Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

Using Lustre on Juropa

An Introduction

21 May 2012 | Frank Heckes

Topics

- ***Lustre Overview***
- ***Lustre Infrastructure on Juropa***
- ***Storage Concepts***
- ***Usage Recommendations***
- ***Summary***
- ***Documentation & Links***

Lustre Overview

- ***Why parallel I/O***
  - *Disks are slow*

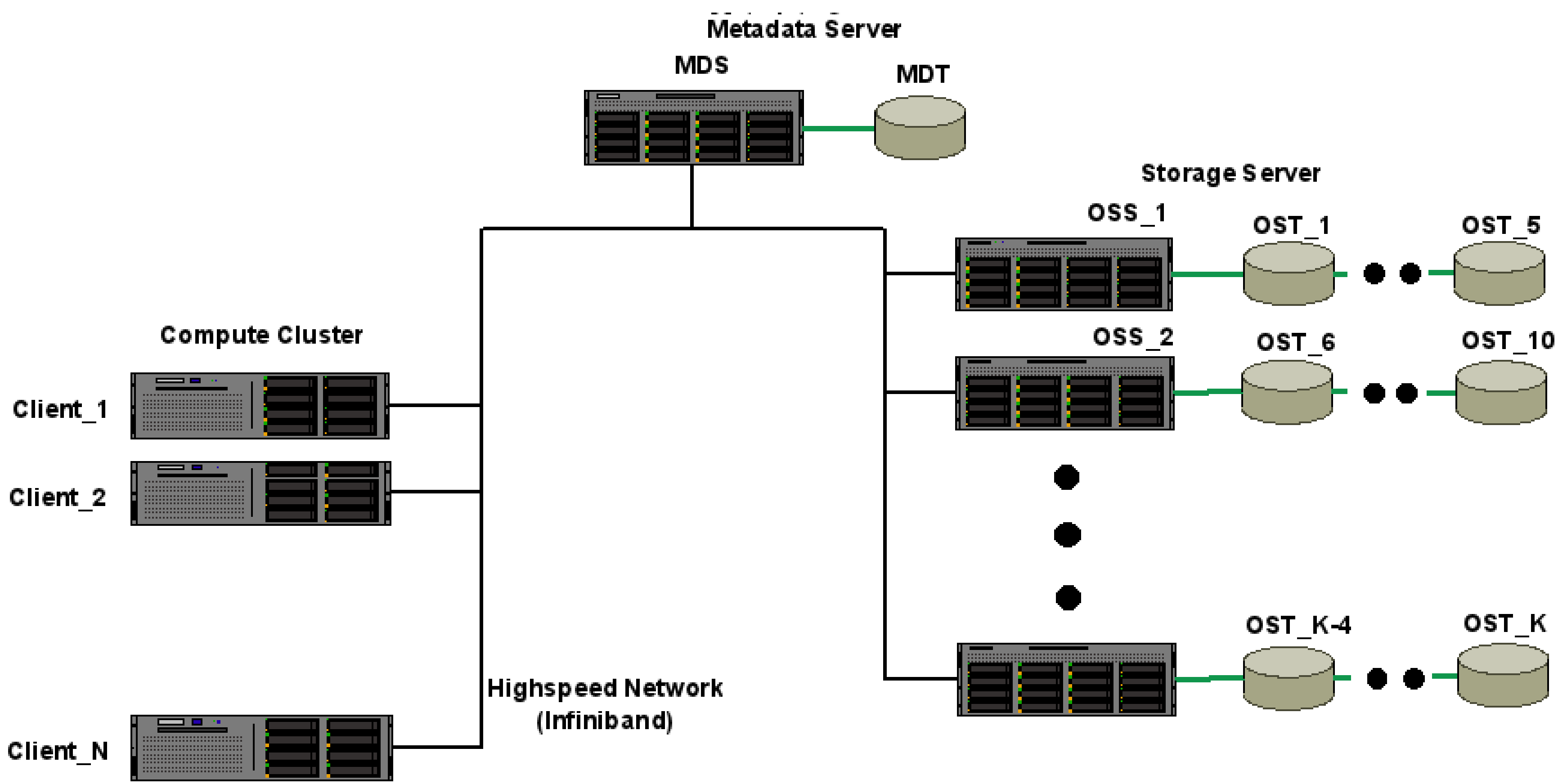
	REAL		SCALED	
Component	time	unit	time	unit
CPU cycle	0.31	ns	0.31	s
L2 cache	1.25	ns	1.25	s
DRAM chip	60.0	ns	1	min
Disk seek	3.5	ms	1.35	month
Tape access	5	s	1.59	century

  - *Can't increase speed → do as much in parallel as possible*
  - *need for huge capacities, throughput, common name space*
- ***Problem is addressed by parallel file systems***
  - *Lustre, GPFS*
  - *Other: Panasas, PVFS, pNFS, FhGFS*

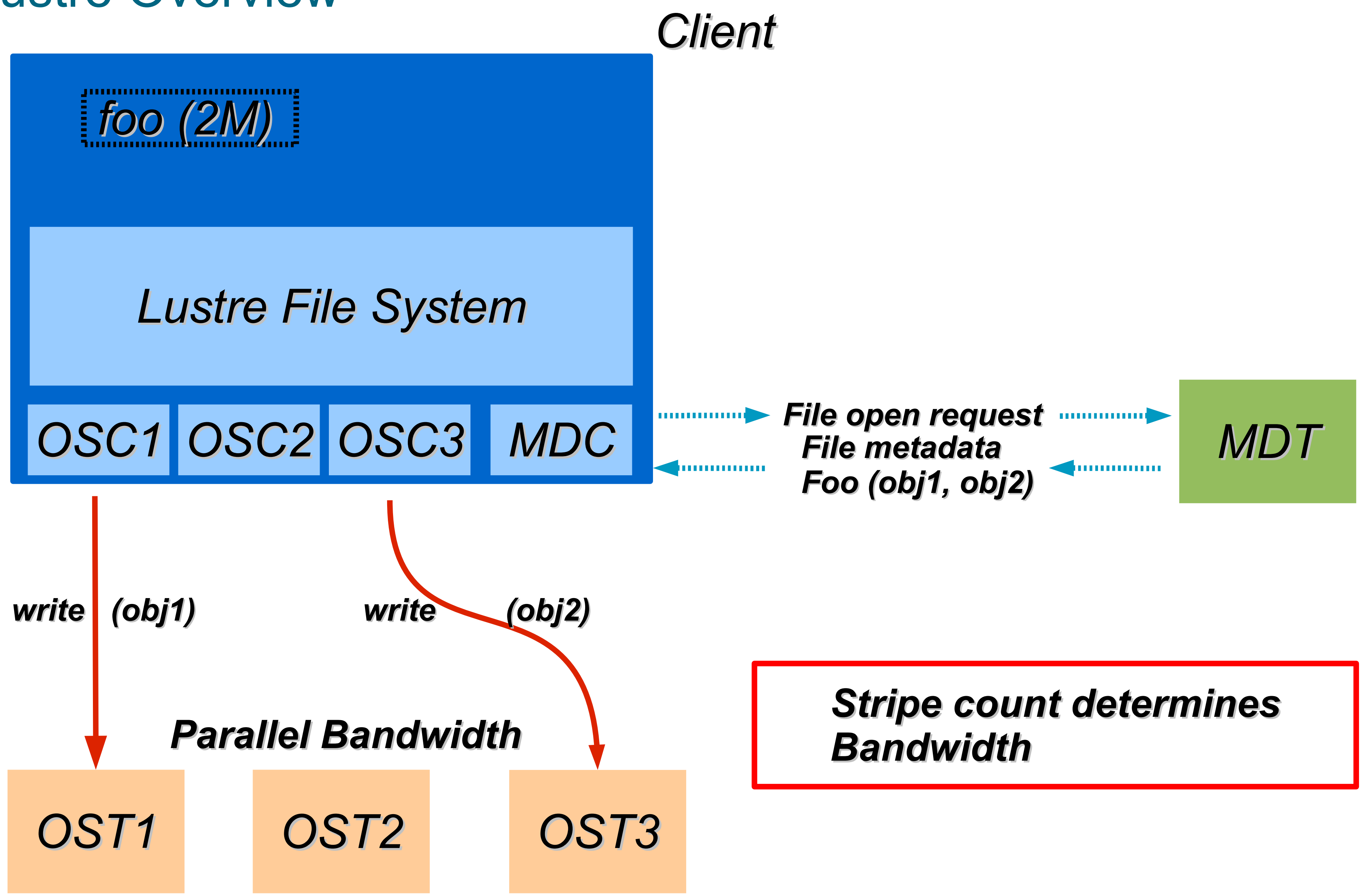
Lustre Overview

- ***How does Lustre accomplish parallism***
  - *Meta Data operation separted from bulk data operations*
  - *Client directly communicate with Storage Nodes*
  - *Performance of storage nodes, targets scale nearly linear*
- ***Complexity is hidden from user via common POSIX file systems interface***
  - *UNIX directories, files, ownership, permissions, ACLs, …*
  - *Common UNIX commands can be used*

Lustre Overview



Lustre Overview



Lustre Infrastructure on Juropa

- ***Two type of file systems***
  - *24 small filesystems for homedirectories (`$HOME`)*
  - *Fast scratch file system (`$WORK`)*
- ***$HOME***
  - *`/lustre/jhome1-14`: 2 OSSes and **4 OSTs / per file system***
  - *`/lustre/jhome15-24`: 4 OSSes and 8 **OSTs / per file system***
  - *Possiblity to backup / restore within reasonable time*
  - *Throughput rates 1 - 2 GB/s*
- ***$WORK***
  - *Based on 8 OSSes and **120 OSTs***
  - *No backup!; automatic clean-up for files older than 28 days*
  - *Throughput rates 19.2 GB/s*

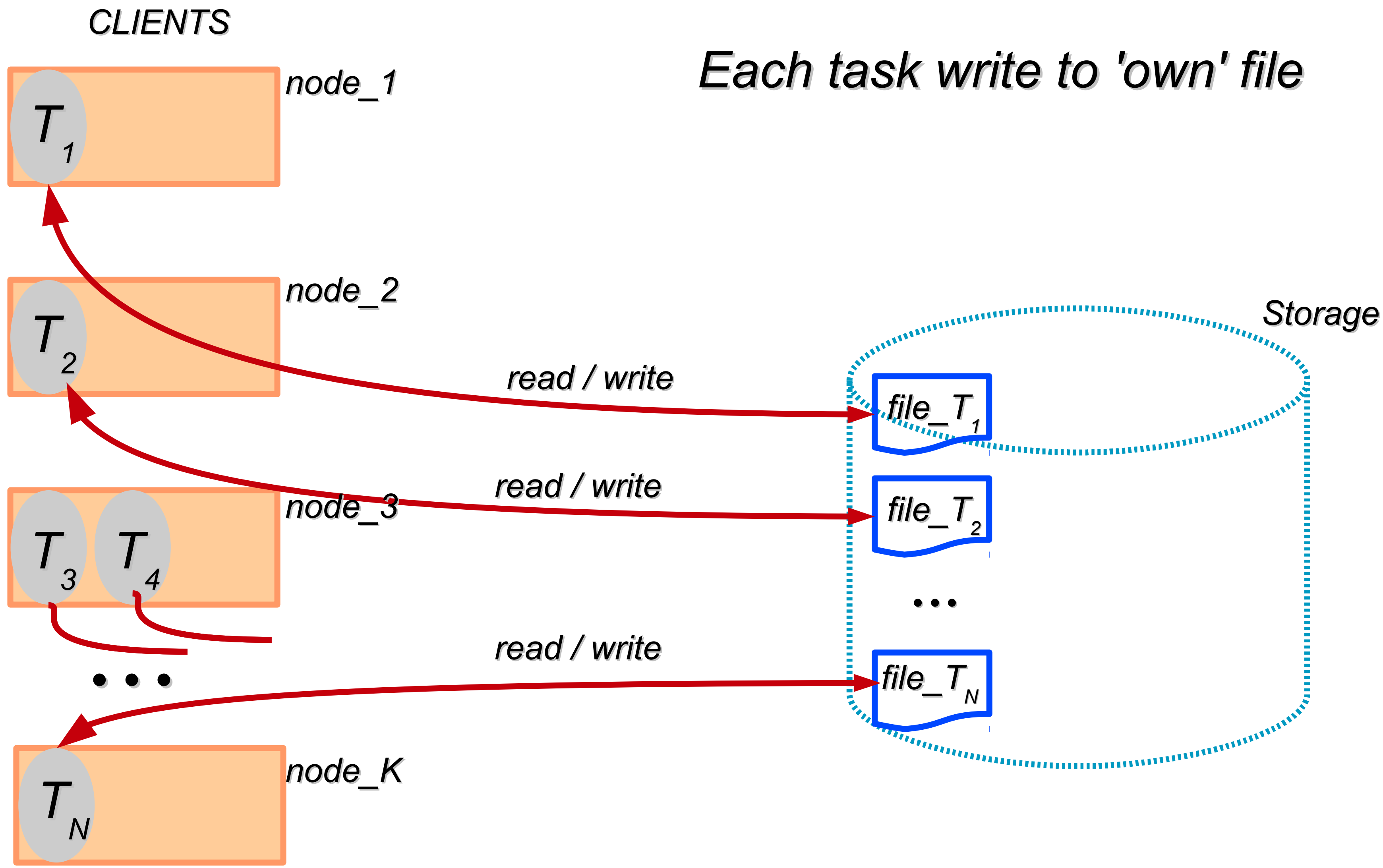
Lustre Infrastructure on Juropa

- ***Each group (project) and group member has:***
  - *storage location (directory) in* `$HOME`
    (Note: Project is alway located on SINGLE lustre file system)
  - *Storage location (directory) in* `$WORK`
- ***Storage allocation are controlled by Lustre quota system. Quota applied on group / project level***
  - *=> Project members share storage resources*
    (potentially block their work)
  - *Default quota policy for* `$HOME`, `$WORK`*:*
    - *3 TB (block quota)*
    - *2 000 000 files (file (i-node) quota)*

Storage Concepts

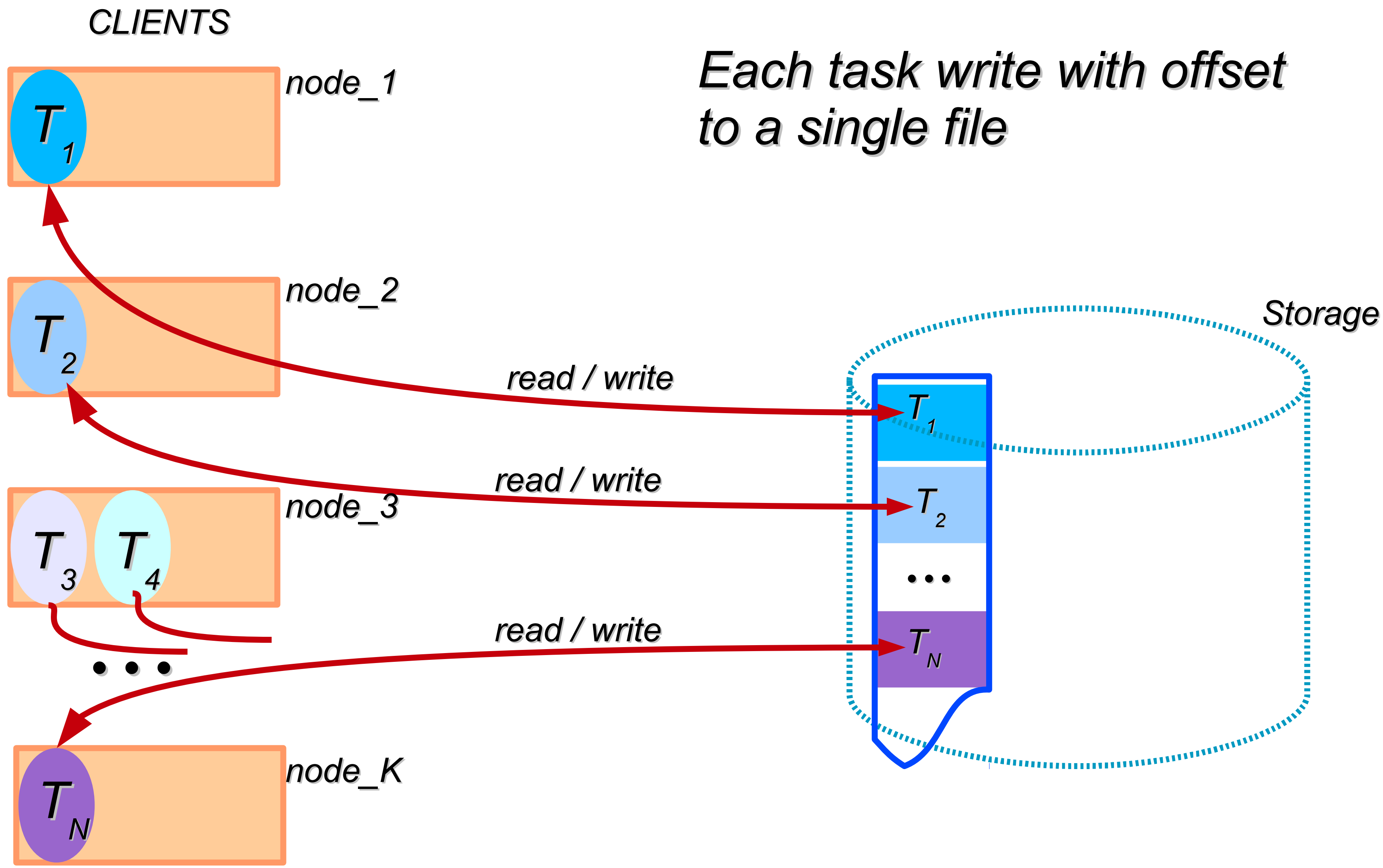
- ***Different strategies how parallel application can create persistent data:***
  - *Each task write to an own file (task local)*
  - *All tasks write to a single file*
  - *Multiple files and each set of tasks write to an associated file*

Storage Concepts (task local)

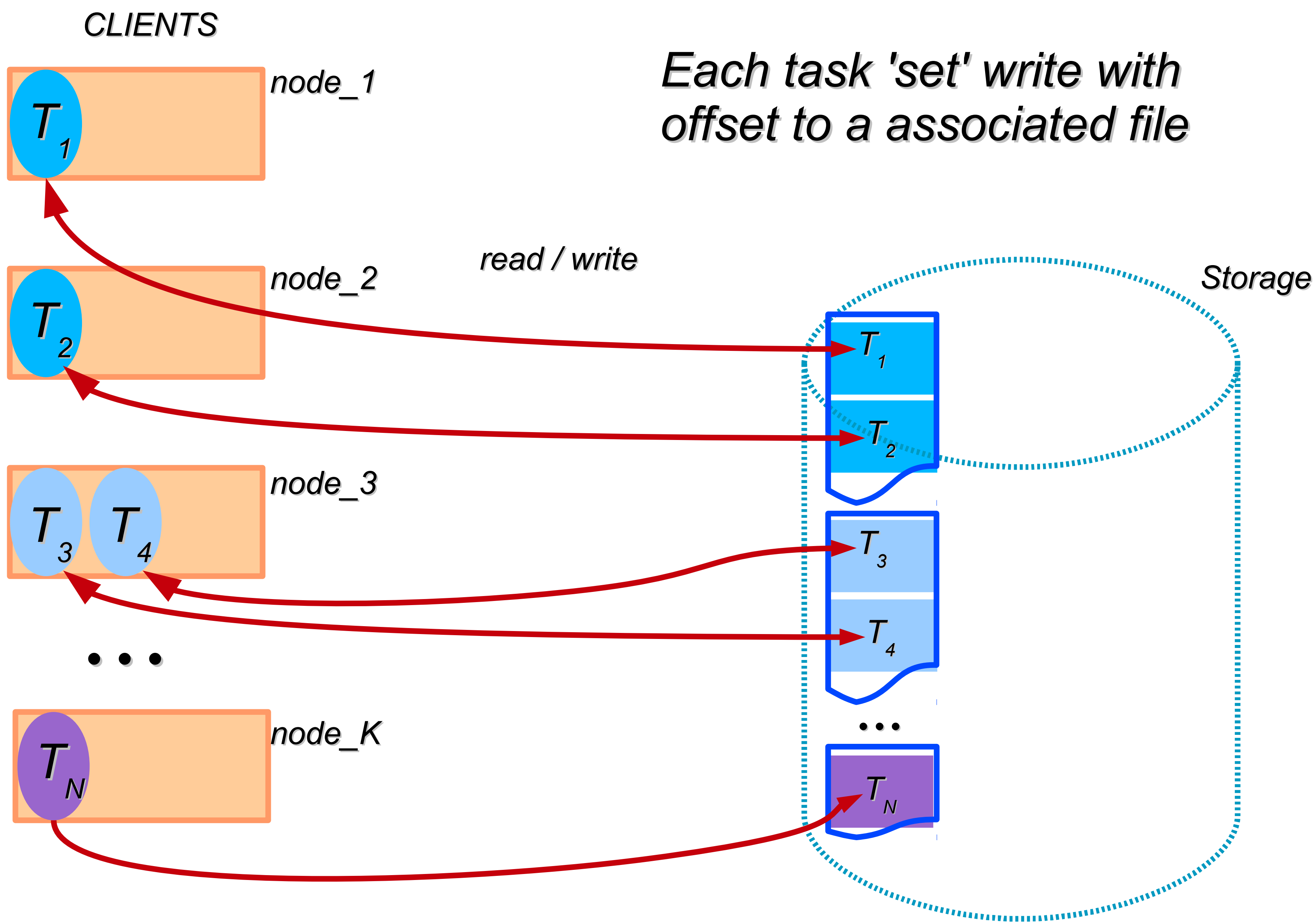


Storage Concepts (single file)

Each task write with offset to a single file



Storage Concepts (multiple files)



Each task 'set' write with offset to a associated file

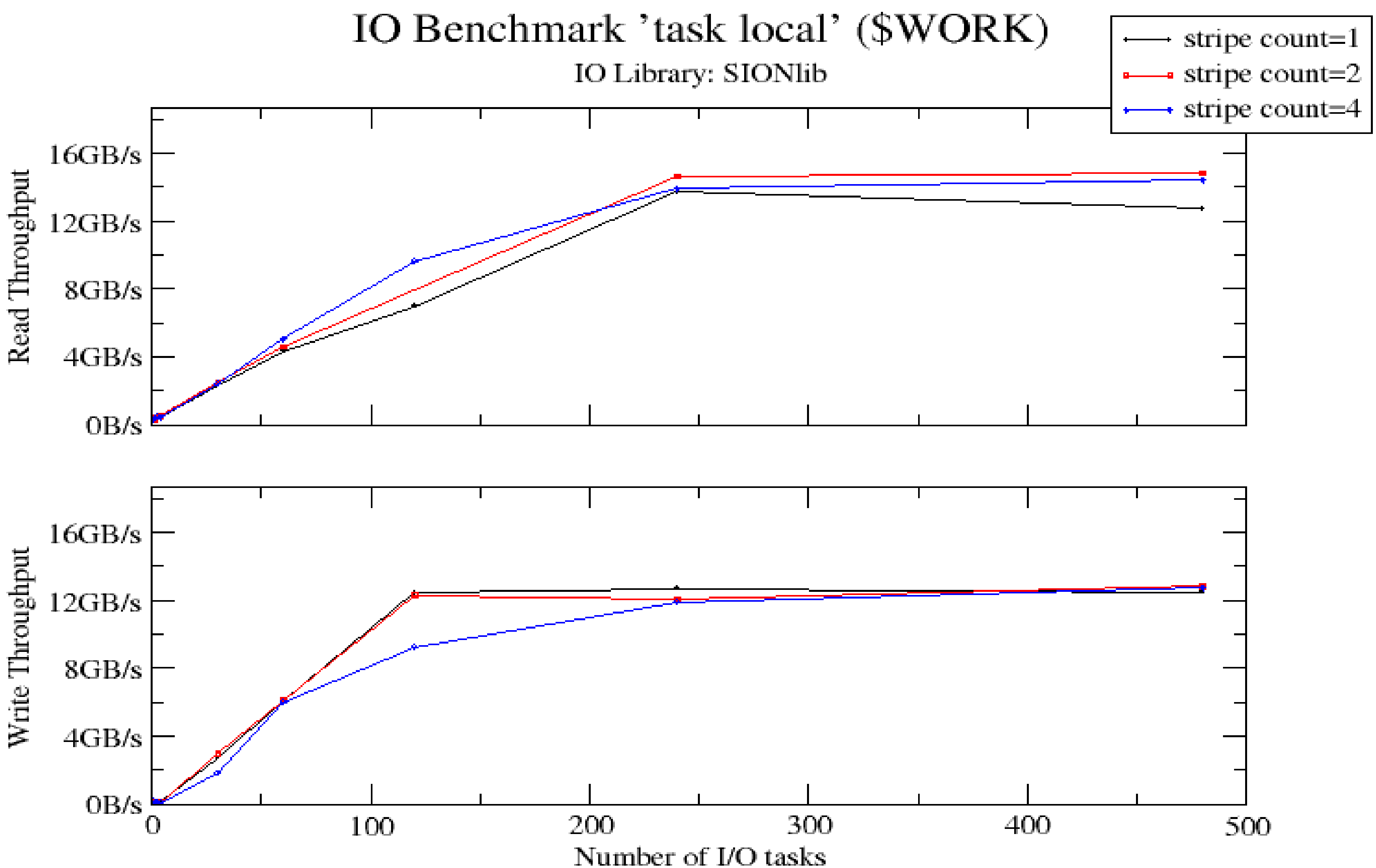
Storage Concepts

- ***IO Libraries***
  - *Standard IO (fopen, printf, cin, cout,...)*
  - *MPI I/O*
  - *SIONlib (portable data format)*
    - *Supports all I/O models*
    - *Via module file (`/usr/local/sionlib`)*
  - *HDF5 (portable data format)*
    - *Support single file*
    - *Includes, Libraries in `/usr/local/hdf5`*
  - *NetCDF (portable data format)*
    - *Includes, Libraries in `/usr/local/netcdf`*
  - *…*

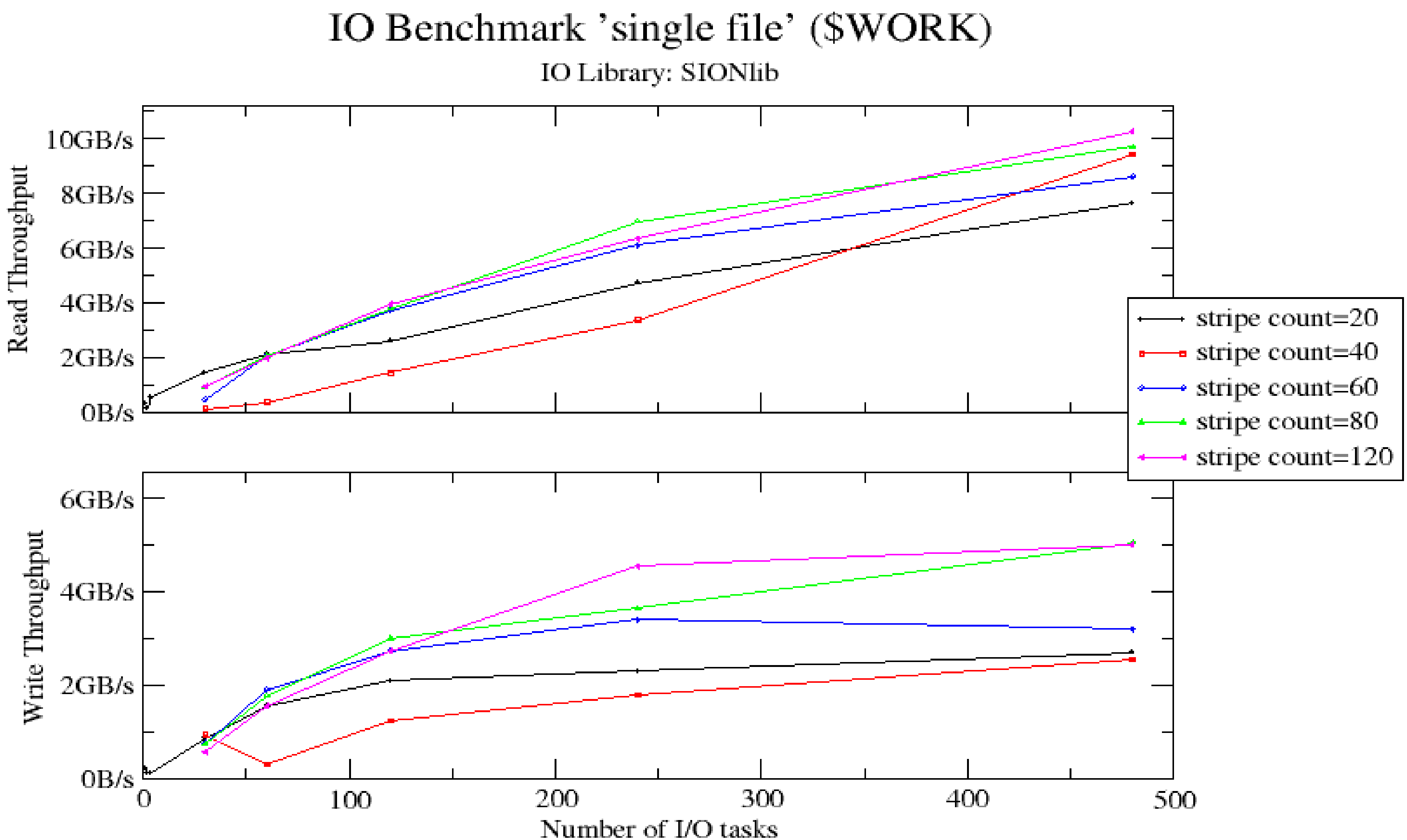
Usage Recommendations

- ***Essential parameter: Number of OSTs used for striping***
- ***'Multiple files' gives the best performance***
- ***For single file increase number of OST***
- ***For task local use default (4 OSTs) or less***
- ***Use number of OST (stripe count) as tuning parameter***
- ***Use number of files as additional tuning parameter for 'Multiple files'***

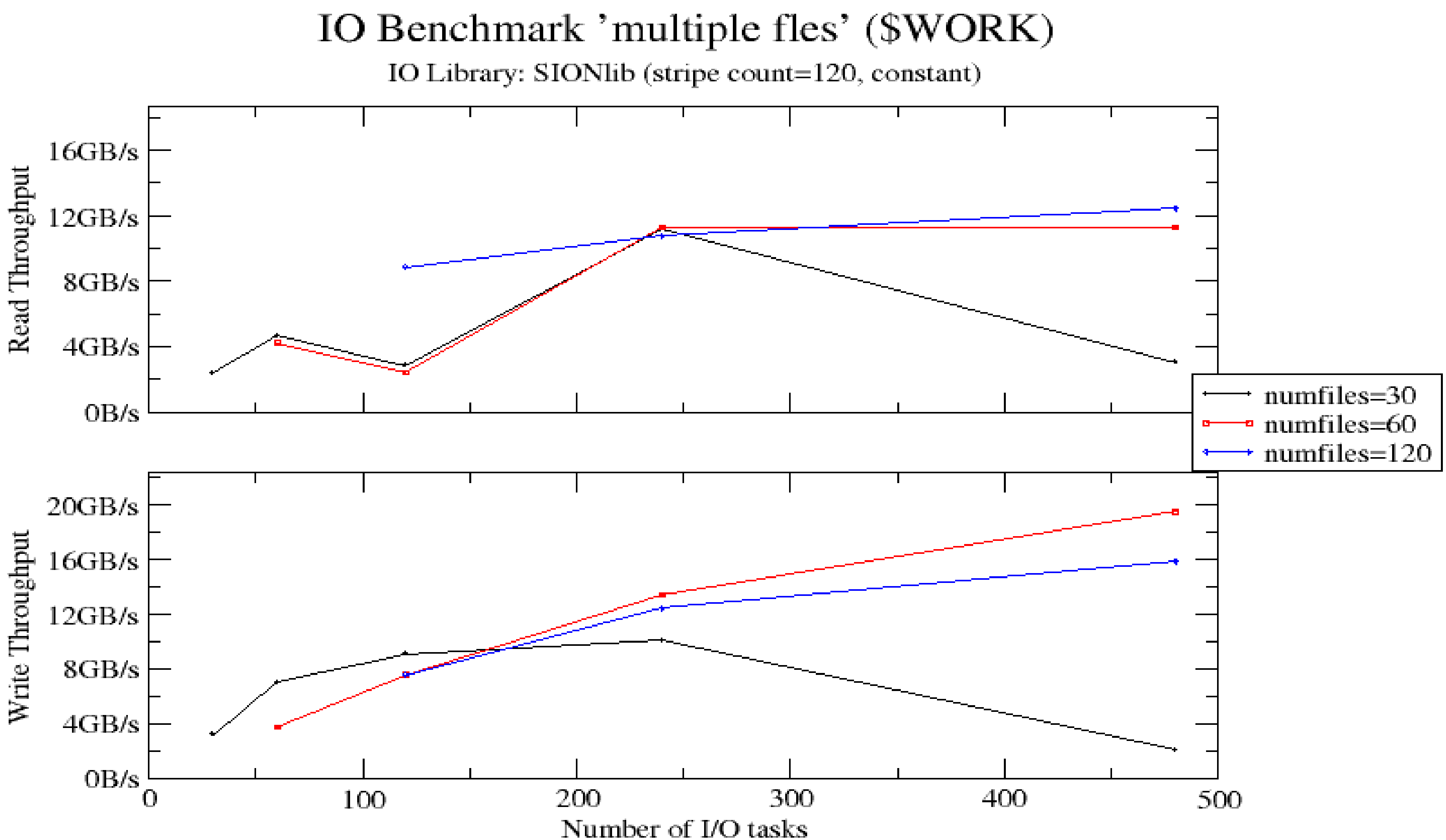
Usage Recommendations



Usage Recommendations



Usage Recommendations



Usage Recommendation

- ***Always use*** `$HOME`***,*** `$WORK` ***to refer to files***
- ***Note: JUROPA use*** *'little endian'*
- ***Don't store more than 10000 files / directory***
- ***Avoid*** `ls -tty=color` → ***All OSS will query for extended file attributes; slow down up to factor 100***
- ***Use Lustre Find tool to search in Lustre file systems (less options, but more efficient):***

```
lfs find <option> <start_directory>
lfs find -name my_foo\*.cpp $WORK/big_files
```

Usage Recommendations

- ***Quota***
  - *Check quota allocation with help of:*
    - `q_lustrequota`
    - `cat $home/../usage.quota`
    - *List disk allocation on all Lustre file systems*
  - *Quota exceeded; application (write) fail with error:*

```
<application_name>: writing `<file_name>': Disk quota exceeded
<application_name>: closing `<file_name>': Input/output error
```

  - *Move data meant for long term storage to* `$GPFSARCH`
  - *Delete superfluous files*

Usage Recommendations

- ***Lustre parameter***
  - *Use default settings*
    - `$WORK` **:** *stripe count : 4*
    - `$HOME` **:** *stripe count : 1*
    - *Use 1 M stripe size (default)*
  - *Command line interface* `lfs`
    - ***Get stripe count***
      `lfs getstripe <file_name>`
    - ***Set stripe count***
      `lfs setstripe -c #number_of_OSTs <file_name>`
      (Directory bequeath stripe count to 'children')
    - ***Find files***
      `lfs find -name <pattern> <lustre-fs-directory>`
  - *C programming API ; Documentation in manual pages*
    `liblustreapi(3),`
    `llapi_file_create(3),...`

Summary

- ***Lustre behave like a 'ordinary' Linux file system***
- ***Follow some basic rules:***
  - *Keep file number / directory below 10000*
  - *Use `$WORK` as project directory for fast / big IO*
    Use `$HOME` for source code, libraries, results
    Use `$GPFSARCH` as long term storage
  - *Keep track of disk allocation (quota)*
  - *Clean-up `$HOME, $WORK`*
  - *Save results from `$WORK`*
  - *File system are a shared resource*
- ***Find best I/O model and library for your application(s)***

Documents / Links

- ***Usefull manuals / documents pages***
  - `lfs(1)` *- User command line interface for all stripe related operations, find, quota operations*
  - *C API:*
    - *llapi_file_create(3)*
    - *llapi_file_get_stripe(3)*
    - *llapi_file_open(3)*
    - *llapi_quotactl(3)*
    - *Liblustreapi(7)*
    - *Examples in Lustre User Group Presentation (page 11):*
      LUG 11, Nick Cardo, Detecting Hidden File System Problems
      http://www.olcf.ornl.gov/wp-content/events/lug2011/4-13-2011/230-300_Nick_Cardo.pptx

Documents / Links

- ***Need help: contact sc@fz-juelich.de***
- ***JUROPA Web-pages***
  - *http://www2.fz-juelich.de/jsc/juropa*
    - Filesystem specific:
    http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUROPA/UserInfo/Filesystems.html
    - I/O Tuning:
    http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUROPA/UserInfo/IO_Tuning.html
    - Data Conversion:
    http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUROPA/UserInfo/DataConversion.html
- ***Lustre Documentation***
  - *http://www.lustre.org*
  - *http://www.whamcloud.com*

Documents / Links

- ***Parallel I/O and Data formats***
  - *SIONlib*
    - *http://www2.fz-juelich.de/jsc/sionlib*
  - *NetCDF webpage*
    - *http://www.unidata.ucar.edu/software/netcdf/*
  - *HDF5 webpage*
    - *http://www.hdfgroup.org/HDF5/*
  - *MPI I/O*
    - *MPI: A Message-Passing Interface Standard, Version 2.2, Chapter 13, I/O*

QUESTIONS?

The End

Thank you