

# BlkKin: A Low-overhead tracing infrastructure for software-defined storage systems

Marios-Evangelos Kogias

National Technical University of Athens  
School of Electrical and Computer Engineering

October 9, 2014



# 1 Introduction

## 2 Motivation

## 3 Background

# Thesis Background

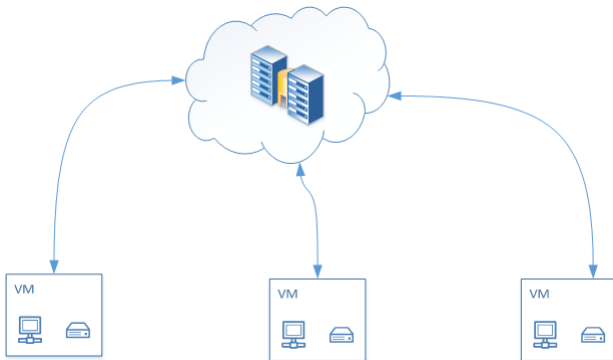


Open source, production-ready, cloud software.  
Designed since 2010 by GRNET.



- IaaS service
- Targeted at the Greek Academic and Research Community
- Designed by GRNET
- In production since 2011

# VM Volume storage

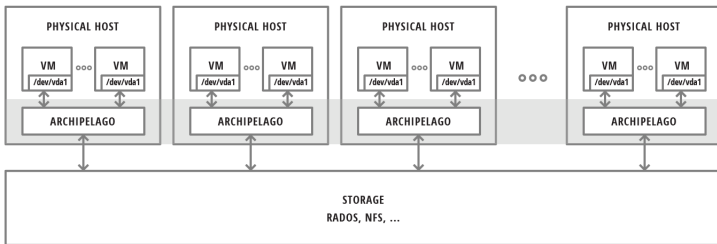


# Archipelago I

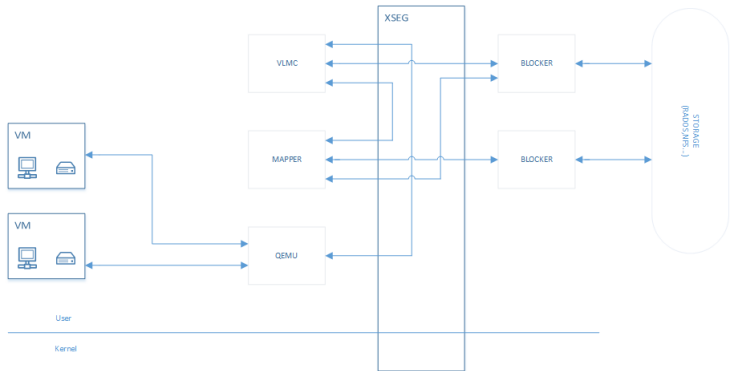
A thin distributed storage layer aiming to:

- Decouple storage logic from the actual data store
- Provide logic for thin cloning and snapshotting
- Provide logic for deduplication
- Provide different endpoint drivers to access Volumes and Files
- Provide backend drivers for different storage technologies

# Archipelago II



# Archipelago III



# RADOS

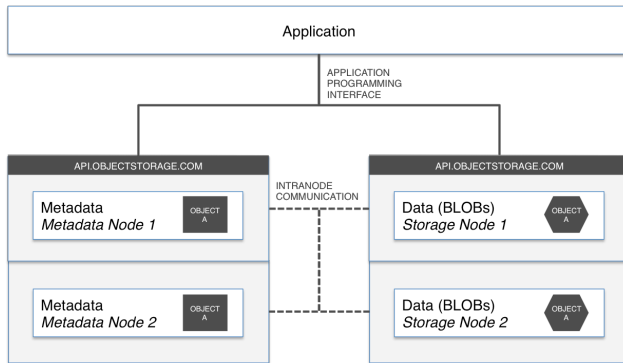
is the storage component of Ceph

RADOS basic characteristics are:

- *Replication*
- *Fault tolerance*
- *Self-management*
- *Scalability*



# Storage Abstraction



# The Problem

- Complex service oriented architectures
- Difficult debugging
- Difficult monitoring
- Non-deterministic execution
- Context-bound faults

# Solution

Distributed end-to-end tracing

&

Central data collection

# BlkKin

A distributed tracing infrastructure to track the IO request from Qemu until RADOS

BlkKin main characteristics:

- low-overhead tracing
- live-tracing
- End-to-end tracing of causal relationships
- User interface

# Main Challenges

- Meaningful and easily correlated tracing data
- Low overhead tracing backend

# Schools of thought

## black-box schemes

They assume there is no additional information other than the message record described above and use statistical regression techniques to infer that association.

## annotation-based schemes

They rely on applications or middleware to explicitly tag every record with a global identifier that links these message records back to the originating request.

# The Dapper System

- Large scale distributed systems tracing infrastructure created by Google
- Annotation-based tracing scheme
- Common libraries instrumentation
  - RPC System
  - Control Flow
- BigTable backend
- Closed-source

# Dapper tracing concepts

- annotation** The actual information being logged. Either *timestamp* or *key-value*
- span** The basic unit of the process tree. Can represent a subsystem or a function call. To depict causal relationship each span has a parent span or is a *root* span.
- trace** A different trace id is used to group data related to the same initial request



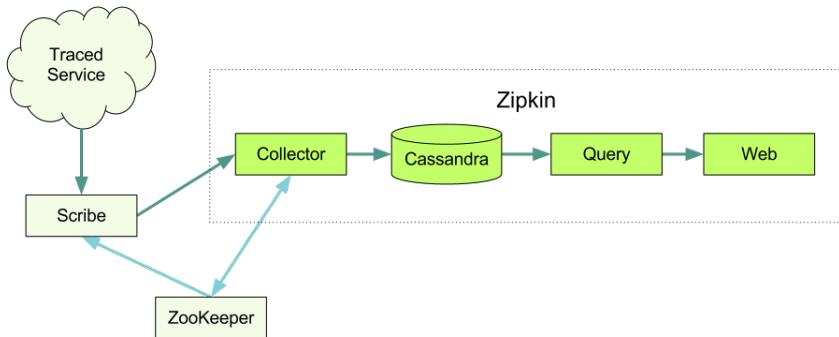
# Zipkin

An open-source Scala implementation of the Dapper paper by Twitter

Zipkin services:

- Data collector
- Database service
- Web UI

# Zipkin Architecture



# Scribe

Scribe is a scalable and reliable logging server created by Facebook

- Written in C++
- Directed graph architecture
- Batch messaging
- HDFS support
- Based on Apache Thrift

# Thrift

A software framework for scalable cross-language services development.

Includes a code generation engine to create RPC services across programming languages based on a Thrift file

Sample target languages: C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, OCaml

# Zipkin sum up

Zipkin is a full stack tracing system using

Scribe as its logging server using

Thrift as its transport protocol