THE ITALIAN RESEARCH & EDUCATION NETWORK | Consortium GARR

INFN Istituto Nazionale di Fisica Nucleare

Federazione di Object Storage e Replica multi-sito basate su Swift

Una sperimentazione congiunta INFN-GARR

F. Farina, M. Reale, P. Velati, A. Colla, A. Barchiesi, F. Galeazzi, M. Carboni

S. Stalio, G. Donvito, M. Antonacci, R. Valentini

Fabio Farina - GARR

Workshop GARR | Roma, 18-21 aprile 2016

Outline

Obiettivi e motivazioni

Introduzione all'object storage con Swift

Proof of concept

Conclusioni ed evoluzione

Genesi della collaborazione

- Incontro a CS3
  Cloud Services for Synch and Sharing
  ETH Zürich (CH) Jan 18-19 2016

- Object storage come tema ricorrente
  - Potenzialità per la comunità della Ricerca
  - Utilizzo in espansione
  - Paradigma differente
  - Non ne sappiamo abbastanza

Obiettivi concordati

- Attività congiunta GARR-INFN
  - Condivisione della conoscenza e risorse a fattore comune
  - Tecnologie di base comuni: Openstack
- Requisiti del Proof of concept
  - Gestire agevolmente i dati, grandi e numerosi
  - Verificare funzionalità non elementari
  - Validare i tool in multi-dominio, anche con federazioni d'identità
- Come procedere praticamente
  - Formulazione di un quadro generale
  - Casi d'uso reali in scala ridotta

Object storage: che cosa è?

- Analogia: bacino di pesca sportiva
  - Pesci = Oggetti
  - Vasche = Contenitori
  - Osserva, pesca, butta in mare,…

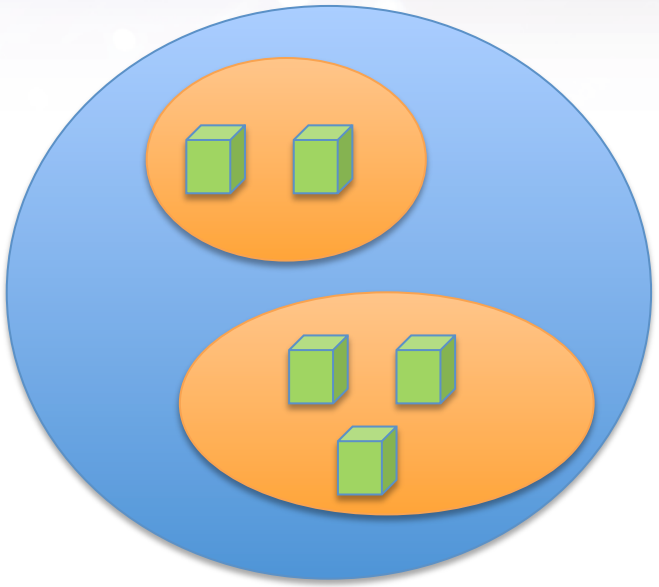
"You're going to need a bigger net."

- Promette
  - Aggregazione di risorse storage a basso costo
  - Ridondanze e scalabiltà “per definizione”

- Strumenti disponibili in Openstack
  - Swift Object Storage service
  - Keystone Identity service

Swift: modello dati gerarchico

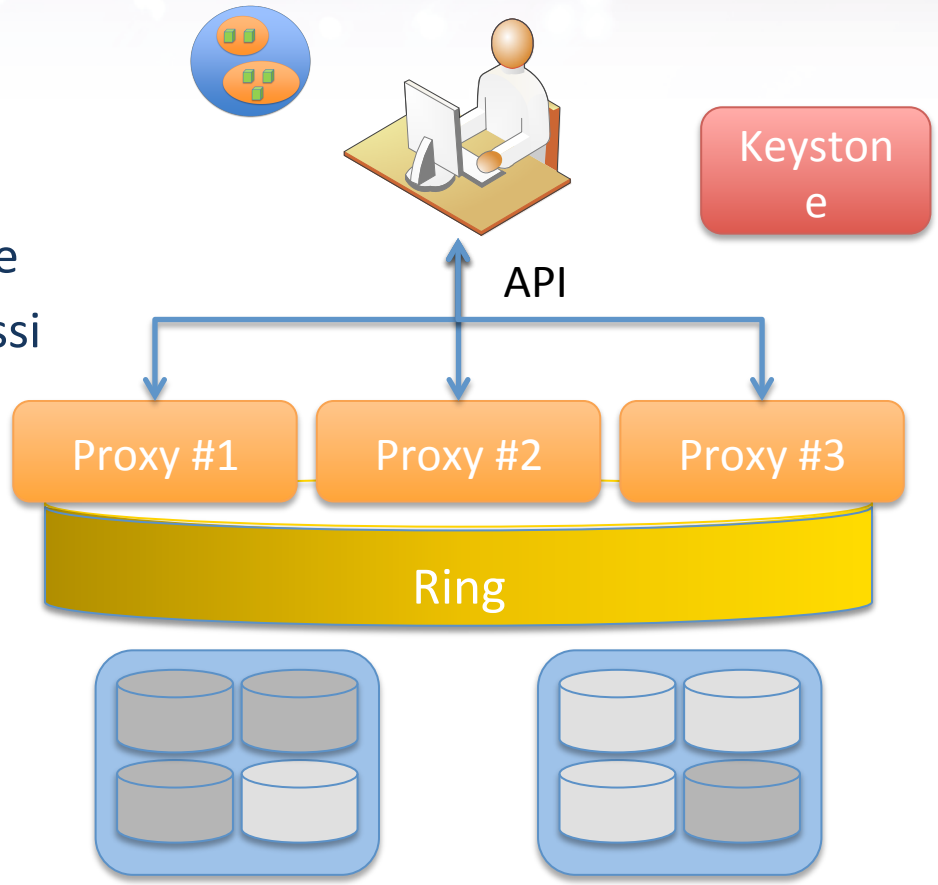
- Account
  - Namespace dei container
  - Corrispettivo “tenant”
- Container
  - Namespace degli oggetti
  - Quote, ACL, replica, policy



- Object
  - Contenuti binari: testi, documenti, immagini, dataset, …
  - URL univoca “<account>/<container>/<obj>”
  - API Create, Retreive, Update, Delete native e Amazon S3
  - Metadati aggiuntivi chiave/valore

Swift: Architettura

- Keystone Identity Service
- Proxy Node
  - Server
  - Mappature logico-fisiche
  - Gestione policy e processi
  - Recupero degli errori
- Object Node
  - Storage backend
  - Trasferimenti diretti
  - Repliche dati utente
  - Repliche database Ring

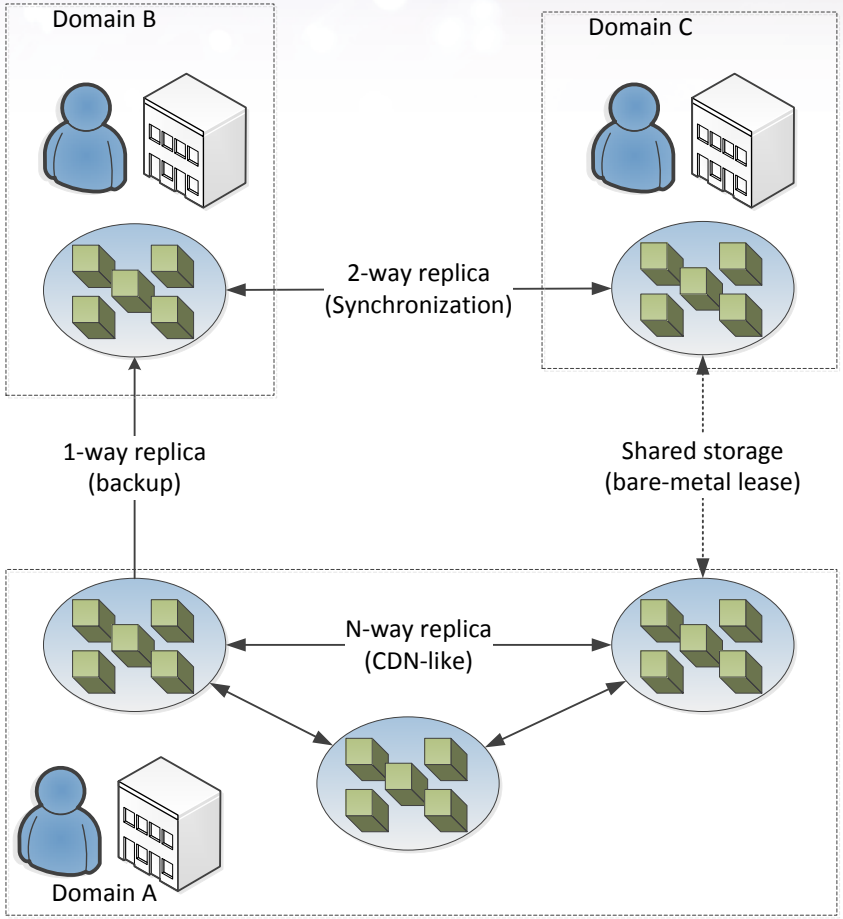


Casi d'uso

- Analisi dei requisiti: 6 attributi
  - Pattern sui dati tra istanze Swift
  - Autenticazione, autorizzazione e sicurezza
  - Attori: provider, utente, agenti software
  - …
- Scelti per la sperimentazione
  - Sincronizzazione bidirezionale con agenti Swift
  - Rassegna dei client per gestione e replica dei dati
  - Federazione di object storage con accesso AAI

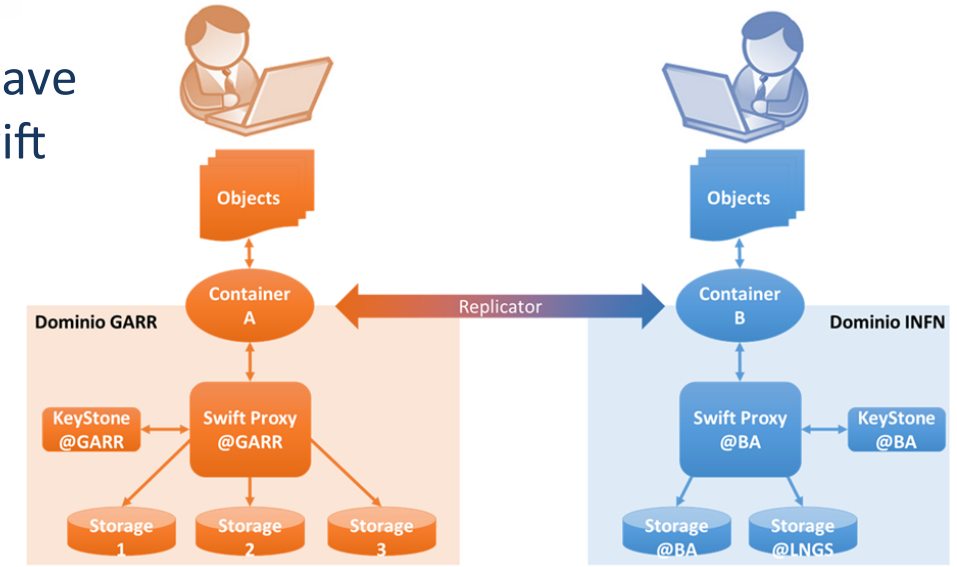
Piattaforma di test

- Openstack versione Kilo
- Dominio INFN – distribuito
  - Bari: Keystone, Proxy, Object server
  - LNGS: Object server distaccato
- Dominio GARR – due Swift
  - Bari: Keystone, Proxy e Object server
  - Cosenza: Keystone e Ceph – escluso poi



Sincronizzazione con Swift

- Abilitazione delle estensioni
- Creazione del trust
  - Provider condividono chiave
  - Configurazione reami Swift
  - Assegnazione dei ruoli
- Abilitazione
  - Selezione dei container
  - Aggiunta metadati per le repliche direzionali
- Esecuzione
  - Automatica e trasparente
  - Comuncazione diretta tra backend storage



Sincronizzazione con Swift

- Abilitazione estensioni sync

In /etc/swift/proxy-server.conf:
…
[pipeline:main]
pipeline = … container_sync …
[filter:container_sync]
use = egg:swift#container_sync
…

- Definizione del trust

In /etc/swift/container-sync-realms.conf:
[realm]
key = realm1key
key2 = realm1key2
cluster_clustername1 = https://host1/v1/
cluster_clustername2 = https://host2/v1/

- Abilitazione dei metadati

A → B
swift post -t '//realm_infn-garr/recas/<account_INFN>/<container1>' -k 'secret' <container1>

B → A
swift post -t '//realm_infn-garr/garr/<account_GARR>/<container2>' -k 'secret' <container2>

Risultati test di sincronizzazione

- Positivi: funziona!
- Parzialmente negativi
  - Iter di attivazione non immediato
  - Solo Swift “vanilla”, no alternative (Ceph)
- Punti aperti
  - Criteri di reattività non chiari
  - Efficienza? Throughput, latenze di sync e convergenza
  - Sicurezza del trust: segreto condiviso
  - Sicurezza del sync: comunicazione diretta tra backend

Rassegna dei client

- Profilo utente di riferimento
  - Amministratore di sistema
  - Esigenze di backup remoto o disaster recovery
  - Dati di backup sistema, dataset, snapshot VM
- Requisiti sugli strumenti
  - Operare ciclo completo del processo dati
  - OS multipli
  - Esperienze d'uso: GUI, CLI/scripting, device
  - Complessità di installazione configurazione
  - Licenza open/closed source, eventuali costi

Rassegna dei client

Applicazione	Utilizzo	Test	Sistema	Free
Cloudberry explorer	GUI	✓	Win	✓
Webdrive	Drive		Win. OSX	✓
Cyberduck	GUI	✓	Win. OSX	✓
S3 browser	GUI		Win	✓
Tntdrive	Drive		Win	✓
Rclone	CLI	✓	Win, OSX, Linux	✓
Duplicity	CLI	✓	Linux	✓
Dropshare	GUI	N/D	OSX	
Transmit 4	GUI	N/D	OSX	
Swift Client	CLI	✓	Win, OSX, Linux	✓
Curl	CLI	✓	Win, OSX, Linux	✓

Prossimi passi

- Completamento test di sincronizzazione
  - Consapevolezza sui punti aperti
  - Complessità di un servizio reale?
  - Operare Swift: stabilità e affidabilità sul lungo periodo
- Federazione object storage e AAI
  - Maggiore complessità
  - Estensioni Keystone multi-dominio, KS-Federation e supporto a AAI SAML
  - Sviluppo attivo in corso per i prossimi Openstack

Conclusioni

- Object storage
  - Paradigma con possibile grande impatto
  - Potenzialità degli strumenti Openstack
  - Validazione mirata a scenari reali
- Sinergia su interesse comune
  - Risultati dai primi 3 mesi
  - Premesse per ulteriori indagini
- Estensione della collaborazione
  - Coinvolgere altri partner
  - Da ripetere su altri temi

Grazie per l'attenzione