CMPS229 Project Proposal: Horus Prototype (DRAFT)

Yan Li
yanli@ucsc.edu

January 22, 2012

1 Short version: The Elevator Pitch

(Elevator pitch is written in present perfect and/or past tense). I have implemented a prototype of Horus and evaluated it's performance. Horus enables on disk data encryption for HPC and lets you share only part of a file to a user, giving you fine grained access control. This problem is important because in scientific computing, sometimes you only want to share a port of file to a specific user. In the old ways, you have to extract that part of data, create a new file and share it to the user. Now with Ceph, you can create a range key for the user in O(1) (or O(log*n)?) and the user will be able to access that specific range of a file, regardless of the data size. We evaluated the performance of our Horus implementation and discovered ....

2 Clearly state the question to be solved

The question to be answered: how does Horus perform?

What will be covered:

- What is the performance overhead of Horus?
- There are several different ways of implementing Horus, and we want to compare their performances.
- Identify bottleneck and try to solve them, for example, we will try caching on KDS.
- We plan to implement Horus by using extattr, if this turned out to be a performance bottleneck, we will try some other alternatives.

What will not be covered:

- Horus on file system other than Ceph (what FS shall we use under Ceph? Ext4 should have the best metadata access performance so we should begin with Ext4)

- Horus on OSes other than Linux

2.1 Who will benefit when the problem is solved (e.g. CSP/DataMesh, HPL, IBM)?

HPC, Ceph. Horus can also be used in other SSRC projects like Cthulhu.

3 Why it is important? Why is it a problem?

Horus is currently only an idea on paper without implementation or performance evaluation, which must be addressed before it can be taken as a plausible design.

4 What is my approach?

What is the basic approach, method, idea or tool thats being suggested to solve the problem?

We will try to Implement the Horus protocol on Ceph in two methods: as a user-land library, and as a FUSE file system. Then we will do performance evaluation of it.

5 Hypotheses

What exactly are the expected effects of the proposed solution? (E.g. disk I/O time will increase to 2 seconds per request.) Why is this?

There should be only a little O(1) overhead for each open file request from client, therefore the overhead for reading one large file should be negligible. The most interesting part will be evaluating the performance impact when reading a lot of small files.

Since in Horus communication protocol there's no large amount of data exchange, we speculate that the network latency will be the decisive factor for performance.

For KDC, we only expect moderate CPU usage because computing the range key isn't very complex. But we will implement caching on KDC to minimize it's communication with MDS.

Do we need a multi-threaded implementation of the KDC server?

6 Collaborator

Yasu and Nakul will work with me on this project. Yasu and Nakul has already built the framework of the code and implemented the FUSE and user-land library parts. I will focus on implementing the encryption code, running experiments, result analysis and performance tuning.

7 Experiments

What will be done to test out the hypotheses? (E.g. measurements, simulations, constructing code, thinking beautiful thoughts, hard vacationing). How will this confirm (or deny) the hypotheses? Why will the conclusions be believable?

Since Horus will mainly be used in HPC, we'd like to use some file system benchmarks that reflect scientific computing workload.

We will do the following experiments:

- benchmark using SOE servers and network
- optionally introduce a 1-second network latency and do benchmark again
- multi-client stress benchmark to KDC
- multi-MDS if MDS turned out to be bottleneck (unlikely)

7.1 Device/Equipment Needed

We will begin with 3 servers and SOE network. Will need some more servers for stress testing.

8 Results

What will be the outcome of the work (papers, a working system, a graph of ...)? When?

A paper and a prototype system that can be finished within the Winter 2012 quarter.

What are the measures for success? (E.g. faster, smaller, more available.) How will we know to declare the project a success?

A prototype system should be built, experiments run and results analyzed and published.

9 Plan

The Winter 2012 ends on Mar 16. We have 8 weeks from now on (Jan 21st). Week 1 is Jan 22 to Jan 28. All milestones are to be checked at the Friday of that week.

Week	Milestones
0	Project begins.
2 (Feb 3)	Finish implementation.
4 (Feb 17)	Finish experiments.
6 (Mar 2)	Result analysis and re-run experiments if needed.
8 (Mar 16)	Finish paper and poster.