BulkFS - a Distributed Fault-Tolerant File System for Massive Data Applications

Antti Tuominen & Ville H. Tuulos

Complex Systems Computation Group

Helsinki Institute for Information Technology

{attuomin,tuulos}@cs.helsinki.fi

http://www.cs.helsinki.fi/u/attuomin/bulkfs/

Outline

- our computing environment and needs
- BulkFS design principles
- BulkFS components
- application’s point of view
- performance

A cluster using regular PC hardware

- cheap
- more prone to hardware failures
- no central storage

What is needed from a file system

- a convenient way of using all those separate disks
- storing large amounts of data for batch processing
- some protection against hardware failures
- speed and scalability

What isn’t needed from a file system

- directory hierarchies
- handling of small files
- locking
- permissions

Design principles

- maximize simplicity
- distribute all heavy lifting to avoid perfomance bottlenecks
- store metadata along with data for full reconstruction
- if something is easier to do outside BulkFS, do it there

BulkFS components

IOserf provides reading and writing over network to a single file/partition

one per each *volume*

bookkeeper manages metadata

one per each BulkFS

client library provides a simple API to applications, talks to bookkeeper and IOserfs

one per each application

Application's point of view

Using BulkFS is quite straight forward

write_block(directory, block name, block data, redundancy level)

read_block(directory, block name, block data)

Performance - benchmarking setup

- One server running the bookkeeper.
- Five nodes with two volumes each, one raw 250G disk and one 100G file under ReiserFS.
- All connected with gigabit ethernet.
- BulkFS figures are from a stress test program which reads/writes random blocks from a data set.
- Stress test clients are run on the same nodes as IOserfs - one operation out of five doesn't need to transfer data over the network.

Performance - data points for reference

hdparm: 61 MB/sec

Local read: `dd if=/dev/sdb1 of=/dev/null bs=128M`
51 MB/sec

`Read over network:  dd if=/dev/sdb1 bs=1M | nc other_node | (socket) | nc >/dev/null`
50 MB/sec

- dd doesn't seem to do simultaneous reading and writing

Performance - BulkFS & NFS

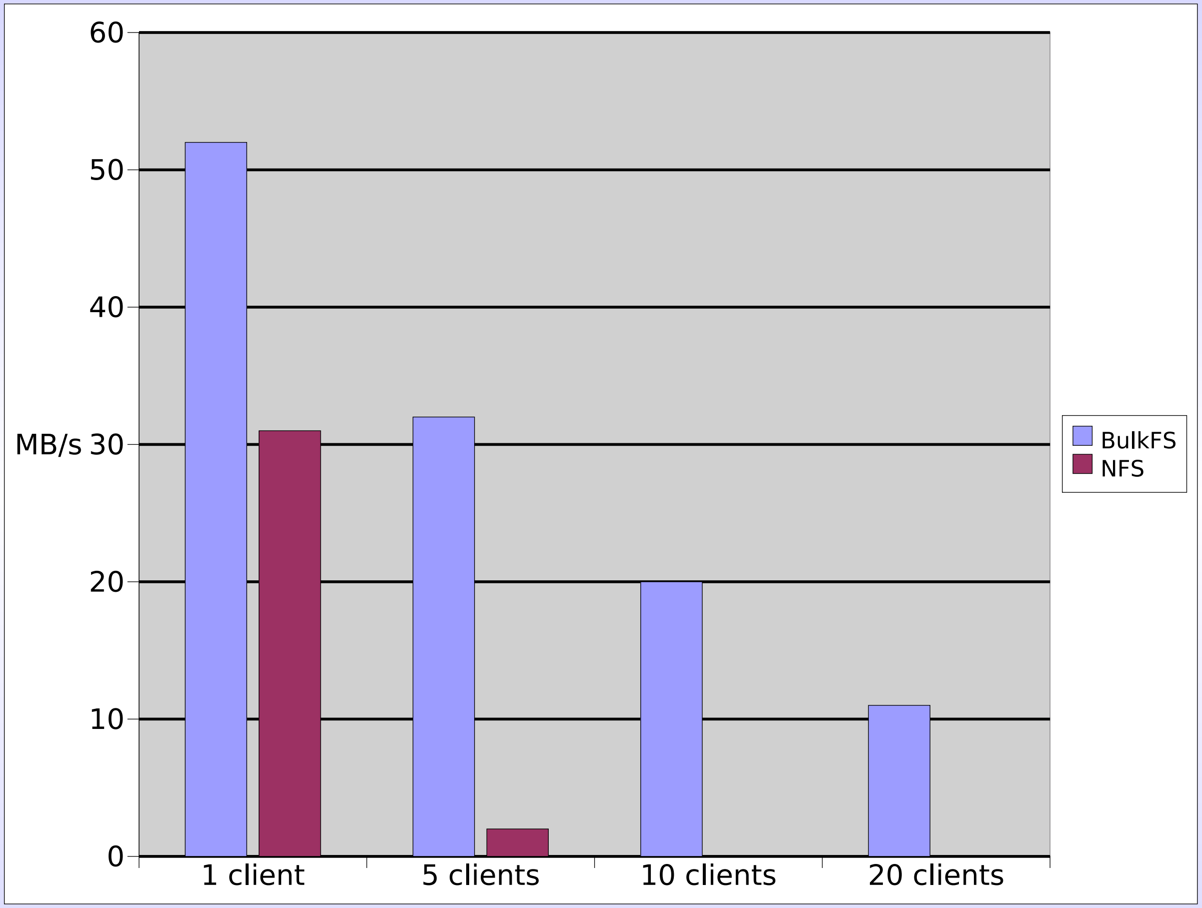

Figure 1: Transfer speed per client

Finally...

- Looks promising but hasn't been used much yet in real applications.
- All feedback is highly appreciated.

BulkFS is available at

http://www.cs.helsinki.fi/u/attuomin/bulkfs/