TIERED FILE SYSTEM: OPTIMIZATION OF ARCHITECTURE AND MANAGEMENT ALGORITHMS

Master's Thesis Presentation

Author:
Sergey Morozov

Scientific supervisor:
Dr. Sc. (Phys.-Math.), Professor Vyacheslav Nesterov

Reviewer:
Andrey Pakhomov, Senior Solutions Manager at Dell EMC

June 22, 2017

Mathematics and Mechanics Faculty, St. Petersburg State University

Introduction

- Enterprise Storage
    - High Performance
    - Limited Capacity
    - Expensive
- Cloud Object Storage
    - Low Performance
    - Limitless Capacity
    - Cheap
- Automated Storage Tiering
    - Smart Data Migration Policies
    - “Hot” Data on High-Performance Tier (Expensive)
    - “Cold” Data on Low-Performance Tier (Cheap)

Problem Statement

Design and implement a file system-agnostic policy-based software component responsible for data synchronization between a POSIX-conformant file system and cloud object storage.

- Investigate automated storage tiering problems in an environment that includes a distributed file system and cloud object storage.
- Extract and compare important features of modern distributed file systems from the perspective of automated storage tiering.
- Design a software component that enables automated storage tiering between a POSIX-conformant file system and cloud object storage.
- Implement the designed software component and evaluate its performance.

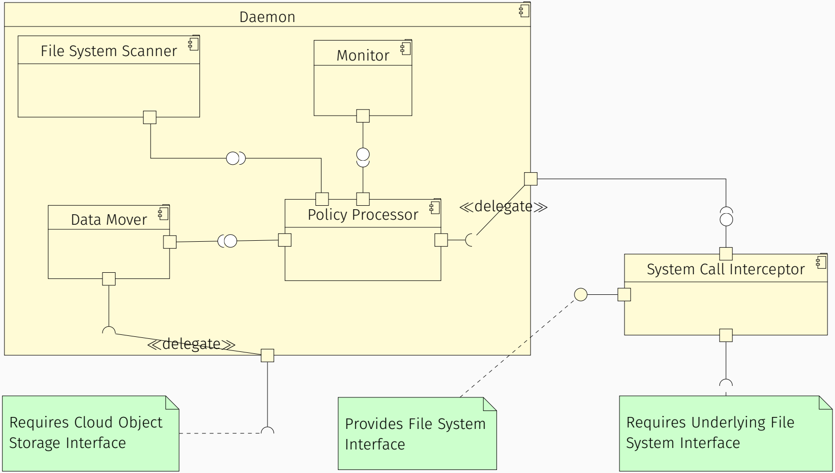
Automated Storage Tiering Problems

- Weak consistency model of cloud object storage
- Partitioning
- Data access pattern for a general case
- Metadata size
- Graphical file managers

Comparison of Some Distributed File Systems

	MooseFS	CephFS	GlusterFS	OrangeFS
POSIX Conformance	full	near	full	near
Tiering Support	yes	yes	yes	no
Fault Tolerance	yes	yes	yes	yes
License	GPLv2	LGPLv2.1	GPLv2 / LGPLv3	LGPLv2.1

Design: Component Diagram

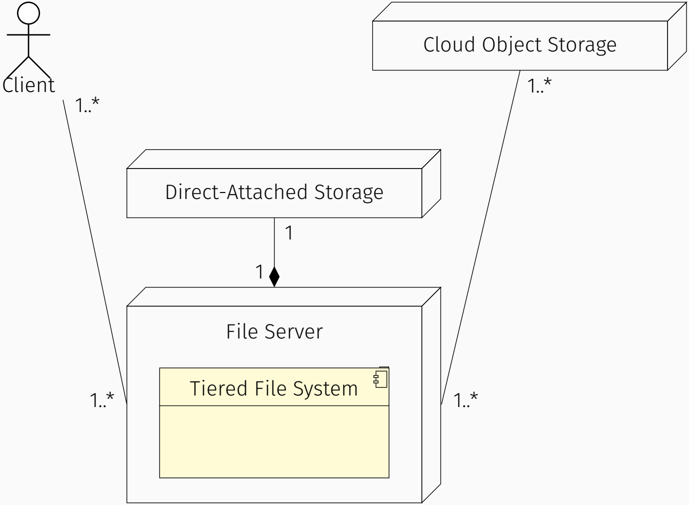


Requires Cloud Object Storage Interface

Provides File System Interface

Requires Underlying File System Interface

Design: Deployment Diagram



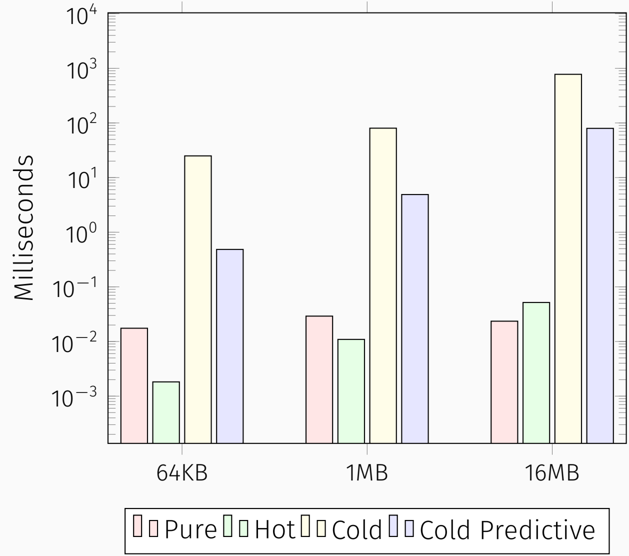
Implementation: Overview

Source Code:	`https://github.com/aoool/CloudTieringFS`
Language:	C
Libraries:	libs3, dotconf
License:	GNU General Public License v3.0

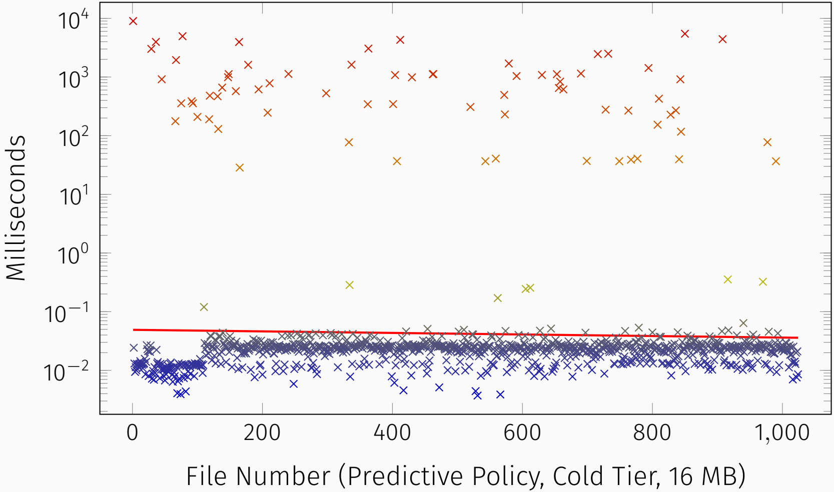
Implementation: Performance

- Compute nodes—Amazon EC2
  - SUSE Linux Enterprise Server 12 SP2
  - t2.micro (1 vCPU, 1 GiB RAM, 30 GB of Elastic Block Storage)
- Cloud object storage—Amazon S3
- Region—North Virginia
- Configurations
  - Single node—BtrFS
  - Multiple nodes—OrangeFS

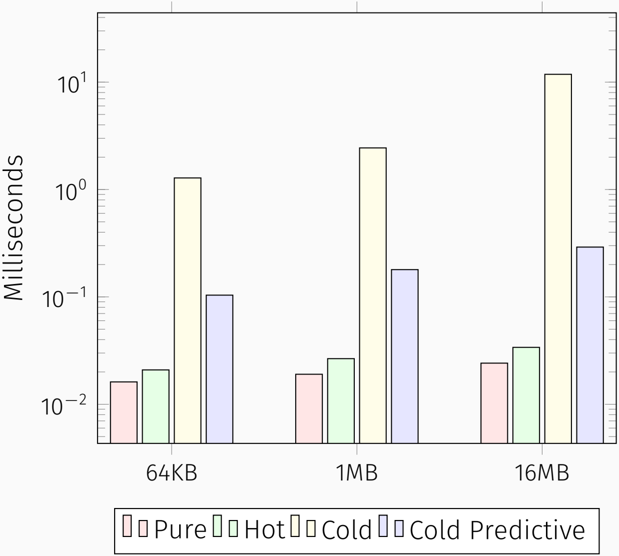
Implementation: Performance (BtrFS)



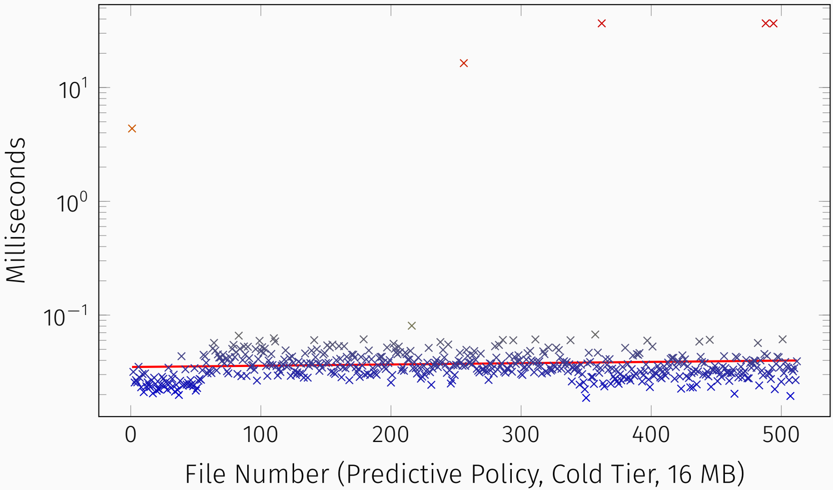
Implementation: Performance (BtrFS)



Implementation: Performance (OrangeFS)



Implementation: Performance (OrangeFS)



Results

Designed and implemented a file system-agnostic policy-based software component responsible for data synchronization between a POSIX-conformant file system and cloud object storage.

- Problems of automated storage tiering in an environment that includes a distributed file system and cloud object storage were investigated.
  [weak consistency, partitioning, metadata size, graphical file managers]
- Important features of modern distributed file systems related to the automated storage tiering were identified and compared.
  [MooseFS, CephFS, GlusterFS, OrangeFS]
- A software component enabling automated storage tiering between a POSIX-conformant file system and cloud object storage was designed.
  [file system-agnostic, distributed]
- The designed software component was implemented and its performance evaluated in single- and multi-node configurations.
  [BtrFS, OrangeFS]