BLUE WATERS

SUSTAINED PETASCALE COMPUTING

November 15, 16 2016

Blue Waters Local Software To Be Released: Module Improvements and Parfu Parallel Archive Tool

Craig P Steffen csteffen@ncsa.Illinois.edu

Blue Waters Science and Engineering Applications Support Team (SEAS)

GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION

see parfu.net

Software On Blue Waters

- Cray-provided software on the System
  - Linux SLES OS
  - Cray-provided compilers
  - System management software
- Users bring application software
  - Applications
  - Build system (specific to application)
  - Control scripts (specific to application and specific scientific problem)

Software on Blue Waters continued…

- Software includes scripts and configuration to fit application/problem combination to Blue Waters environment.
- Some such configuration is specific to a problem
- Some scripts are not
- However, some users have to solve *the SAME (or similar) problems* that have already been solved

SEAS-Provided Software On Blue Waters

- Science and Engineering Applications Support Team has created software to help teams use Blue Waters more efficiently
- Examples in production on Blue Waters:
  - internal *TopAware* tool that evaluates communications and recommends a custom rank order to optimize communications
    - by Bob Fiedler of Cray
  - *The Aggregate Job Launcher of Single-core or Single-node Applications on HPC Sites*
    - by Victor Anisimov of the SEAS team
    - https://github.com/ncsa/Scheduler

Presented Here: Two Pieces of Software Soon To Be Released

- *Module Improvements* (Cray-specific)
  - Greatly streamlines using module commands on Cray systems
  - In production on Blue Waters almost 2 years
  - Open-source release as soon as new feature set tested on Blue Waters
- *Parfu* Parallel Archive Tool
  - Analgous to **`tar`**
  - Creates & extracts archives of directories and files
  - Second version under test on Blue Waters
  - Will be launched locally then open-source released for external testing

see parfu.net

Potential Users of This Software

- *Module Improvements*:
  - Cray systems controlled by Environment Modules
  - Sysadmins of same
  - Users too! (Can easily be installed in a user account)
- *Parfu*:
  - Sites with user workflows that involve extremely numerous small files
  - Bioinformatics workflows
  - Anyone who wants to increase efficiency of storing directories

Module Improvements (Modimp) Background

- Module Environments allows multiple software versions to co-exist; controls effects my shell environment
  - Cray uses Module Environments to control compiler versions and options, among other things
- Cray-provided Environment Modules is as close as possible to upstream source (modules.sourceforge.net)
- We implemented changes for our own use on Blue Waters:
  - Some general usability enhancements
  - Some Cray-specific tweaks

ModImp: bash only

- some features only possible in bash
- 99%+ of Blue Waters users use have bash as their shell, so worth the effort

see parfu.net

Module Improvement 5 Major Features

- module command outputs to standard-out (not standard-error)
- New tab-completion of parameters of module commands (including Cray-specific tweaks)
- Environment-sensitive dynamic prompt
- Disambiguate for tab-completion
- New Cray-specific module sub-command: PrgEnvLoad
  - (new feature; currently under pre-production test)

ModImp: module commands output to stdout

- module available | grep huge
  - (doesn't work in stock Module Environments)
- module available | grep huge
  - (does work with Module Improvements installed)
- Background and history too long to cover here
  - Good news: upstream now has solution in their roadmap
  - No upstream release yet with this fix (3 years)
  - Bad News: no major version yet; won't be available in Blue Waters time frame

ModImp: Tab-completion for module commands

<TAB> means "push TAB key once"

- module **<TAB>**
  - tab-completes module sub-commands
- module load **<TAB>**
  - tab-completes available modules
- module unload **<TAB>**
  - tab-completes loaded modules
- module swap mymod **<TAB>**
  - tab-completes all mymod/*<version>*

(note: the Cray module package now does include upstream tab-completion. It did not in 2015 when we started work on Module Improvements.)

ModImp: Tab Completion 2: Cray-specific tweaks to module name mask

- module load hug**<TAB>**
  - completions include all hugepages modules,

  including “craype-hugepages2M”

(see demo for examples)

ModImp Tab-Completion 3…

- Cray PrgEnv-* modules used to select compiler:
  - PrgEnv-cray, PrgEnv-pgi, PrgEnv-gnu, PrgEnv-intel
- module swap Prg**<TAB>**
  - auto-completes to currently-loaded PrgEnv-* module name
- module swap PrgEnv-cray **<TAB>**
  - tab-completes with ALL PrgEnv-* modules, not just PrgEnv-cray versions

(see demo for examples)

ModImp: Dynamic Prompt

- Bash-only
- Uses PROMPT_COMMAND to change prompt in response to shell environment changes (or other system state)
- Useful for keeping track of frequently loaded/unloaded modules, compiler types, (perhaps other system information?)

Modimp Dynamic Prompt: Compiler

```
csteffen@h2ologin2 23:07 ~/tmp _D_-  1-Cray-_ $ module swap PrgEnv-cray/5.2.82 PrgEnv-gnu
csteffen@h2ologin2 23:08 ~/tmp _D_-  1-Gnu-_ $
csteffen@h2ologin2 23:08 ~/tmp _D_-  1-Gnu-_ $ module PrgEnvLoad PrgEnv-cray
csteffen@h2ologin2 23:08 ~/tmp _D_-  1-Cray-_ $ module PrgEnvLoad PrgEnv-pgi
csteffen@h2ologin2 23:08 ~/tmp _D_-  1-PGI-_ $ module PrgEnvLoad PrgEnv-intel
csteffen@h2ologin2 23:09 ~/tmp _D_-  1-Intel-_ $ module PrgEnvLoad PrgEnv-cray
```

ModImp Dynamic Prompt: Stripe Count of Current Directory

```
csteffen@h2ologin2 23:15 /scratch/staff/csteffen/striping _D_-  1 Cray-_ $ cd stripe_004/
csteffen@h2ologin2 23:15 /scratch/staff/csteffen/striping/stripe_004 _D_-  4-Cray-_ $ cd ..
csteffen@h2ologin2 23:15 /scratch/staff/csteffen/striping _D_-  1-Cray-_ $ cd stripe_032/
csteffen@h2ologin2 23:15 /scratch/staff/csteffen/striping/stripe_032 _D_- 32-Cray-_ $ cd ..
csteffen@h2ologin2 23:15 /scratch/staff/csteffen/striping _D_-  1-Cray-_ $ cd stripe_160/
csteffen@h2ologin2 23:15 /scratch/staff/csteffen/striping/stripe_160 _D_-160-Cray-_ $ cd ..
csteffen@h2ologin2 23:15 /scratch/staff/csteffen/striping _D_-  1-Cray-_ $ cd /tmp/
csteffen@h2ologin2 23:15 /tmp _D_-XXX-Cray-_ $
```

see parfu.net

ModImp Dynamic Prompt: Conflicting Modules

- `csteffen@h2ologin2 23:20 ~ _D_-  1-Cray-_ $ module unload darshan/2.3.0.1`
- `csteffen@h2ologin2 23:20 ~ ___-  1-Cray-_ $ module load perftools-base`
- `csteffen@h2ologin2 23:20 ~ P__-  1-Cray-_ $`

Dynamic Prompt: presence of Makefile

```
csteffen@h2ologin2 23:27 ~ P__-  1-Cray-_ $ cd
csteffen@h2ologin2 23:27 ~ P__-  1-Cray-_ $ ls Makefile
ls: cannot access Makefile: No such file or directory
csteffen@h2ologin2 23:27 ~ P__-  1-Cray-_ $ cd build_dir/
csteffen@h2ologin2 23:27 ~/build_dir P__-  1-Cray-M $ ls
Makefile  my_source.c
csteffen@h2ologin2 23:27 ~/build_dir P__-  1-Cray-M $ cd ..
csteffen@h2ologin2 23:27 ~ P__-  1-Cray-_ $
```

ModImp Dynamic Prompt: Possible future new tags?

- User tools
  - Jobs in queue, jobs running
  - Allocation state
  - Permissions for current dir
  - Physical location of current dir
  - Svn/git status of current dir
- Admin tools:
  - Utilization
  - Queue pressure/health

ModImp: Display All When Ambiguous

- bash default behavior:
  - tab 3 times until bash displays all possible completions
- turn “show all if ambiguous” on
  - tab *immediately* displays list of completions
- (not enabled by default since it effects ALL tab-completion, not just for “module”)

Modimp: New Module Subcommand: PrgEnvLoad

- PrgEnv-cray, PrgEnv-gnu, PrgEnv-pgi, PrgEnv-intel mutually exclusive
- To swap, must use “module swap <from> <to>
- Swap subcommand not re-entrant:
  - “module swap PrgEnv-gnu PrgEnv-cray” works when compiler is gnu, NOT when it is cray
- new subcommand:
  - module PrgEnvLoad PrgEnv-cray
    - works if module was gnu
    - works if module was cray
    - works if no PrgEnv-* module was loaded at all

ModImp is Completely Configurable

- All four major features can be turned on and off independently by user with a single command
  (modimp_module_to_stdout_on,modimp_module_to_stdout_off, etc.)
- System install has sensible defaults
  - module→stdout and tabcompletion on
  - dynamic prompt and display-all-if-ambiguous off
  - customizable at install time
- Typically you'll put your modimp initialization in your .profile

Modimp Dynamic Prompt is Completely Configurable

- Each element of dynamic prompt can be enabled and they can be ordered; user can experiment with configuration by changing PROMPT_COMMAND environment variable
- modimp_prompt_reset puts your prompt back to a sensible configuration
- modimp_prompt_commit writes current configuration to .profile for future use

ModImp Current Status

- 4 features in production on Blue Waters more than a year
- new version including PrgEnvLoad command now in pre-testing for a month or two
- Has preliminary installers, need to construct final versions
  - (Some features increase user's environment size by tens of kB; final installers will have options to (dis/en)able those features)
- Will release under U of I OS license when finished (December?)

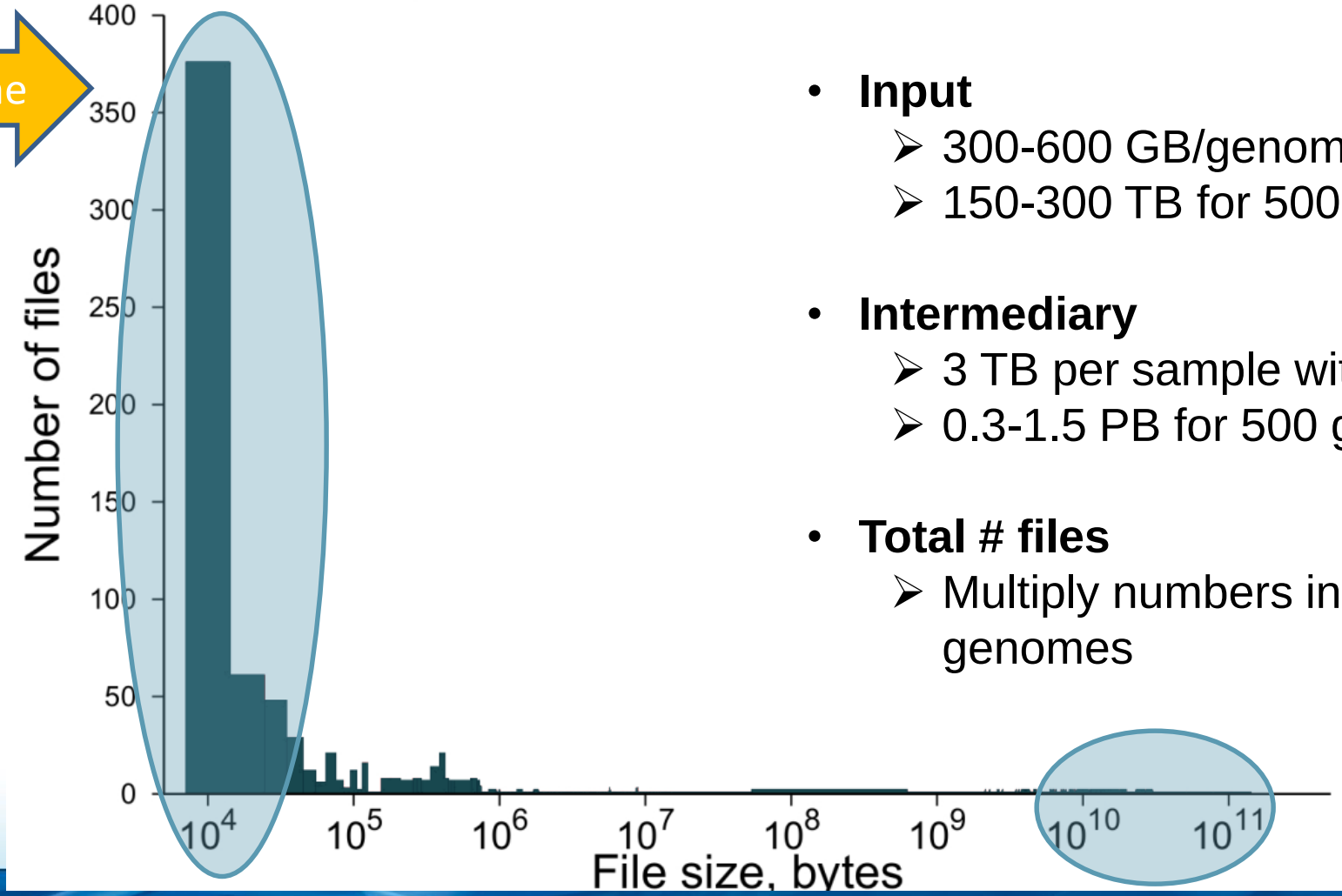
Parfu Parallel Archive Tool

- Motivation: workloads with very numerous small files
  - Many (>10,000) entries in a directory makes Lustre less happy
  - Storing directory trees in tape libraries with millions of files fragments the storage, making retrieval slow to impossible

see parfu.net

Human genotyping

Each genome

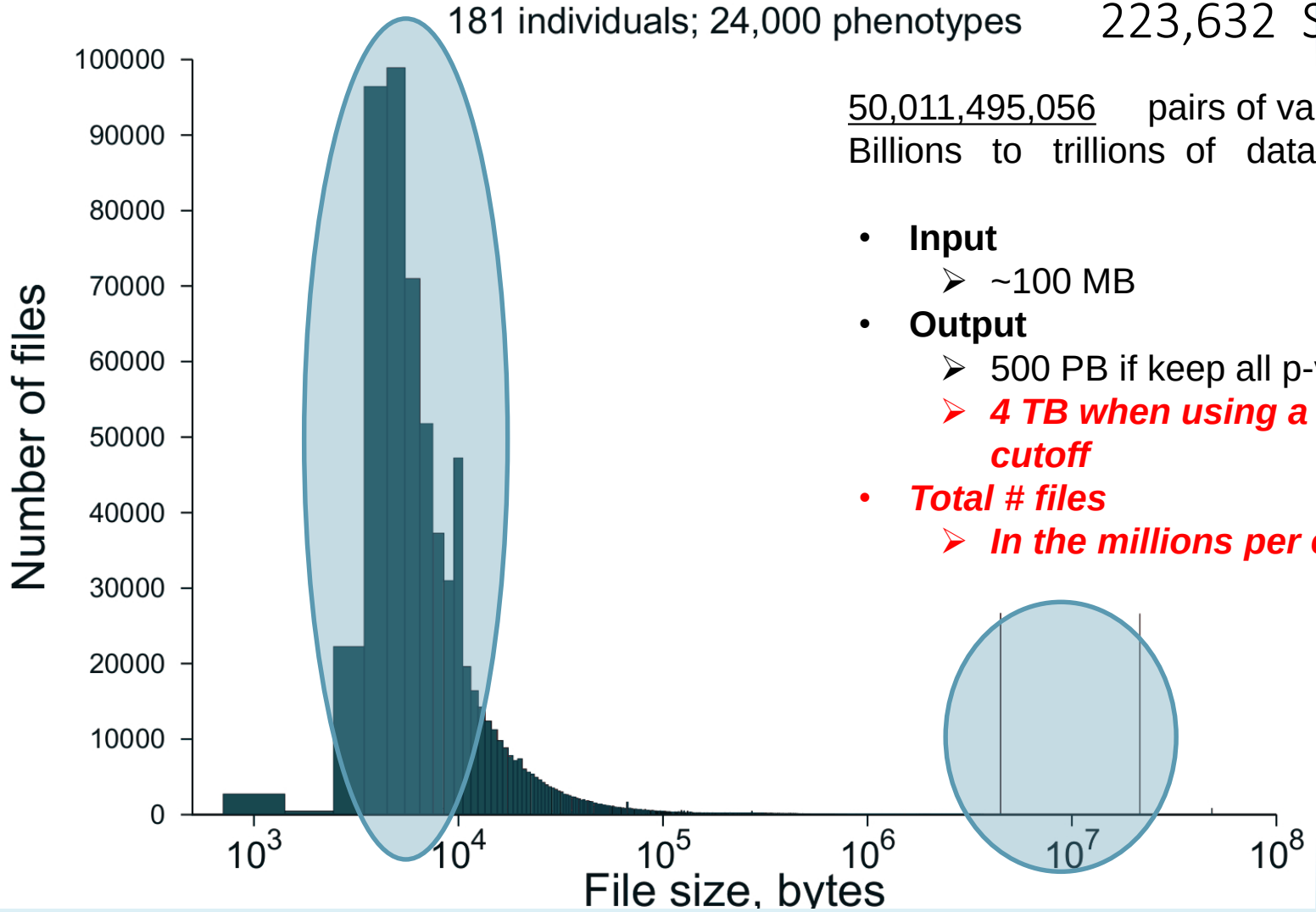


- **Input**
  - 300-600 GB/genome
  - 150-300 TB for 500 genomes
- **Intermediary**
  - 3 TB per sample with intermediaries
  - 0.3-1.5 PB for 500 genomes
- **Total # files**
  - Multiply numbers in the graph by 500 genomes

see parfu.net

Alzheimer's eGWAS, epistasis

181 individuals; 24,000 phenotypes 223,632 SNPs



50,011,495,056 pairs of variants
Billions to trillions of datapoints

- **Input**
  - ~100 MB
- **Output**
  - 500 PB if keep all p-values
  - ***4 TB when using a conservative p-value cutoff***
- ***Total # files***
  - ***In the millions per experiment***

Parfu Motivation: Why not just `tar` them up?

- tar is too slow; burns too much job time
- ptar, pigz are better, aren't enough to solve problem
- (haven't had a chance to test htar; in any case, it's tied to storage and requires special priviledges)
- Does a tool exist to do this that's a parallel application? We couldn't find one available.

Other Possible Candidate Codes

- pltar at ORNL?
  - Being developed <2012
  - I talked author; never released
  - ORNL says: that did exist but it's not around any more

see parfu.net

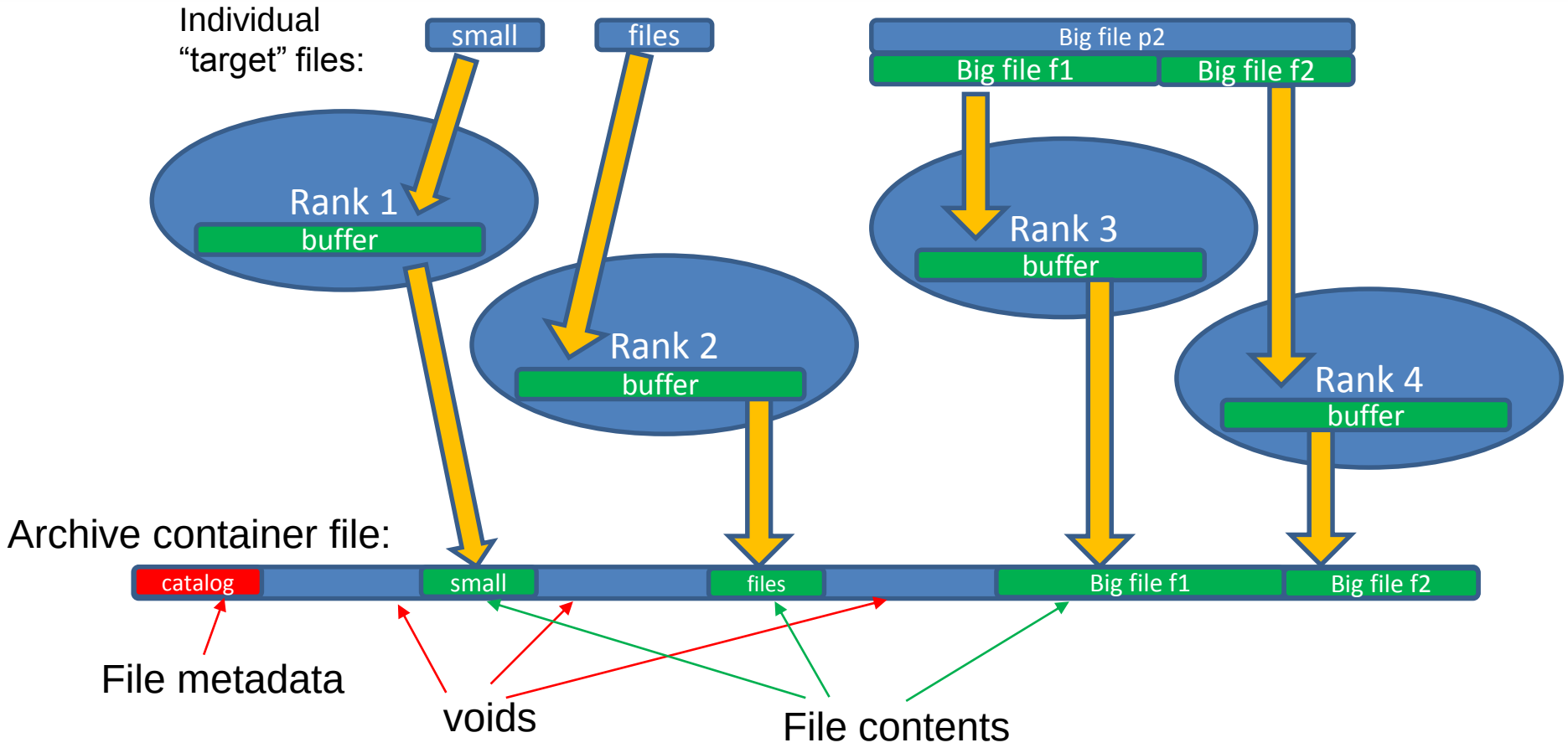
What We Need: Many-to-One Solution That's Fast

- Is there an existing solution that allows you to seriously throw nodes at this problem? Not that I've found.
- Solution speed ideally should be proportional to the number of nodes (RAM bandwidth/I/O bandwidth/network bandwidth should all scale)
- We want: something to integrate into the workflow with minimal disruption to established workflow(s)
  - Possibly integrate into storage solutions *once it's in a production version* (a future step)

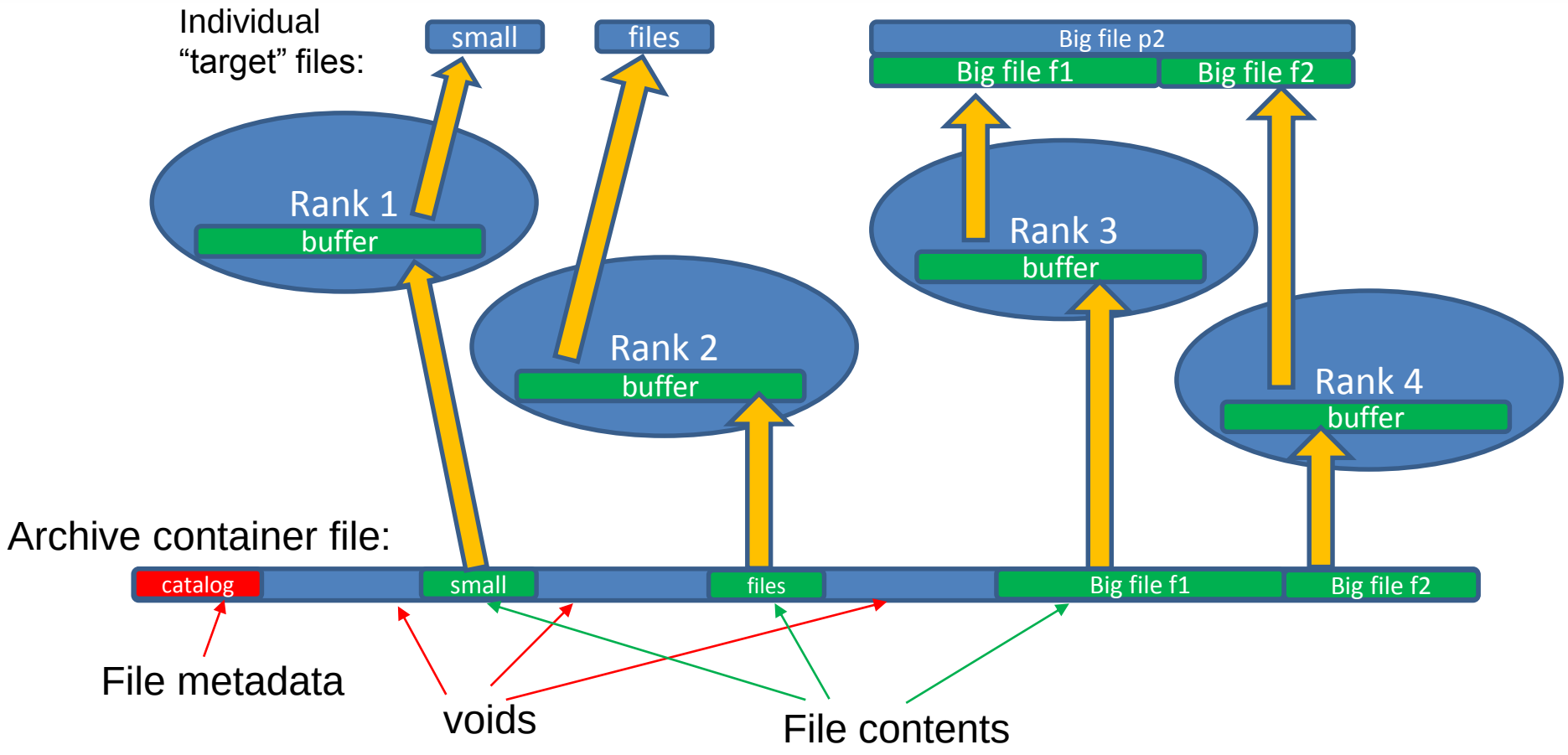
Our Solution: Parfu

- Distributed (runs n ranks)
- Each separate file (or file fragment for big files) read and written by a separate rank
- Tar-analagous (many-to-one; files are NOT tar-compatible)
- MPI with MPI_IO

Parfu: How Data Moves 1: ("Create" mode)



Parfu: How Data Moves 2: ("Extract" mode)



File Storage Philosophy

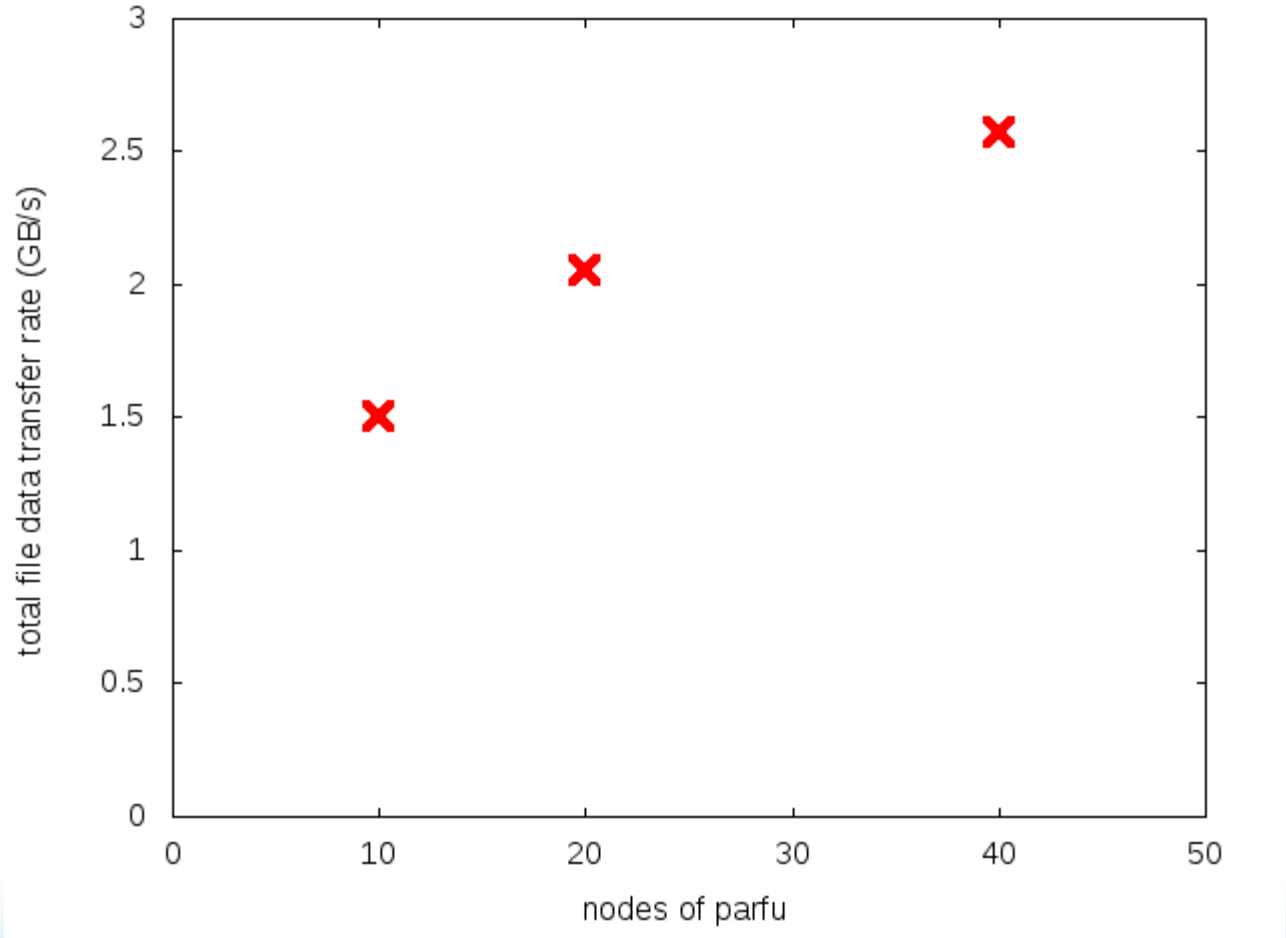
- Files are spaced out in (Parfu-defined) “blocks”, (the largest of) which are multiples of file system stripe size, for I/O efficiency
- Files are sparsely stored, with voids in between
- No compression
- Block size is dynamic, so that a 20 byte file doesn’t allocate 1 MB of space
  - A possible trade-off between file locking and efficiency of storage
  - minimum block size is being studied and can be controlled by command-line flags

Performance: Better than Tar etc., Has Scaling Worries

- No limit found for nranks (at least 1380 across 60 nodes)
- No limit found for max file size or max number of files in a single archive (successful at > 1 M files)
- Baseline performance roughly 10x speedup above tar and tar-like solutions (3-hour tar operation parfu can do in 20 minutes)
- Seems to have unknown or artificial total bandwidth limit (2-3 GB/s)

Non-Understood Performance Limitation

- Seems to have at least one bottle-neck in implementation
- Speed tops out at 2 to 3 GB/s to archive file, scales very sub-linearly with number of nodes and ranks
  - 10 nodes: ~ 1 GB/s
  - 60 nodes: ~ 2 GB/s
- Why? (Under investigation.)



see parfu.net

Parfu History and Status

- A couple of prototype versions run and tested on Blue Waters by staff and Bioinformatics research Luda Mainzer
- No fundamental limitations found so far (total archive file size, number of archived files, Nranks)
- Using current version understand bandwith scaling limitations, testing new (more tar-like) command-line configuration
- Plan to release in January 2017

see parfu.net

Upcoming feature list (AFTER initial release)

- make archive files tar-compatible
- explore the possibility of compressing files or file fragments
  - (may not be compatible with parfu's fast-and-efficient philosophy, but it's worth checking)
- possibly, eventually, explore integrating with storage technologies

see parfu.net

Thanks

- The Blue Waters sustained-petascale computing project for supporting this work, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and The State of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications
- Bill Kramer for supporting this work and for allowing this software to be released.
- Luda Mainzer for extensively testing parfu and breaking the command-line.
- Jeremy Enos and the Blue Waters sysadmin team and the Cray sysadmins for allowing me to contribute to system software.
- Greg Bauer for (almost) always supporting and encouraging me when I announce that I'm going to spend time slaying a Dragon.

Where To Get Information and Status

- Package information pages:
  - ncsa.illinois.edu/People/csteffen/parfu
  - ncsa.illinois.edu/People/csteffen/Module_Improvements
  - github.com/ncsa/parfu_archive_tool
  - github.com/ncsa/module_improvements
- ***announcement page***: ncsa.illinois.edu/People/csteffen/sc2016
- link page to the above: ***parfu.net***
- Feel free to contact me with questions: csteffen@ncsa.illinois.edu or if you're interested in an announcement when they are released
  - please put "parfu" or "modimp" in the subject line