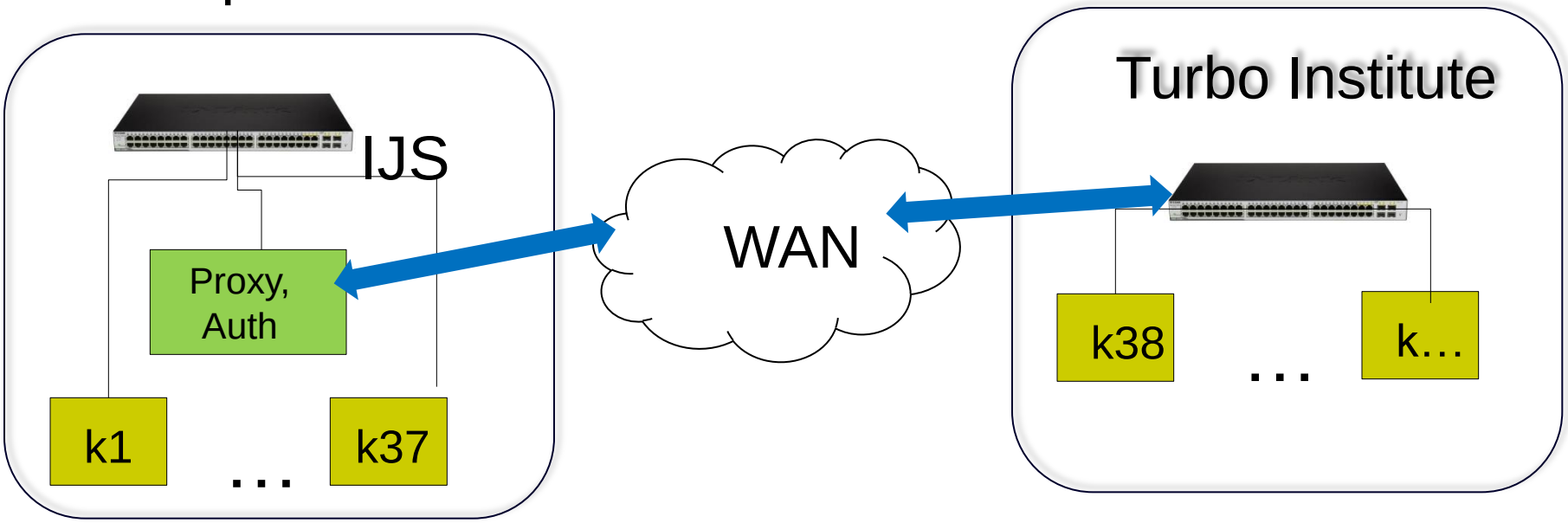
Current IJS Installation details



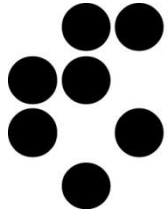
- Number of partitions: 2^9=512
- Number of replicas: 3
- Number of zones: 5
- XFS file system (XATTRS) – recommended
- Authorization:
  - Swauth
  - TempAuth – is used at the present time

Work in Progress

- Extend Swift to the whole cluster
- Perform additional tests and benchmarks (Hadoop)
- Multiple data centers installation:

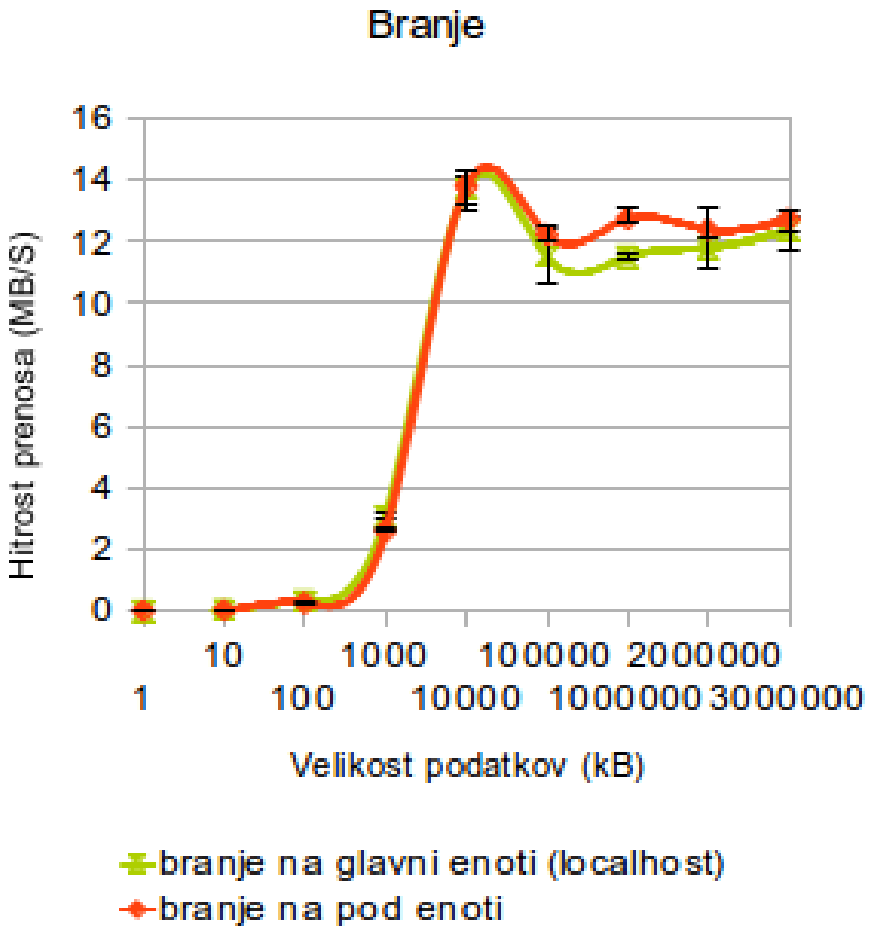


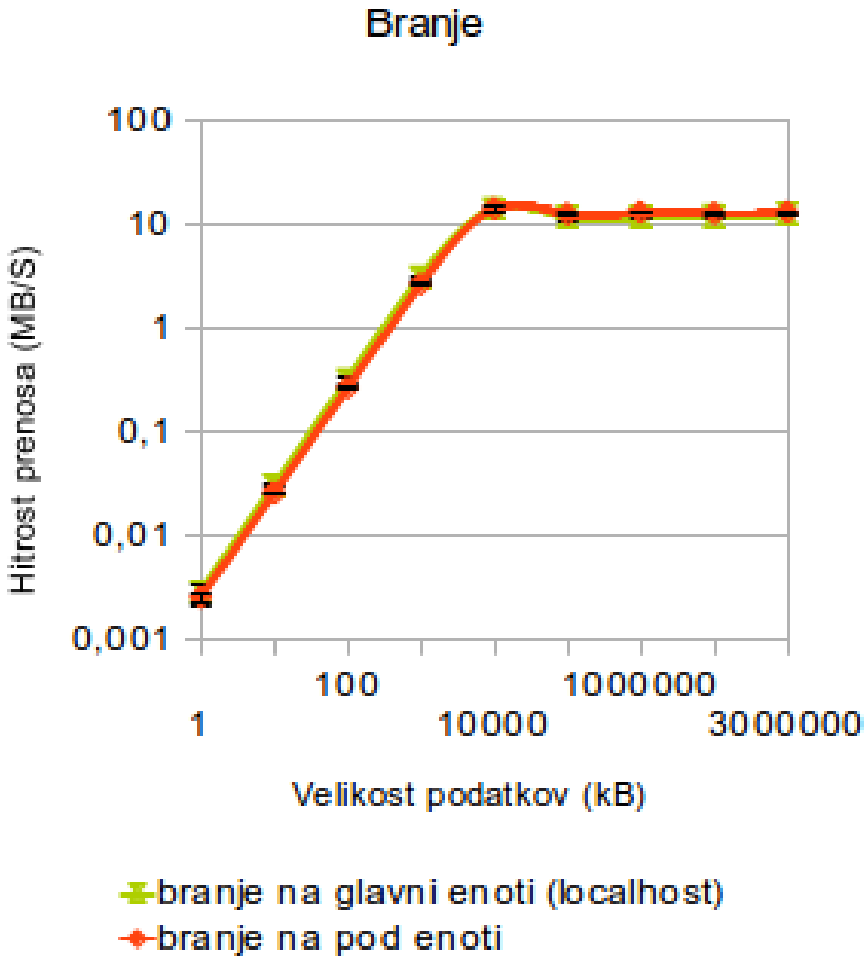
First Benchmark Results



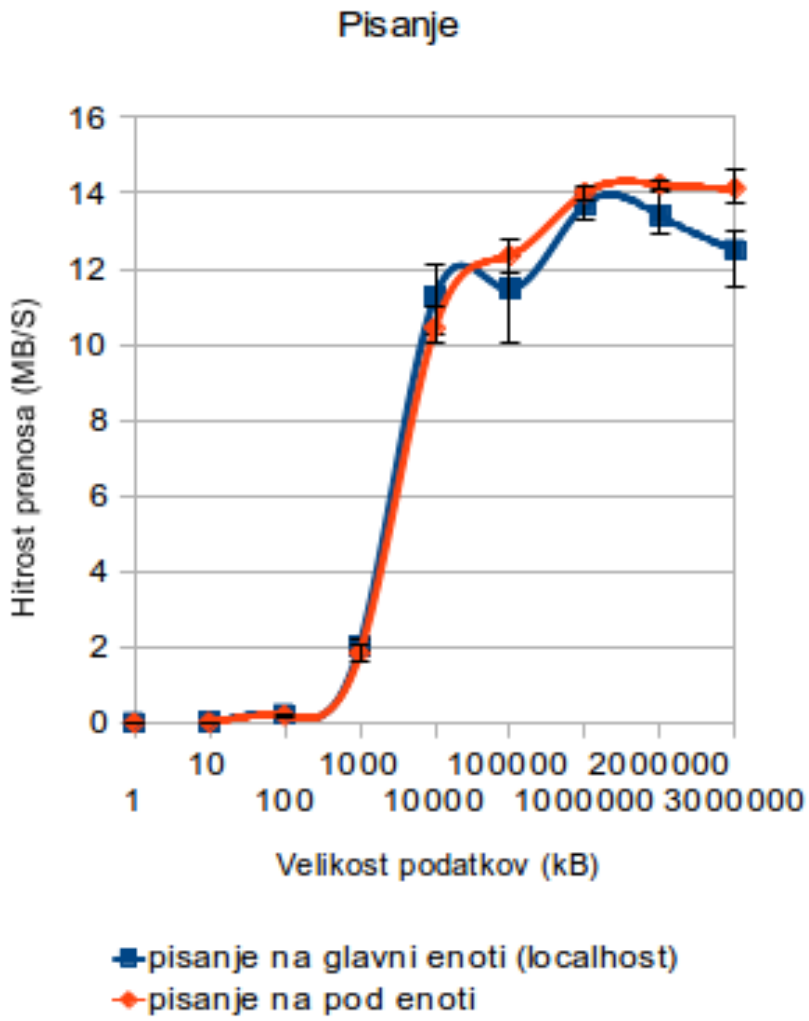
- Test procedure:
  - Writting nad reading
  - Two types of access:
    - From master machine (local host)
    - From distant machine - 1Gb/s connection
- Expected values:
  - Speed from master machine **>** speed from distant machine
  - Writting and reading speeds **>** 50 MB/s

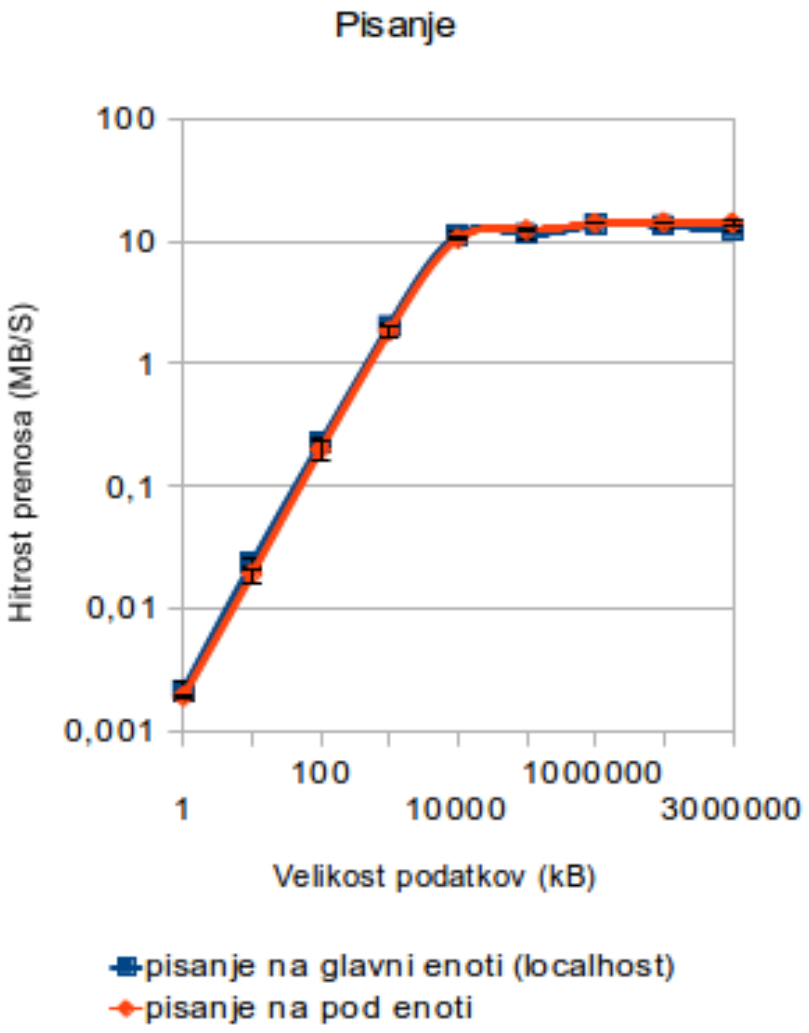
First Benchmark Results - Reading



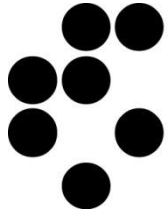


First Benchmark Results – Writing



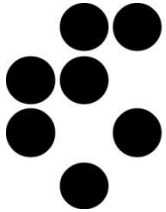


Benchmark Results - Comments



- Reading and writting from distant machine is a bit faster - unexpected
- We noticed that while read or writte operation one core on the master node is at 100% utilization
- We expected higher speeds
- System fine tuning -> reruning benchmarks

Conclusion

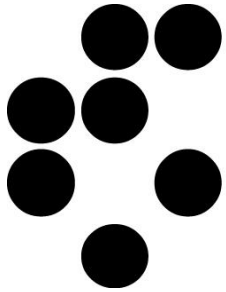


- Difference in bandwidth up to 10x between Swift and Hadoop in reading and writing to the same architecture.
- We have to find out more details about performance
- Perform additional tests and benchmarks, also between multiple data centers installations
- Result – system evaluation and prescription for fine tuning

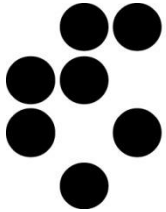
Hybrid Cloud Storage
(OpenStack Storage and AmazonS3)

Aleksandra Rashkovska

Aleksandra.Rashkovska@ijs.si



Outline

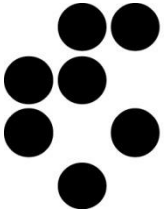


- Cloud Storage: public, private, hybrid.
- Approaches for migration to public cloud storage:

  Three routes to implementation.
- OpenStack Storage and AmazonS3:

  Implementation examples.

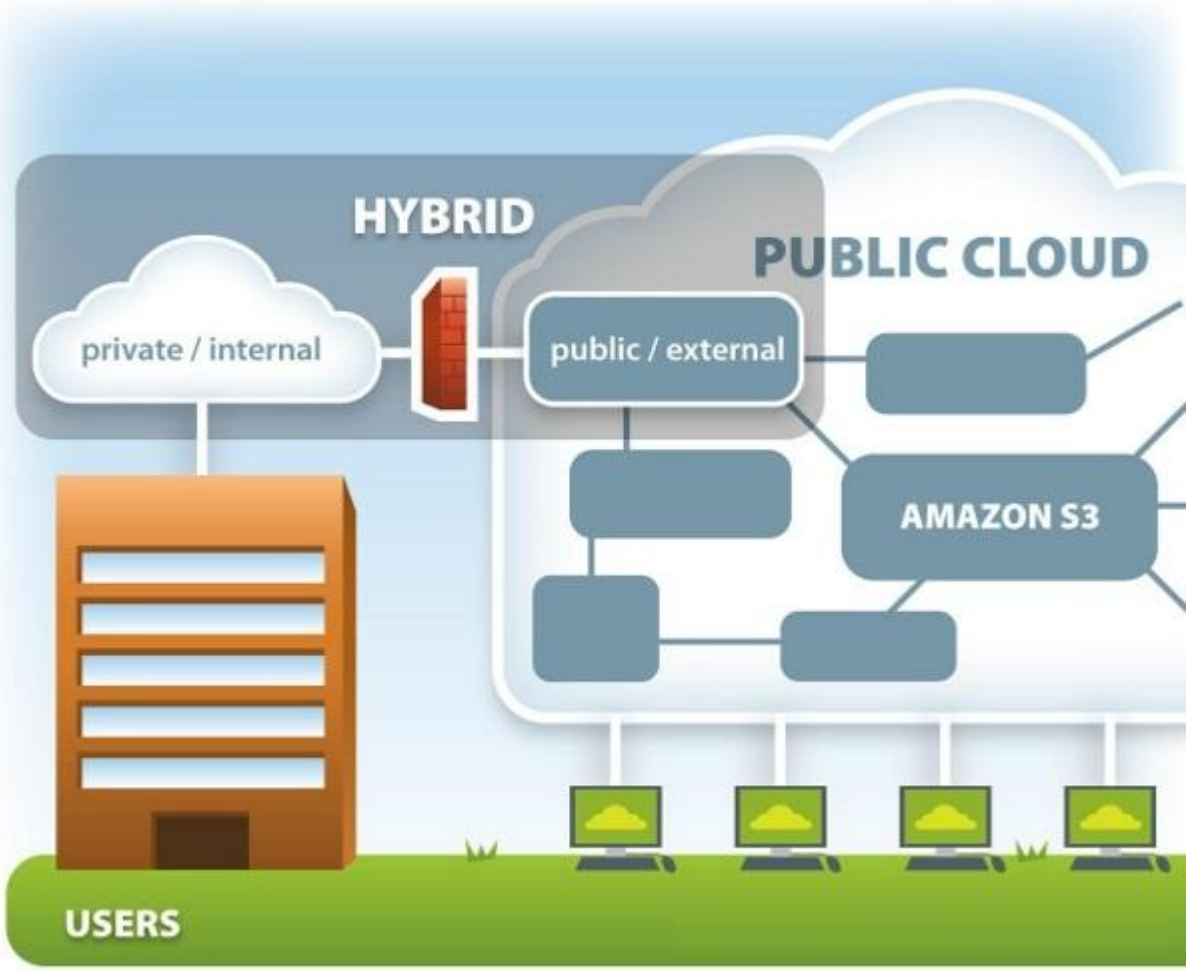
Public vs. Private Cloud Storage



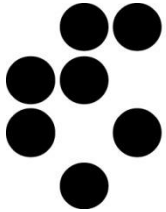
- ***Public Cloud Storage***
  - Cloud storage option offered by a fast growing list of service providers (Amazon, AT&T, Iron Mountain Inc., Microsoft Corp., Nirvanix Inc., Rackspace Hosting Inc.)
  - Infrastructure: usually low-cost storage nodes with an object-based storage stack.
  - Data in the cloud is accessed mostly via REST and SOAP.
  - Redundancy is achieved by storing each object on at least two nodes.
  - Usage is charged on a dollar-per-gigabyte-per-month basis.
  - Designed for massive multi-tenancy that enables isolation of data, access and security for each client.
- ***Private Cloud Storage***
  - Runs on dedicated infrastructure.
  - Usually for a single tenant.
  - Do not scale to the degree public storage clouds can.

Hybrid Cloud Storage

The best of both worlds

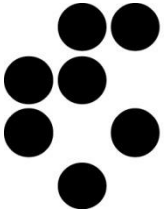


Hybrid Cloud Storage



- *Hybrid cloud storage* → when traditional storage systems or private cloud storage are supplemented with public cloud storage.
- Key requirements :
  - The hybrid cloud storage must behave like homogeneous storage.
  - The hybrid cloud storage should be transparent.
  - Mechanisms to keep active and frequently accessed data on-premise and push inactive data into the cloud (policy engines to define the circumstances when data gets moved into or pulled back from the cloud)

Migration to public cloud storage



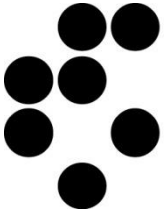
- **Three routes to implement hybrid cloud storage**
  - Via cloud storage software that straddles on-premise and public cloud storage

    (Cloud storage software implementation)
  - Via cloud storage gateways

    (Cloud storage gateways implementation)
  - Through application integration

    (Application integration implementation for hybrid cloud storage)

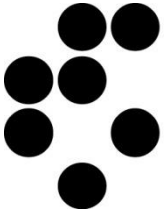
Cloud storage software implementation



- Only possible today if the internal and external storage clouds run the same cloud storage software.
- Standardization initiatives in progress:

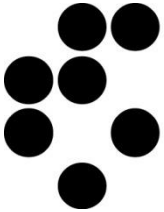
  Storage Networking Industry Association (SNIA) Cloud Data Management Interface (CDMI)
- Cloud software vendors:
  - *Nirvanix* - Nirvanix hNode internal cloud storage complemented with Nirvanix Storage Delivery Network cloud storage
  - *EMC Corp.'s Atmos* - software-based, hardware-agnostic, object-based storage stack. EMC sells Atmos to enterprises and providers, so on-premise Atmos deployments can federate with Atmos services in the cloud. EMC's most prominent customer is AT&T.

Cloud storage gateways implementation



- Cloud storage gateways sit between on-premise storage and public cloud storage.
- Cloud gateways perform data migration of data from on-premise storage into public cloud storage and vice versa, usually via policy engines.
- Cloud storage gateways vendors:
  - *Cirtas Systems Bluejet Cloud Storage Controller* - block-based cloud storage gateway appliance; currently integrated with public cloud storage services from Amazon and Iron Mountain.
  - *Riverbed Technology Riverbed Whitewater* - cloud backup appliance offering inline data deduplication; currently integrated with the AT&T and Amazon storage clouds.

Application integration implementation for hybrid cloud storage



- All public cloud storage services offer APIs to interact with private cloud storage software and cloud gateways.
- Cloud storage APIs enable custom in-house and commercial applications to tap into public cloud storage via REST interfaces.
- *Example:*

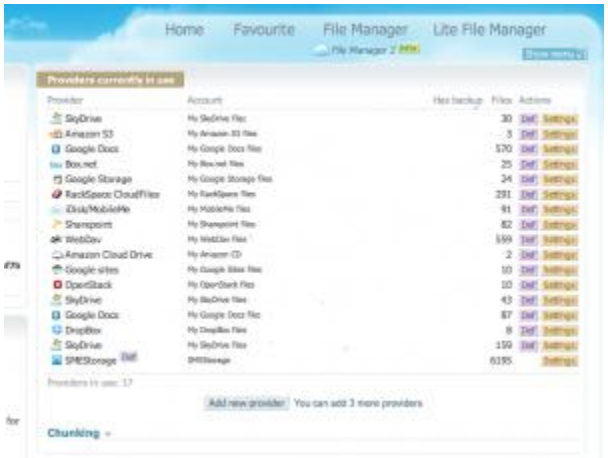
  Backup application vendors have started to add public cloud storage support to their backup suites.

OpenStack Storage and AmazonS3:

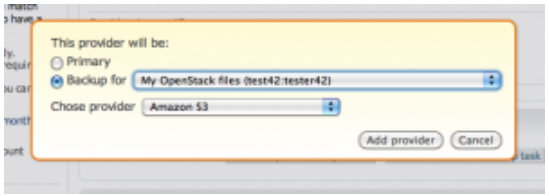
Cloud storage gateway implementation example

- Automating OpenStack Swift backup to Amazon S3 using **SMEStorage Cloud Storage gateway**
  - Requirements: Personal Lifetime Cloud or an Organisation Cloud File Server SMEStorage Account, or SMEStorage Open Cloud Platform Appliance
  - SMEStorage Cloud Dashboard:

1. Add Cloud Provider



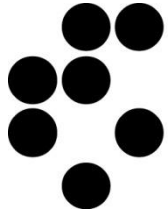
2. Select backup cloud provider



*3. E*nter Amazon S3 Details

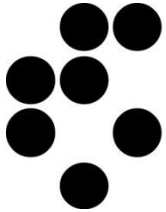
OpenStack Storage and AmazonS3:

Cloud storage APIs example



- **OpenStack AmazonS3 Compatible API**
  - Swift3 middleware emulates the Amazon S3 REST API on top of Swift.
- **Jclouds BlobStore API**
  - **Jclouds -** open source library for the cloud in java and clojure.
  - **BlobStore -** Portable means of managing key-value storage providers: AmazonS3 and OpenStack supported.

Conclusion



- Implementing hybrid clouds in data storage environment can be done in three different ways.
- Choose to implement hybrid clouds via cloud storage software, cloud storage gateways or through application integration - all viable options with several providers and products to choose from.
- Weigh your options and choose the hybrid cloud approach that best suits your storage environment.

Thank You for your attention.