LUFS-GfarmFS File System Module for Grid Datafarm System

Itthichok Jangjaimon

Putchong Uthayopas

High Performance Computing and Networking Center

Kasetsart University

Agenda

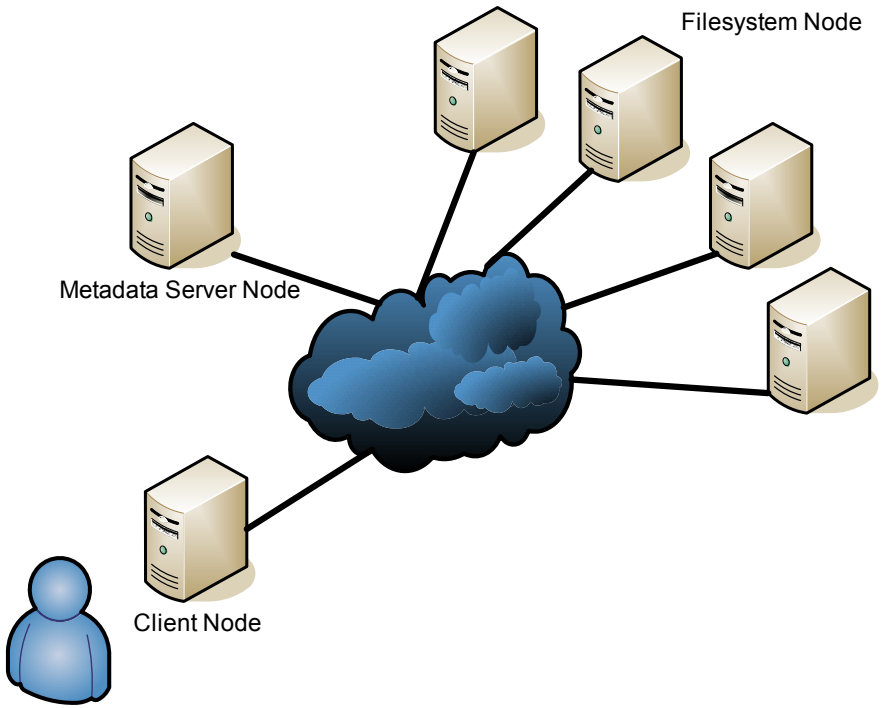
- Introduction
- Proposed Techniques
- Result
- Experiments
- Conclusion

Introduction

- Many large-scale scientific research projects are data intensive applications
- DataGrid is built to facilitate that demand
- There are 2 kinds of data grid
  - Replication based data grid
  - Parallel File System based Data Grid
- Grid Datafarm is one of the widely used File system based data grid software

Introduction (Cont.)

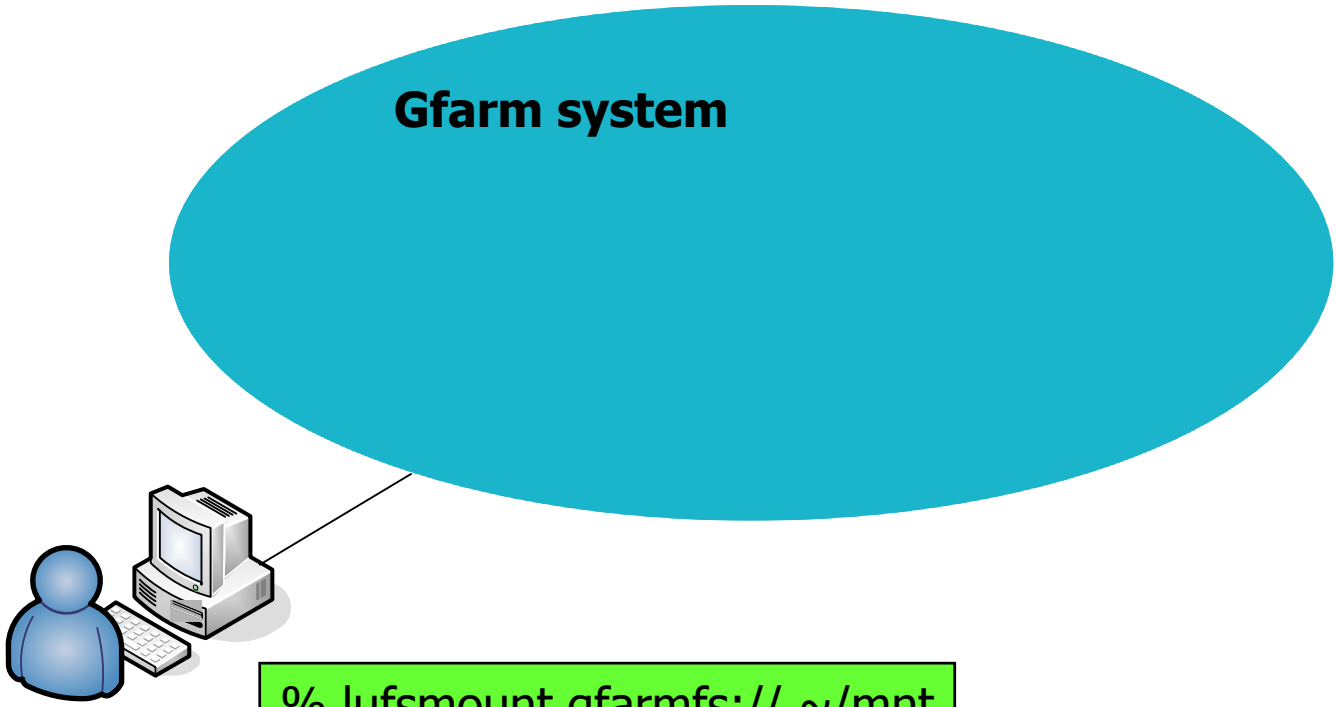
- Gfarm Architecture
  - Filesystem nodes
  - Meta data server
  - Client node
- Problem
  - Gfarm data access can only be done through API and command line
  - Difficult to use



Motivation

- A support for seamless integration with operating system can substantially simplify user task of accessing Gfarm data
- Very useful for non performance critical tasks
  - Data staging, application staging
  - Result checking
- This work is a development of GfarmFS- Virtual File System Module for Grid Datafarm System

Goal

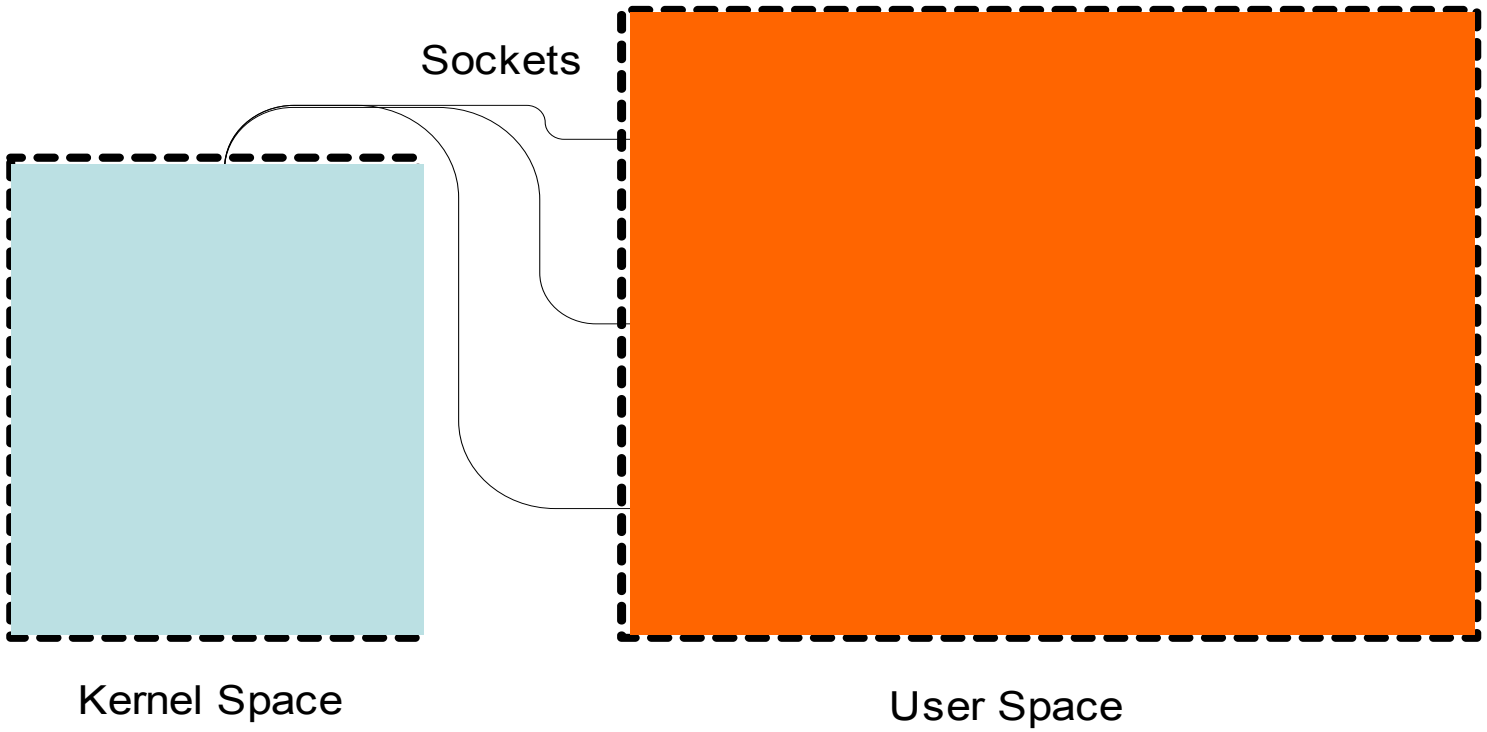


Proposed Techniques

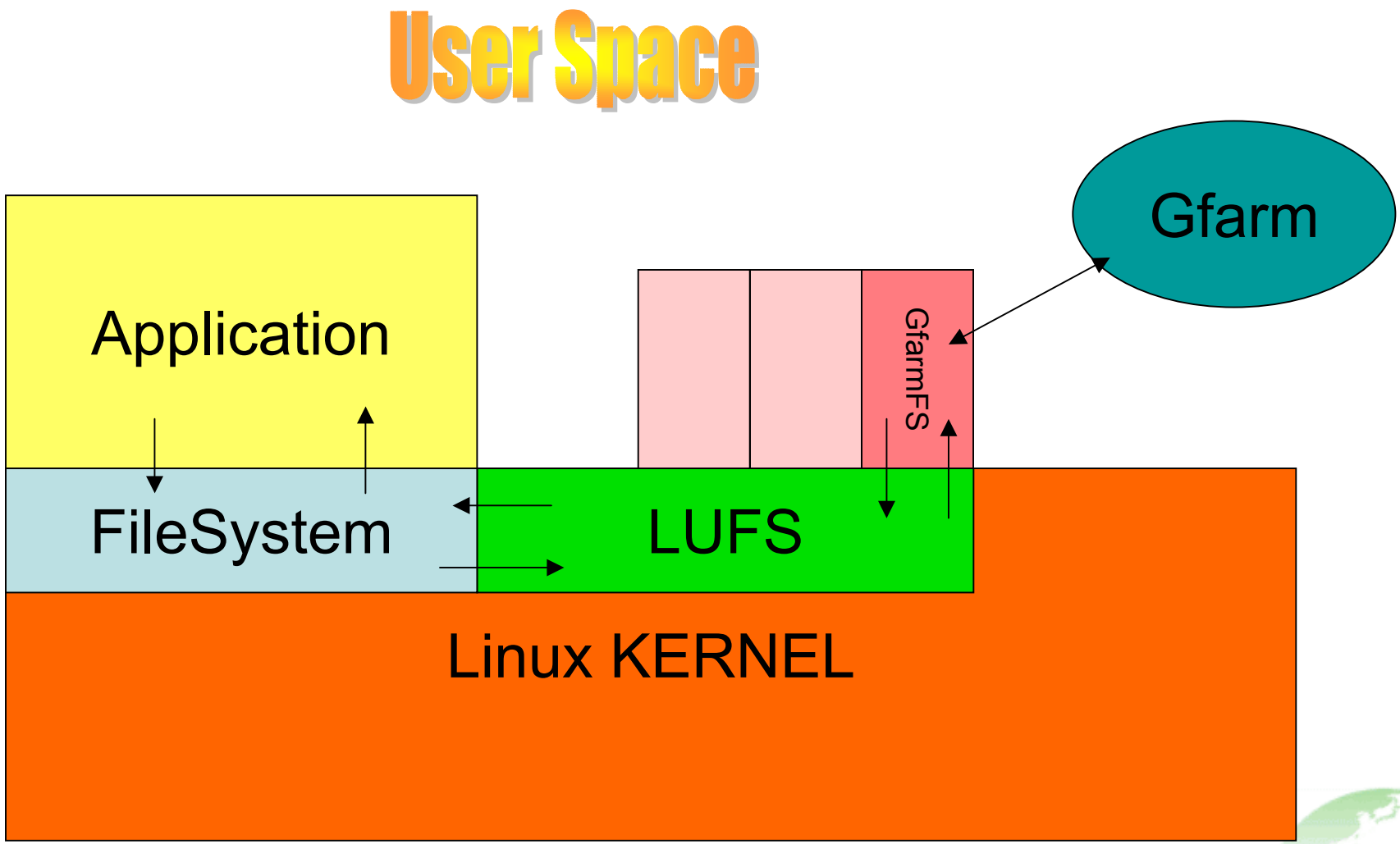
- 1st method : Implement VFS module in the kernel environment
  - Implement libraries into kernel
  - Fast but difficult to maintain
- 2nd method : Split module into 2 parts
  - One in kernel space
  - One in user space
  - Slower but more flexible

Proposed Techniques - LUFS

- LUFS = Linux Userland File System
- Hybrid userspace file system framework
- Support undefinite number of filesystems



GfarmFS Operation



GfarmFS implementation

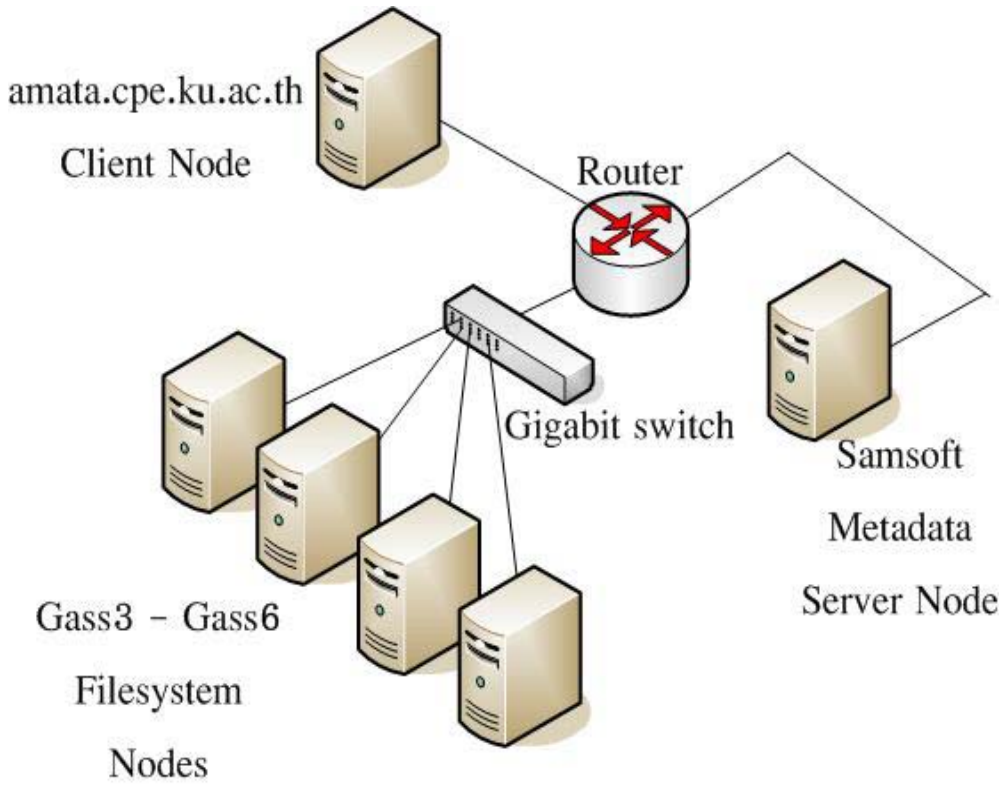
- Implement methods required by LUFS
  - Open, close , read, write, stat
  - Directory access, symbolic link
- Implement Gfarm Interface
  - Calling Gfarm API
  - Caching

Challenges

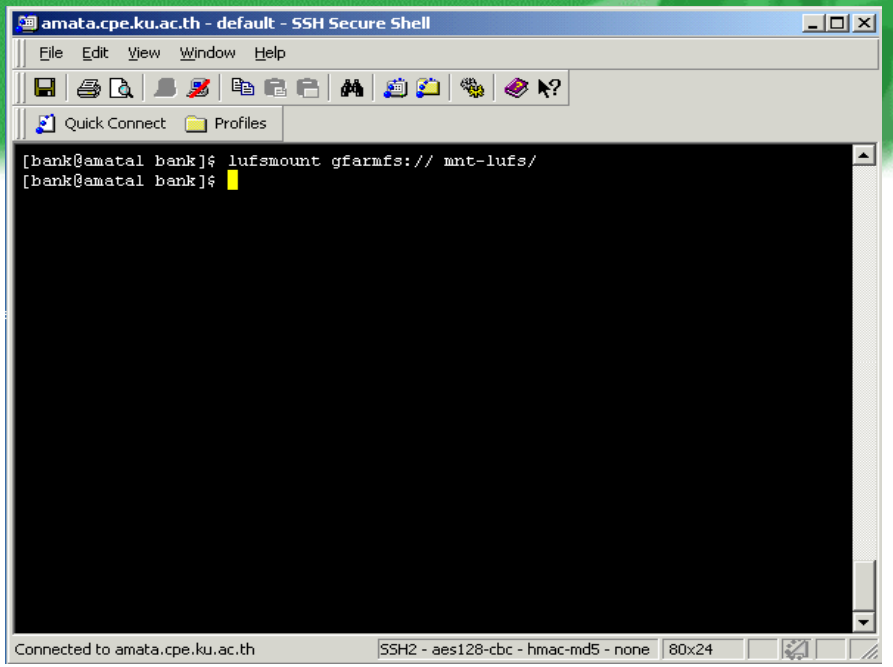
- LUFS does not support file descriptor (just give you a filename, buf , offset and length when open, read, write, close)
- Gfarm does not support
  - Writing to file with append mode
  - Opening file with R/W mode (LUFS required)
  - Writing to an existence file in the system
  - Require knowing the number of file fragment at the beginning
- When write, first write to temporary file and then dump it to Gfarm
- Fix file fragment size (1 MB now) when write (for experimental)

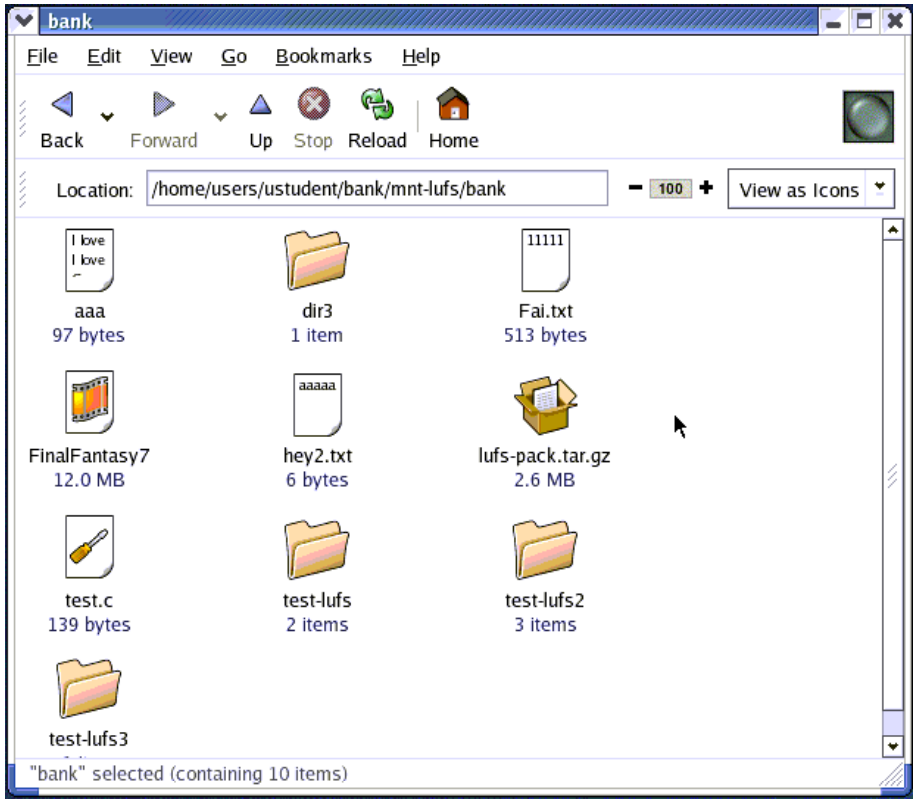
Experiments - Setup

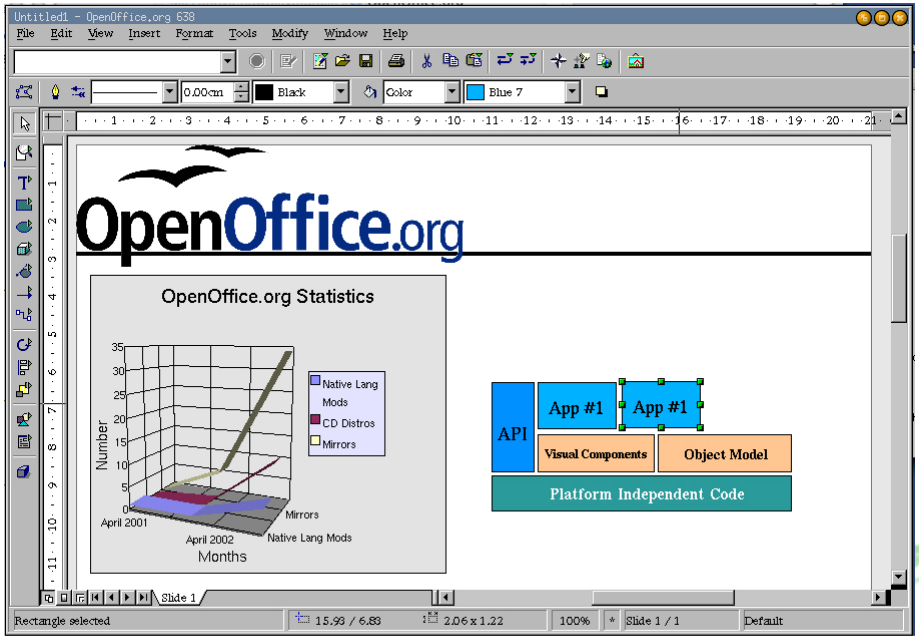
- Client Node (AMATA - Athlon 1 GB, 512 MB ram)
- Metadata Server Node (Samsoft – Pentium III 900 MHz, 2 GB ram)
- Filesystem Nodes (GASS3-GASS6 – AthlonMP 1800+ dual processor, 1GB ram)
- Link with 3Com Gigabit switch



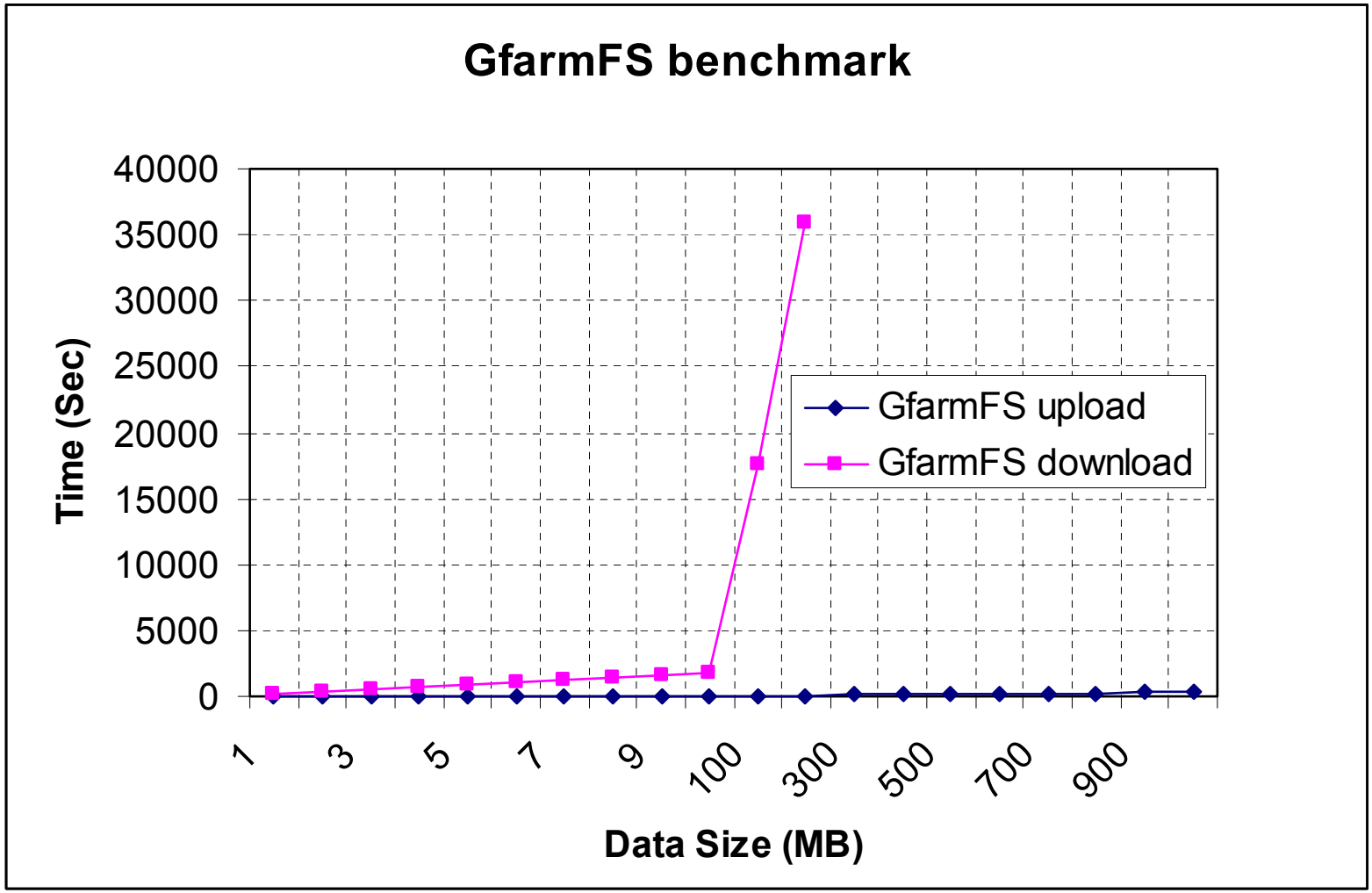
Result (Cont.)


amata.cpe.ku.ac.th - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
[bank@amata1 bank]$ lufsmount gfarmfs:// mnt-lufs/
[bank@amata1 bank]$
Connected to amata.cpe.ku.ac.th
SSH2 - aes128-cbc - hmac-md5 - none
80x24

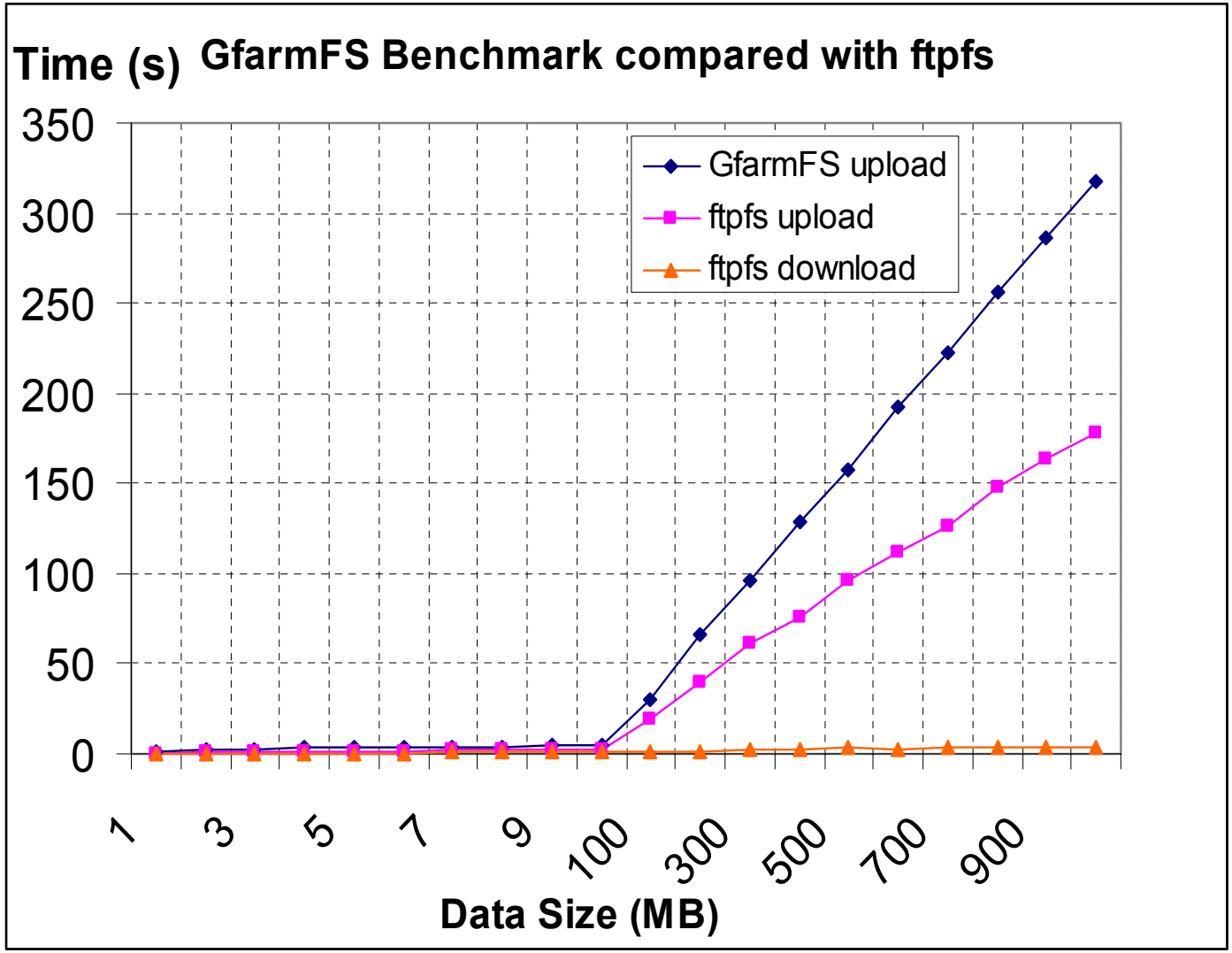

bank
File Edit View Go Bookmarks Help
Back Forward Up Stop Reload Home
Location: /home/users/ustudent/bank/mnt-lufs/bank
100
View as Icons
aaa
97 bytes
dir3
1 item
Fai.txt
513 bytes
FinalFantasy7
12.0 MB
hey2.txt
6 bytes
lufs-pack.tar.gz
2.6 MB
test.c
139 bytes
test-lufs
2 items
test-lufs2
3 items
test-lufs3
"bank" selected (containing 10 items)


OpenOffice.org
OpenOffice.org Statistics
Number
Months
April 2001
April 2002
Native Lang Mods
CD Distros
Mirrors
API
App #1
App #1
Visual Components
Object Model
Platform Independent Code

Result


GfarmFS benchmark
Time (Sec)
40000
35000
30000
25000
20000
15000
10000
5000
0
1
3
5
7
9
100
300
500
700
900
Data Size (MB)
GfarmFS upload
GfarmFS download

Result (Cont.)


Time (s)
GfarmFS Benchmark compared with ftpfs
350
300
250
200
150
100
50
0
GfarmFS upload
ftpfs upload
ftpfs download
1
3
5
7
9
100
300
500
700
900
Data Size (MB)

Conclusion

- Provide more familiar approach to access Gfarm file system
  - User can freely use any file manager or tools to manipulate Gfarm file system data
- Performance is less than other file system module, such as FTPFS
- Potentially speed up the work of scientists who need a simple way to manage huge and scattered data set

Thank you for your attention

HPCNC