#　小组会记录
#　20071010
#　author: pb

参加：yzf,pb,fankai,tqc,tsunami,zhulei

1。项目远景

Why do we do this?

- Build an infrastructure to support large-scale web data processing, mining researches.
- What to do with the huge volume web data on this infrastructure, which could be a more interesting problem.

2。本周工作安排：

tfs系统目前集中在功能测试，和bug修正，期望能在10月底release一个可基本稳定运行的“可用”版本。

本周工作集中在client库和datatransfer库测试，以能保证正确工作。大家都帮助阅读这部分的代码。

fankai:
客户库的基本模块测试
tqc:
传输模块的基本模块测试
zhulei:
检查client library和datatransfer库的代码

通过测试testClient

tsunami:

继续 shell 实现

yzf:

master 端的目录浏览

pb :

考虑 mapred

3。研究工作准备

在本学期末争取在写作上有进展。

fankai : 毕业论文
tsunami : 研究报告论文
tqc : 论文
yzf : 毕业论文

阅读 Amazon's Dynamo
http://www.allthingsdistributed.com/2007/10/amazons_dynamo.html

TianwangFileSystem

2007-10-10

Yzf,tqc,fankai,tsunami,zhulei,pb@sewm

Why do we do this?

- Build an infrastructure to support large-scale web data processing, mining researches.
- What to do with the huge volume web data on this infrastructure, which could be a more interesting problem.

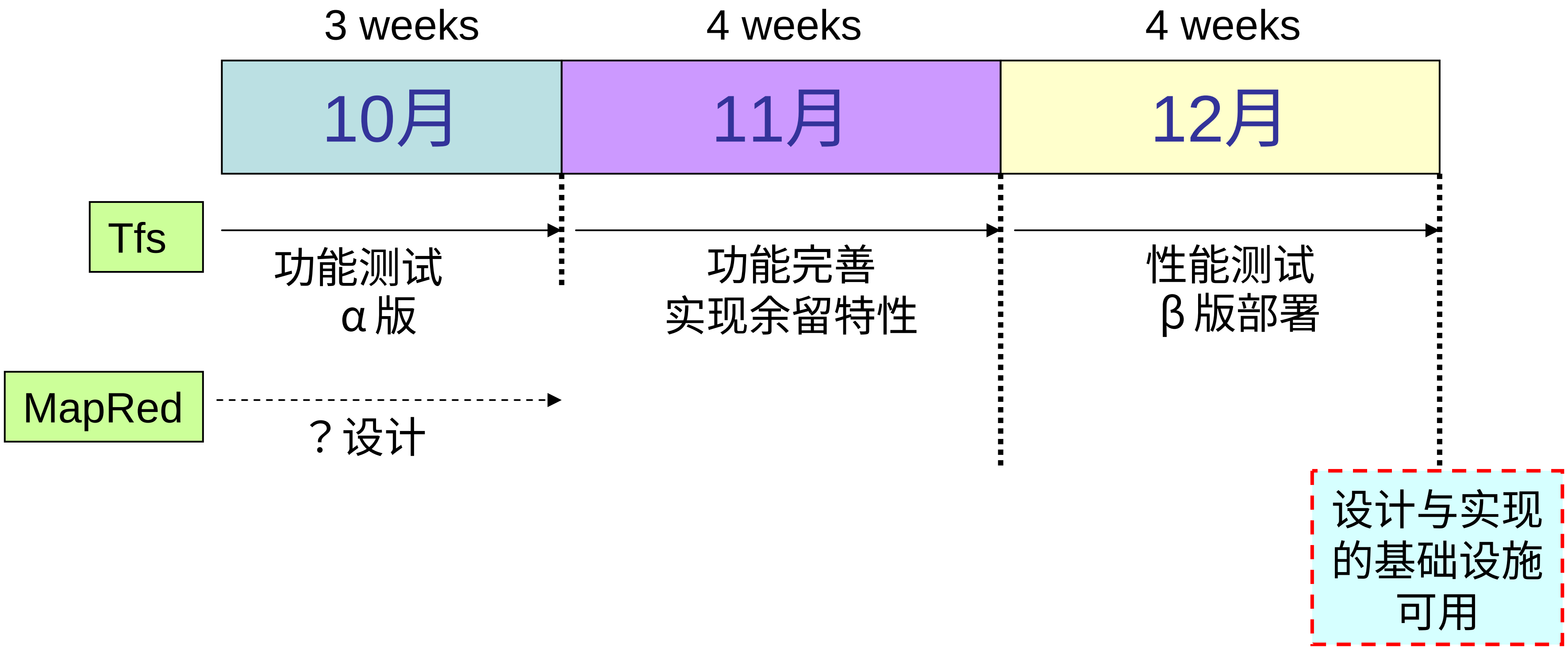
Related Works

- Hadoop
  - Release 0.13.0 June, 2007
  - Used in Google & IBM cloud computing initiative for university (Oct,8 , 2007)

Related Works

- KosmosFS (Sep 27 , 2007)
  - atomic record append
  - multipleconcurrent writes to a file
  - rebalancing
  - exports a POSIX file interface
  - integrated with Hadoopusing Hadoop's filesystem interfaces (see Hadoop-Jira-1963)
  - Fuse binding

Our TimeLine



3 weeks

4 weeks

4 weeks

10月

11月

12月

Tfs

功能测试
α 版

功能完善
实现余留特性

性能测试
β 版部署

MapRed

？设计

设计与实现
的基础设施
可用

Research

- What to do?
- Read this
  - **Amazon's Dynamo**
  - In two weeks, will present a paper on the Dynamo technology at SOSP
  - http://www.allthingsdistributed.com/2007/10/amazons_dynamo.html

http://www.mail-archive.com/hadoop-user@lucene.apache.org/msg02362.html

==================

KFS implemented in C++. It also contains support for Java/Python applications.

In a nutshell,
 - KFS supports file replication, that is configurable on a per-file basis
 - Chunks of a file are replicated, typically, 3-way. This is used to provide data availability during chunkserver outages.
 - Re-replication is used to recover chunks that were lost due to extended chunkserver outages.
 - For data integrity, KFS stores check sums on data blocks, which are verified on read; if corrruption is detected, re-replication is used to recover the corrupted data
 - KFS supports incremental scalability; new storage nodes can be added to the system
 - To enable better disk utilization, KFS metaserver may periodically rebalance the chunks by migrating chunks from "over-utilized" servers to "under-utilized" servers.
 - KFS exports a standard filesystem API (such as, create, read, write, etc.). Files can be written to multiple times; KFS supports append operation on files.

To enable applications to use KFS, KFS has been integrated with Hadoop using Hadoop's filesystem interfaces (see Hadoop-Jira-1963). This enables existing Hadoop applications to use KFS seamlessly.

=================

Looks like the big differences are that KFS is a more complete clone of GFS:

* KFS supports atomic append, Hadoop does not
* KFS supports rebalancing, Hadoop does not
* KFS exports a POSIX file interface, Hadoop does not (GFS does not, either)

Frankly, I find the competition good for Hadoop. However, KFS is not really attractive until they have a MapReduce-alike. You have to be able to do something with all that data, after all ;-) Maybe KFS can be integrated with Hadoop's MapReduce to make up for the current lack of such from Kosmix?

Hi Ian,

>
> > what is the fundamental difference between KFS and hadoop such that
2
> > seperate projects are required?

Backing up a bit,
- Hadoop: is map-reduce engine + HDFS
- KFS: is a filesystem

Your question is probably more of what is the difference between KFS and HDFS.

Toby's reply to your mail gives me a bit more credit :-)

>
> * KFS supports atomic append, Hadoop does not
>

KFS, currently, DOES NOT support atomic record append. We have designed the system for atomic record append; we have held off this feature as others have taken priority. There is support for multiple concurrent writes to a file---the chunkserver that has the write lease serializes the writes.

> * KFS supports rebalancing, Hadoop does not

Currently this is the case. Owen had replied to one of the posts over the weekend/last week about adding block rebalancing in HDFS in an upcoming Hadoop release.

> * KFS exports a POSIX file interface, Hadoop does not (GFS does not, either)

To be a bit clear,
- with HDFS, you open a file for writing once; you write sequentially start->end
- with KFS, you can open a file for writing multiple times; you can seek anywhere and write; the KfsClient::Open() supports O_APPEND in POSIX-style appends; not record-append.

There is also the issue with HDFS's writes to a file becoming visible on close. With KFS,

- when a client creates a file, the file is part of the fs namespace
- writes are cached at the client
- whenever the cache is full/application calls flush, the data gets pushed out to the chunkservers
- when data is written to chunkservers, it is visible

> Maybe KFS can
> be integrated with Hadoop's MapReduce to make up for the current lack
> of such from Kosmix?

Toby---This *is* done. I have a filed a JIRA issue and submitted the code. See Hadoop-1963.

Sriram

> One thing I wasn't able to get from your website is if KFS is able to
> deal with metadata server failures
> (being a single point of failures in many DFS implementations).

Currently, metadata server is a single-point of failure. That said, we intend to make it a bit more robust (such as, adding remote logging and remote copy of checkpoints).

http://highscalability.com/you-can-now-store-all-your-stuff-your-own-google-file-system
http://www.danga.com/mogilefs/
MogileFS
MogileFS 是一套高效的文件自动备份组件，由 Six Apart 开发，广泛应用在包括 LiveJournal 等 Web2.0 站点上

---

http://highscalability.com/google-architecture

- Video: Building Large Systems at Google
- Google Lab: The Google File System
- Google Lab: MapReduce: Simplified Data Processing on Large Clusters
- Google Lab: BigTable.
- Video: BigTable: A Distributed Structured Storage System.
- Google Lab: The Chubby Lock Service for Loosely-Coupled Distributed Systems.
- How Google Works by David Carr in Baseline Magazine.
- Google Lab: Interpreting the Data: Parallel Analysis with Sawzall.
- Dare Obasonjo's Notes on the scalability conference.

FUSE binding
Filesystem in Userspace
http://fuse.sourceforge.net/

---

hadoop user mailinglist
http://www.mail-archive.com/hadoop-user@lucene.apache.org/

Google and IBM team on cloud computing initiative for universities

the goal is to improve students' knowledge of parallel computing practices and better prepare them for increasingly popular large-scale computing that takes place in the "real world," such as search engines, social networking sites, and scientific computational needs.

“We in academia and the government labs have not kept up with the times,” said Randal E. Bryant, dean of the computer science school at Carnegie Mellon University. “Universities really need to get on board.”

Just posting a link to this paper because I hadn't seen it mentioned here and thought you guys would find it interesting.

http://www.allthingsdistributed.com/2007/10/amazons_dynamo.htm

"Dynamo is internal technology developed at Amazon to address the need for an incrementally scalable, highly-available key-value storage system. The technology is designed to give its users the ability to trade-off cost, consistency, durability and performance, while maintaining high-availability.
...
We submitted the technology for publication in SOSP because many of the techniques used in Dynamo originate in the operating systems and distributed systems research of the past years; DHTs, consistent hashing, versioning, vector clocks, quorum, anti-entropy based recovery, etc. As far as I know Dynamo is the first production system to use the synthesis of all these techniques, and there are quite a few lessons learned from doing so. The paper is mainly about these lessons."

SOSP
OSDI