Research Data Management: Technical Aspects

Ramses van Zon & Marcelo Ponce

SciNet HPC Consortium
University of Toronto

May 25th, 2016

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Second Part

In this section, we will discuss the following topics:

- Command line tools
- Version Control: git, hg, cvs, svn, git-annex
- Storage Limits on SciNet
- Compressing data
- Moving data
- File formats & Metadata

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Section 1

Basics of Research Data Managament

Recap from Dylanne & Leslie presentation...

Research Data Managament

- **Data and the Data Lifecycle**
- **Retention, Preservation, and Sharing**
- **Documentation and Description**
- **File Formats & Metadata**
- **Version Control**

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Section 2

Command Line Tools

Use case

We will look at a somewhat messy file structure to see how you could explore it and clean it up.

```
$ cd myresearch
$ ls
```

Use case

We will look at a somewhat messy file structure to see how you could explore it and clean it up.

```
$ cd myresearch
$ ls
```

```
analyse1.sh run0199.out run0400.in  run0600.txt run0801.out
analyse2.sh run0199.txt run0400.out run0601.in  run0801.txt
app.cc      run0200.in  run0400.txt run0601.out run0802.in
app.o       run0200.out run0401.in  run0601.txt run0802.out
app.x       run0200.txt run0401.out run0602.in  run0802.txt
Makefile    run0201.in  run0401.txt run0602.out run0803.in
previous    run0201.out run0402.in  run0602.txt run0803.out
run0001.in  run0201.txt run0402.out run0603.in  run0803.txt
run0001.out run0202.in  run0402.txt run0603.out run0804.in
run0001.txt run0202.out run0403.in  run0603.txt run0804.out
run0002.in  run0202.txt run0403.out run0604.in  run0804.txt
run0002.out run0203.in  run0403.txt run0604.out run0805.in
run0002.txt run0203.out run0404.in  run0604.txt run0805.out
```

Finding data

- **find**: Where are all the *.log files?

Finding data

- **find**: Where are all the *.log files?

```
$ find -name '*.log'
./runs.log
./previous/runs.log
```

Finding data

- **find**: Where are all the *.log files?

```
$ find -name '*.log'
./runs.log
./previous/runs.log
```

- **file**: What is that file?

Finding data

- **find**: Where are all the *.log files?

```
$ find -name '*.log'
./runs.log
./previous/runs.log
```

- **file**: What is that file?

```
$ file runs.log
runs.log: ASCII text
$ file previous
previous: directory
```

Finding data

- **find**: Where are all the *.log files?

```
$ find -name '*.log'
./runs.log
./previous/runs.log
```

- **file**: What is that file?

```
$ file runs.log
runs.log: ASCII text
$ file previous
previous: directory
```

- **cat**: What is in that file?

Finding data

- **find**: Where are all the *.log files?

```
$ find -name '*.log'
./runs.log
./previous/runs.log
```

- **file**: What is that file?

```
$ file runs.log
runs.log: ASCII text
$ file previous
previous: directory
```

- **cat**: What is in that file?

```
$ cat run0002.in
#project mouse
number=102
dose=14525
```

Counting data

- **wc**: Count word.

Counting data

- **wc**: Count word. Good to find e.g., how many *.in files there are:

Counting data

- **wc**: Count word. Good to find e.g., how many *.in files there are:

```
$ find -name '*.in' | wc -w
2000
```

Counting data

- **wc**: Count word. Good to find e.g., how many *.in files there are:

```
$ find -name '*.in' | wc -w
2000
```

- **du**: Disk usage in directory.

Counting data

- **wc**: Count word. Good to find e.g., how many *.in files there are:

```
$ find -name '*.in' | wc -w
2000
```

- **du**: Disk usage in directory.

```
$ du ./
63200    ./previous
176780   ./
```

Comparing stuff

- **diff**: Are these two files the same?

Comparing stuff

- **diff**: Are these two files the same?

```
$ diff run0001.in previous/run0001.in
3c3
< dose=31471
---
> dose=11923
```

Comparing stuff

- **diff**: Are these two files the same?

```
$ diff run0001.in previous/run0001.in
3c3
< dose=31471
---
> dose=11923
```

- **cmp**: For binary files

Comparing stuff

- **diff**: Are these two files the same?

```
$ diff run0001.in previous/run0001.in
3c3
< dose=31471
---
> dose=11923
```

- **cmp**: For binary files

```
$ cmp run0001.out previous/run0001.out
run0001.out previous/run0001.out differ: byte 1, line 1
```

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Comparing stuff

- **diff**: Are these two files the same?

```
$ diff run0001.in previous/run0001.in
3c3
< dose=31471
---
> dose=11923
```

- **cmp**: For binary files

```
$ cmp run0001.out previous/run0001.out
run0001.out previous/run0001.out differ: byte 1, line 1
```

- **ls -l**: Sort long listings, e.g. by size

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Comparing stuff

- **diff**: Are these two files the same?

```
$ diff run0001.in previous/run0001.in
3c3
< dose=31471
---
> dose=11923
```

- **cmp**: For binary files

```
$ cmp run0001.out previous/run0001.out
run0001.out previous/run0001.out differ: byte 1, line 1
```

- **ls -l**: Sort long listings, e.g. by size

```
$ ls -lS *.out
-rw-r--r-- 1 rzon scinet 105030 May 24 16:09 run0829.out
-rw-r--r-- 1 rzon scinet 105030 May 24 14:57 run0830.out
-rw-r--r-- 1 rzon scinet 105030 May 24 16:21 run0831.out
-rw-r--r-- 1 rzon scinet  87733 May 24 14:57 run0851.out
```

run0829.out, run0830.out, run0831.out have the same size?

- **md5sum**: Might these three or more files be duplicates?

run0829.out, run0830.out, run0831.out have the same size?

- **md5sum**: Might these three or more files be duplicates?

```
$ md5sum run0829.out run0830.out run0831.out
9b1e2069055c61d5fd0fbd0797e17735  run0829.out
9b1e2069055c61d5fd0fbd0797e17735  run0830.out
9b1e2069055c61d5fd0fbd0797e17735  run0831.out
$ cmp run0829.out run0830.out
$ cmp run0830.out run0831.out
```

run0829.out, run0830.out, run0831.out have the same size?

- **md5sum**: Might these three or more files be duplicates?

```
$ md5sum run0829.out run0830.out run0831.out
9b1e2069055c61d5fd0fbd0797e17735  run0829.out
9b1e2069055c61d5fd0fbd0797e17735  run0830.out
9b1e2069055c61d5fd0fbd0797e17735  run0831.out
$ cmp run0829.out run0830.out
$ cmp run0830.out run0831.out
```

Checksums

- A checksum is a "number" derived from the content of a file.
- Deterministic: files that are the same will have the same checksum.
- Having the same checksum does not ensure that the files are the same.
- `ls -l` and `md5sum` is useful as a first pass to rule out duplicates.
- One can do checksums of files on different systems!

Looking inside files

- **cat**: show content of text file

```
$ cat run0001.in
#project mouse
number=101
dose=31471
```

- **more**, **less**: *paged* versions of cat.
- **grep**: file patterns in text files

```
$ grep 'project mouse' *.in
run0001.in:#project mouse
run0002.in:#project mouse
run0003.in:#project mouse
run0004.in:#project mouse
run0005.in:#project mouse
run0006.in:#project mouse
run0007.in:#project mouse
run0008.in:#project mouse
```

Storage usage

- **quota**: how much storage am I using?

```
quota
```

Storage usage

- **quota**: how much storage am I using?

```
quota
```

```
Retrieving user quotas for user rzon:

-----------------------------------------------------------
Filesystem          size        quota      #files       limit
-----------------------------------------------------------
home              25.55GB         50GB      112935     1000000
scratch           11.02GB         20TB       10225     1000000
project            7.53MB           __         245          __
archive           75.66GB           __      123272     1000000
-----------------------------------------------------------

Note: For more information regarding the disk usage of you
      and your group, use the 'diskUsage' command.
```

ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Storage usage

- **diskUsage**: more detailed info in your storage

```
diskUsage
```

Storage usage

- **diskUsage**: more detailed info in your storage

```
diskUsage
```

```
-------------------------|----------------|----------------------
.                        |   Space Limits |        File Limits
fs         name      type |     size quota |    files quota avg_size
-------------------------|----------------|----------------------
home       scinet    GRP  |    2.86T    __ | 12805258    __  239.66K
scratch    scinet    GRP  |   21.29T   80T |  2005546   10M   11.13K
scratch2   scinet    GRP  |   17.12T   80T |   888931   10M   20.20K
project    scinet    FSET |    3.76T    5T |   461884    1M    8.53K
project2   scinet    GRP  |    0.00K    __ |        1    __    0.00K
archive    scinet    GRP  |   43.98T    2T |  5166726   10M    8.93K
reserved1  scinet    GRP  |   15.27T    __ |  6230010    __    2.57K
-------------------------|----------------|----------------------
home       rzon      USR  |  254.55G  800G |  1662935    3M  160.51K
scratch    rzon      USR  |   11.02G   20T |    10225    1M    1.10M
project    rzon      USR  |    7.53M       |      245         31.49K
```

Tools to improve your storage layout

- Best thing is to have a good naming scheme and directory structure from the start.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Tools to improve your storage layout

- Best thing is to have a good naming scheme and directory structure from the start.
- But there'll always be some tuning, so you need to be able to do basic stuff such as:

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Tools to improve your storage layout

- Best thing is to have a good naming scheme and directory structure from the start.
- But there'll always be some tuning, so you need to be able to do basic stuff such as:

- **mv**: move or rename files or directories

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Tools to improve your storage layout

- Best thing is to have a good naming scheme and directory structure from the start.
- But there'll always be some tuning, so you need to be able to do basic stuff such as:

- **mv**: move or rename files or directories
- **cp**: copy files

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Tools to improve your storage layout

- Best thing is to have a good naming scheme and directory structure from the start.
- But there'll always be some tuning, so you need to be able to do basic stuff such as:

- **mv**: move or rename files or directories
- **cp**: copy files
- **rm/rmdir**: remove files or directories.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Tools to improve your storage layout

- Best thing is to have a good naming scheme and directory structure from the start.
- But there'll always be some tuning, so you need to be able to do basic stuff such as:

- **mv**: move or rename files or directories
- **cp**: copy files
- **rm/rmdir**: remove files or directories.
- **cp -r**: copy directories

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Tools to improve your storage layout

- Best thing is to have a good naming scheme and directory structure from the start.
- But there'll always be some tuning, so you need to be able to do basic stuff such as:

- **mv**: move or rename files or directories
- **cp**: copy files
- **rm/rmdir**: remove files or directories.
- **cp -r**: copy directories

- Remember linux/unix command line does not have a trash can or undo!

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Packing things up

- Done with a project, or don't need the separate files anymore?
- Delete unnecessary files and pack it up!

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Packing things up

- Done with a project, or don't need the separate files anymore?
- Delete unnecessary files and pack it up!
- **tar:** pack together file into a single file

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Packing things up

- Done with a project, or don't need the separate files anymore?
- Delete unnecessary files and pack it up!
- **tar:** pack together file into a single file

```
$ tar cf previous.tar previous/
$ rm -rf previous/ # Caution: Removes entire directory tree!
```

Packing things up

- Done with a project, or don't need the separate files anymore?
- Delete unnecessary files and pack it up!
- **tar:** pack together file into a single file

```
$ tar cf previous.tar previous/
$ rm -rf previous/ # Caution: Removes entire directory tree!
```

- **gzip/bzip2:** compress file(s)

Packing things up

- Done with a project, or don't need the separate files anymore?
- Delete unnecessary files and pack it up!
- **tar:** pack together file into a single file

```
$ tar cf previous.tar previous/
$ rm -rf previous/ # Caution: Removes entire directory tree!
```

- **gzip/bzip2:** compress file(s)

```
$ ls -l previous.tar
-rw-r--r-- 1 rzon scinet 54292480 May 25 10:49 previous.tar
$ gzip previous.tar
$ ls -l previous.tar*
-rw-r--r-- 1 rzon scinet 41317890 May 25 10:49 previous.tar.gz
```

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Packing things up

- Done with a project, or don't need the separate files anymore?
- Delete unnecessary files and pack it up!
- **tar:** pack together file into a single file

```
$ tar cf previous.tar previous/
$ rm -rf previous/ # Caution: Removes entire directory tree!
```

- **gzip/bzip2:** compress file(s)

```
$ ls -l previous.tar
-rw-r--r-- 1 rzon scinet 54292480 May 25 10:49 previous.tar
$ gzip previous.tar
$ ls -l previous.tar*
-rw-r--r-- 1 rzon scinet 41317890 May 25 10:49 previous.tar.gz
```

- **gunzip/bunzip2:** decompress file(s)

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Keeping inventory

- **ish:** Create an inventory of stuff in your directories or tarballs

Keeping inventory

- **ish:** Create an inventory of stuff in your directories or tarballs

```
$ ish index previous.tar.gz
```

Keeping inventory

- **ish:** Create an inventory of stuff in your directories or tarballs

```
$ ish index previous.tar.gz
```

- Created `previous.tar.gz.igz` in `~/.ish_register/`

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Keeping inventory

- **ish:** Create an inventory of stuff in your directories or tarballs

```
$ ish index previous.tar.gz
```

- Created `previous.tar.gz.igz` in `~/.ish_register/`
- After that you can do (without needing the tar)

Keeping inventory

- **ish:** Create an inventory of stuff in your directories or tarballs

```
$ ish index previous.tar.gz
```

- Created `previous.tar.gz.igz` in `~/.ish_register/`
- After that you can do (without needing the tar)

```
$ ish previous.tar.gz.igz
ish 0.997
Ramses van Zon - SciNet/Toronto/Canada/Apr 8, 2014
[ish]previous.tar.gz.igz> ls
previous/
[ish]previous.tar.gz.igz> find *.sh
previous/analysis1.sh   previous/analysis2.sh   previous/submit
[ish]previous.tar.gz.igz> du
:
```

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Keeping inventory

- **ish:** Create an inventory of stuff in your directories or tarballs

```
$ ish index previous.tar.gz
```

- Created `previous.tar.gz.igz` in `~/.ish_register/`
- After that you can do (without needing the tar)

```
$ ish previous.tar.gz.igz
ish 0.997
Ramses van Zon - SciNet/Toronto/Canada/Apr 8, 2014
[ish]previous.tar.gz.igz> ls
previous/
[ish]previous.tar.gz.igz> find *.sh
previous/analysis1.sh   previous/analysis2.sh   previous/submit
[ish]previous.tar.gz.igz> du
:
```

- More info: ish page on the SciNet wiki
- Get it from: `https://github.com/vanzonr/ish`

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Section 5

Version Control

Version Control

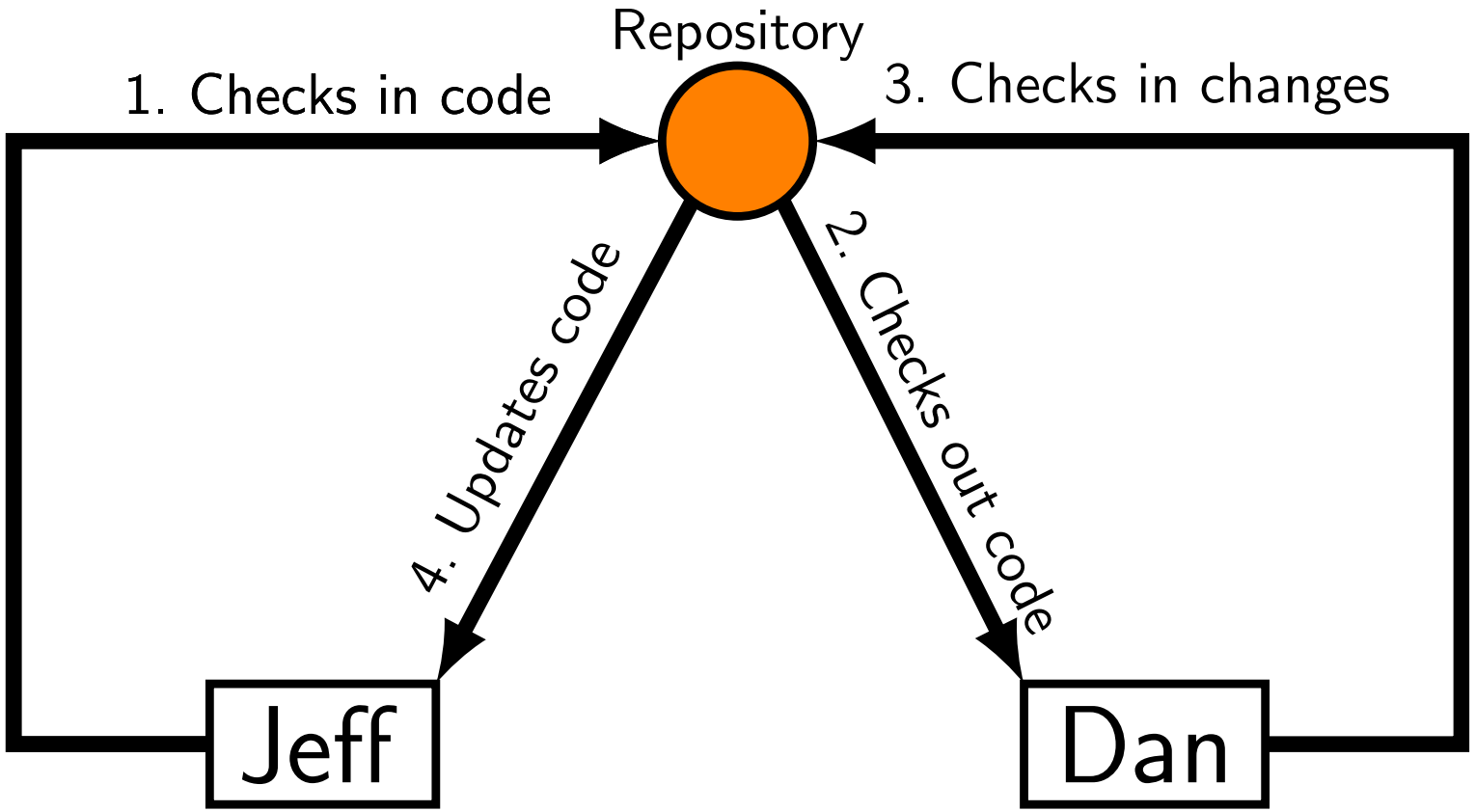
What is it?

- A tool for managing changes in a set of files.
- Figuring out who broke what where and when.
- Synchronize data, ensure data integrity, ...

Why Do it?

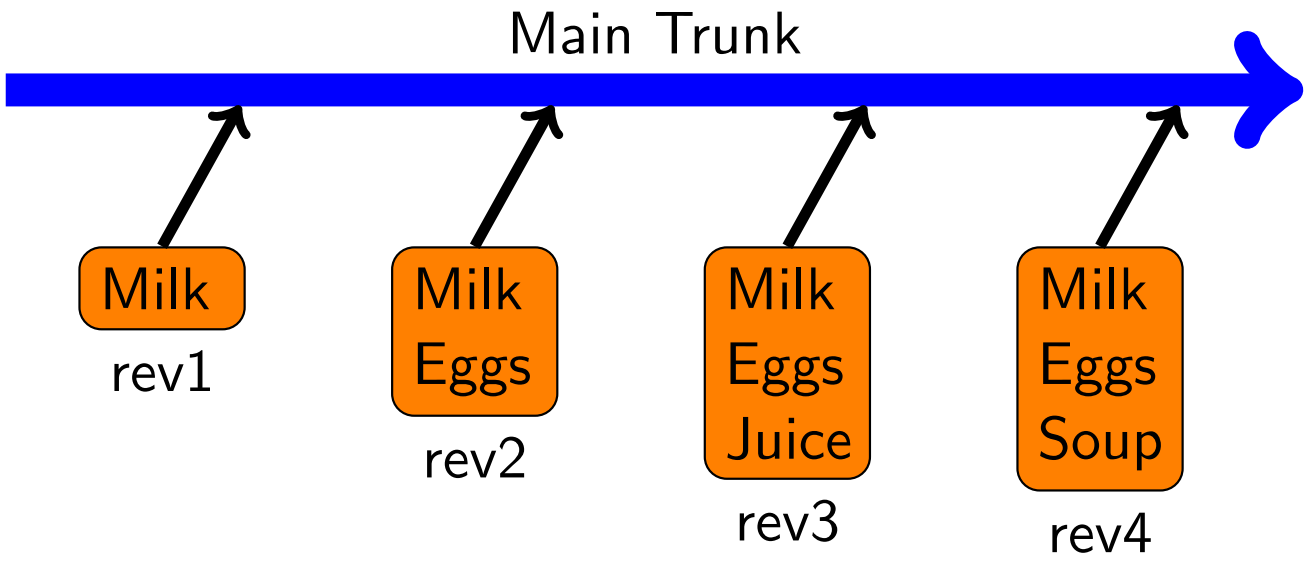
- Collaboration
- Organization
- Track Changes
- Faster Development
- Reduce Errors
- Reproducibility

How does version control work?

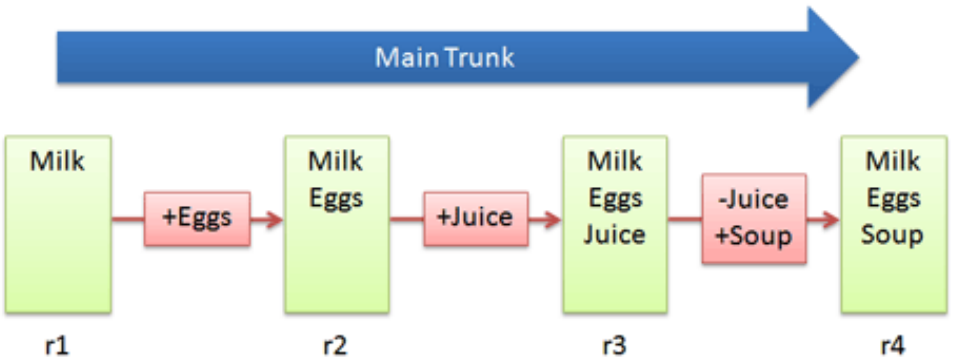


SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Basic Checkins

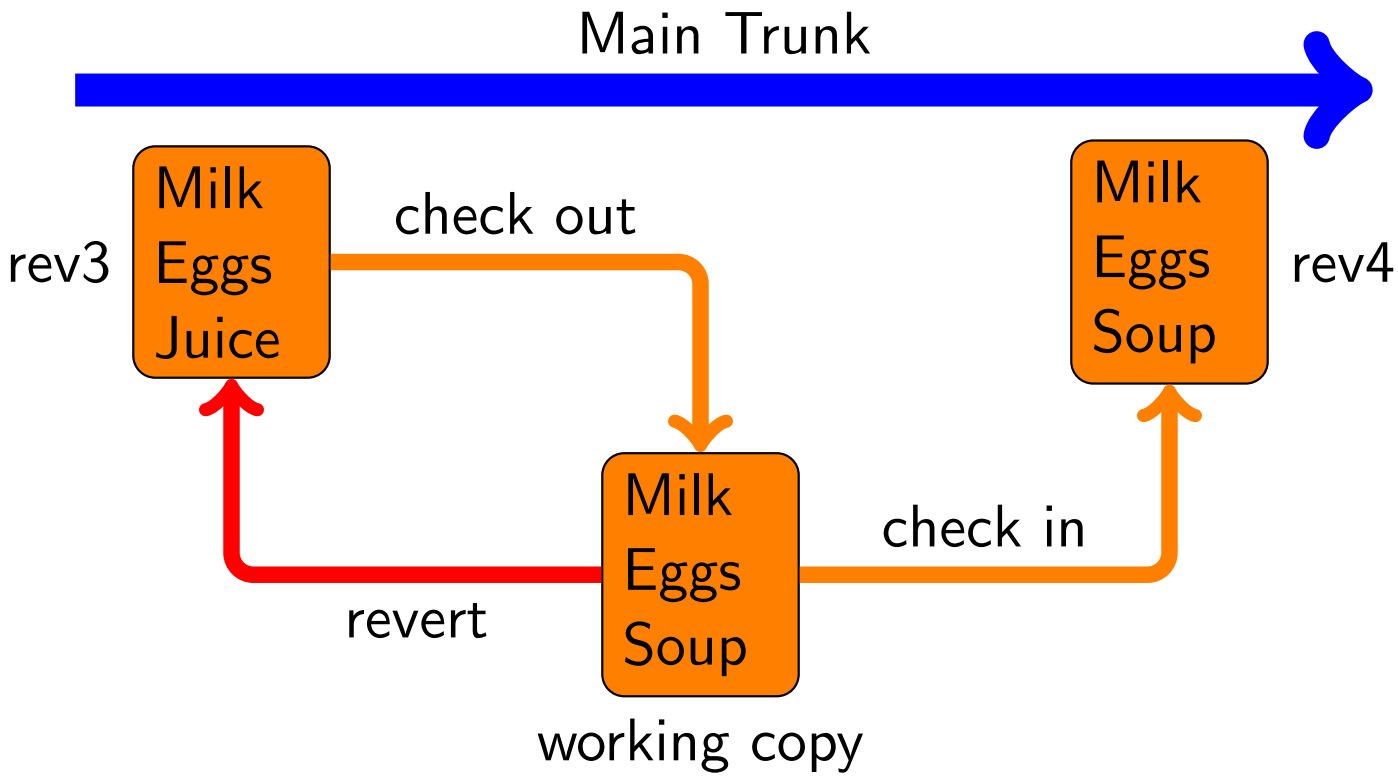


Basic Diffs



SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Checkout and edit



What VC System to Use?

Software

- Open Source
  - Subversion, CVS, RCS
  - **Git**, **Mercurial**, Bazaar
- Commercial
  - Perforce, ClearCase
- Web-based
  - github, bitbucket, ...

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Version Control: git

There are many types and approaches to version control; eg. git, hg, svn, cvs, ...

There are a few main things you need to know how to do, to get started with a VC-repo:

* Setup the VC system on your computer.
- Initialize a repository.
- Commit files to the repository.
- Delete files from the repository.
- Where to find more information.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Version Control: git

There are many types and approaches to version control; eg. git, hg, svn, cvs, ...

There are a few main things you need to know how to do, to get started with a VC-repo:

* Setup the VC system on your computer.
- Initialize a repository.
- Commit files to the repository.
- Delete files from the repository.
- Where to find more information.

eg. git

* Linux
  - `yum/.../apt-get install git`
* MacOS
  - Xcode
  - fink/macports/homebrew
  - git OSX installer
* Windows: MobaXterm
  - `apt-get install git`

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Some Operations with git

Initialize a repository

```
$ git init
```

Commit files to the repository

```
$ git add <files>
$ git commit -m '<commit message>'
```

Delete files from the repository

```
$ git rm <files>
```

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Git Annex

- git not particularly efficient for large files
- no partial checkout of specific files
- in terms, of object storage, use git to manage *metadata*, not file contents

Git Annex

- git not particularly efficient for large files
- no partial checkout of specific files
- in terms, of object storage, use git to manage *metadata*, not file contents

git-Annex

➠ **git-annex** allows managing files with git, without checking the file contents into git

`https://git-annex.branchable.com/`

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

git annex

git-annex tracks files with git, without storing their contents into git

- `git annex add`
  - moves file within git annex objects directory
  - names file according to its content hash (e.g. SHA256)
  - makes file read-only
  - creates symbolic link to the file with original filename
- `git commit`

  commits symbolic link to file, not file contents
- `git annex get`
  - transfers file contents from other repositories
  - uses rsync/wget/over ssh/http/...

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Version Control: more information

There are many other things that can be done with git:

- Review differences between files in different commits.
- Go back to a previous version of the code.
- Branch the code to add new and wonderful features.
- Reconcile different branches of the code.

For a very extensive tutorial, go here:
`http://www.vogella.com/tutorials/Git/article.html`

Web-based options:

- GitHub: `https://github.com/`
- Bitbucket: `https://bitbucket.org`

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Section 7

File Formats

File Formats

In this section, we will discuss the following topics:

- Handling data efficiently; staying organized.
- Storage formats.
- Handling metadata.
- NetCDF.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

File organization

How should your files be organized on disk?

- Human-interpretable filenames lose their charm after a few dozen files (or after a few months pass). Don't use filenames to store (all) run information.
- Avoid using a flat directory structure (no sub-directories). Organize your data in a sensible directory tree.
- If you're doing many runs with many varied parameters, consider using a database to store the filenames of your runs, with associated run metadata.
- Rigorously maintained meta-data (data about the data) is essential.
- Back up your data, especially your metadata or database.

Take-home message: keep your data well-organized.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Storage formats: ASCII

The storage format we all start with is ASCII: American Standard Code for Information Interchange. It's our old friend, plain text:

- Pros:
  - Human Readable.
  - Portable (architecture independent).
- Cons:
  - Inefficient Storage.
  - Precision is lost for floats.
  - Slow to Read/Write (conversions).
  - (Embarrassing.)

Our old friend has done us well, but there are better storage options we can use.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Storage formats: binary

Native binary is the format in which the data is stored in memory:

- Pros:
  - Efficient Storage (256 x floats @4bytes takes 1024 bytes).
  - Efficient (fast) Read/Write (native, no conversion is needed).
- Cons:
  - Not human readable.
  - Have to know the format it's stored in, or else you'll have to reverse-engineer the file format to read it.
  - Not necessarily portable between systems (Endianness).

In terms of speed and file size, there's no contest.
Writing 128 million doubles (128M) to storage:

Format	/scratch (GPFS)	/dev/shm (RAM)	/tmp (disk)
ASCII	173s	174s	260s
Binary	6s	1s !!!	20s

Metadata

But what about that metadata? What is it?

- Metadata is the data about the data. Meaning information that lets you make sense of the data.
- It can (and should) include just about any and all information about how the data was created:
  - what parameters were used in the run?
  - where it was run, when it was run.
  - the version of the code used to perform the run, compiler used to create the code, compiler flags.
  - and anything else that might or not be useful.
- If you're not sure if that bit information should be kept as metadata, then keep it. You never know what information might be needed in the future.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Standard formats

What's the best way to save our metadata? There are several standard file formats which *combine* the metadata with the data:

- CGNS (CFD General Notation System)
- IGES/STEP (CAD Geometry)
- HDF5 (Hierarchical Data Format)
- NetCDF (Network Common Data Form)
- disciplineX version

What are the benefits?

- Most are provided as libraries.
- Self-describing (metadata is embedded with the data).
- Many are binary agnostic, so portable.
- Many support Parallel I/O and native FS support.
- Broader tool support (visualization, etc.)

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Example: netCDF

We are going to focus on netCDF, which is a commonly-used format. What is it?

- Stands for **net**work **C**ommon **D**ata **F**orm.
- Not compatible with NASA's CDF format.
- Used for array-oriented scientific data and metadata format.
- Stores in a binary form, so relatively efficient.
- Though it's in binary, it uses a common output format so different types of machines can share files.
- Self-describing, direct access, appendable.
- Many many wrappers to the API (C, C++, Fortran, Python, ...).

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

NetCDF classic data model

The original netCDF data model contains three entry types:

- Variables: N-dimensional arrays of data, of type char, byte, short, int, float, double.
- Dimensions: these describe the axes of the data arrays. A dimension has a name and a length.
- Attributes: Notes and supplementary information. These are scalar values or 1D arrays.
  - Attributes can be global, or apply to just a dimension or variable.
  - A good place to stick your miscellaneous metadata.
  - Units are a particularly good form of metadata.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

NetCDF conventions

A quick note about netCDF conventions:

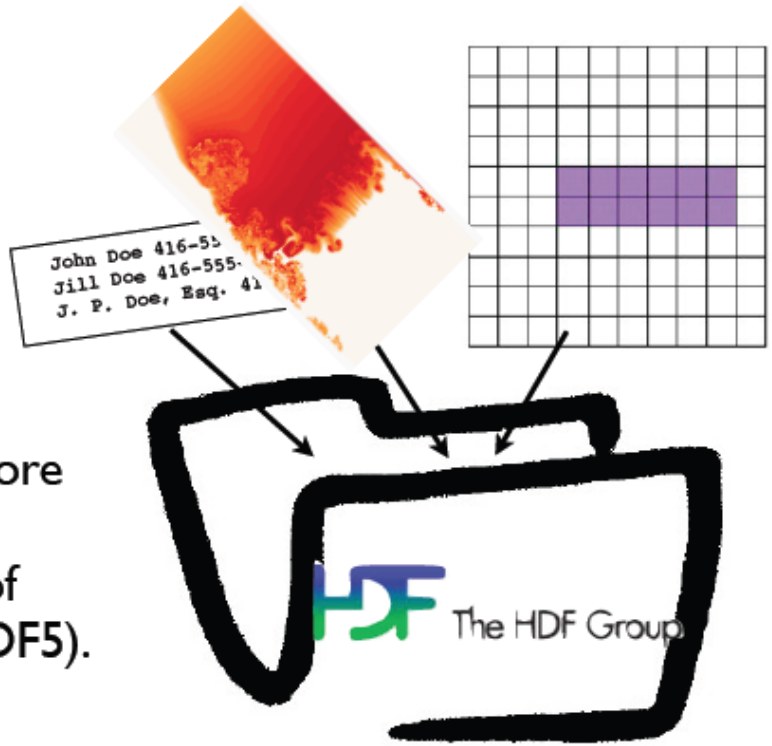
- There are lists of conventions that you can follow for variable names, unit names ("cm", "centimetre", "centimeter"), *etc*.
- If you are planning for interoperability with other codes, this is the way to go.
- Codes expecting data following, say, CF (Climate and Forcast) conventions for geophysics should use that convention.
- `www.unidata.ucar.edu/software/netcdf/conventions.html`

Make life easier for yourself and your collaborators: use the standard conventions.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

HDF5

- HDF5 is also self-describing file format and set of libraries
- Unlike NetCDF, much more general; can shove almost any type of data in there



HDF5

Much more general, and more low-level than NetCDF.
(In fact, newest version of NetCDF implemented in HDF5).
Pro: *can* do more!
Con: **have** to do more.

Summary: File Formats

Things to remember from this section:

- Use a binary format to store you data, not ASCII.
- It's a good practise to make your data "self-describing", meaning store your metadata with your data in the same file.
- NetCDF and HDF5 are commonly used formats to store data that has many useful features.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Section 8

Storage Limits on SciNet

Storage Limits at SciNet

location	quota	block-size	time-limit	backup	dev	comp
$HOME	50GB	256kB	none	yes	rw	ro
$SCRATCH	20TB/1M	4MB	3 months	no	rw	rw
HPSS	2TB	tape-backed	none	maybe	no	no

- Compute nodes do not have local hard drives.
- home and scratch are both part of the GPFS file system.
- GPFS is a high-performance file system which provides rapid reads and writes to large data sets in parallel from many nodes.
- Performs poorly accessing data sets which consist of many, small files.
- Avoid reading and writing lots of small amounts of data to disk.
- Many small files on the system would waste space and would be slower to access, read and write.

http://wiki.scinethpc.ca/wiki/index.php/Data_Management

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Moving large data

Moving less than 10GB through the login nodes

- Only login nodes visible from outside SciNet (1Gb/s link).
- Use scp or rsync.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Moving large data

Moving less than 10GB through the login nodes

- Only login nodes visible from outside SciNet (1Gb/s link).
- Use scp or rsync.

Moving more than 10GB through the datamover1 node

- Can do this from the datamover1 node (10Gb/s link).
- From any SciNet node, ssh to datamover1.
- Transfers must originate from datamover1.
- Your machine must be reachable from the outside.
- Easier (if you do this often): use Globus, a web-based tool for data transfer.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Moving large data

Moving less than 10GB through the login nodes

- Only login nodes visible from outside SciNet (1Gb/s link).
- Use scp or rsync.

Moving more than 10GB through the datamover1 node

- Can do this from the datamover1 node (10Gb/s link).
- From any SciNet node, ssh to datamover1.
- Transfers must originate from datamover1.
- Your machine must be reachable from the outside.
- Easier (if you do this often): use Globus, a web-based tool for data transfer.

Moving data to HPSS

- HPSS is a tape-based storage solution.
- 2TB per group by default (more with special allocation).

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Section 9

Data Management

Data Management: monitoring

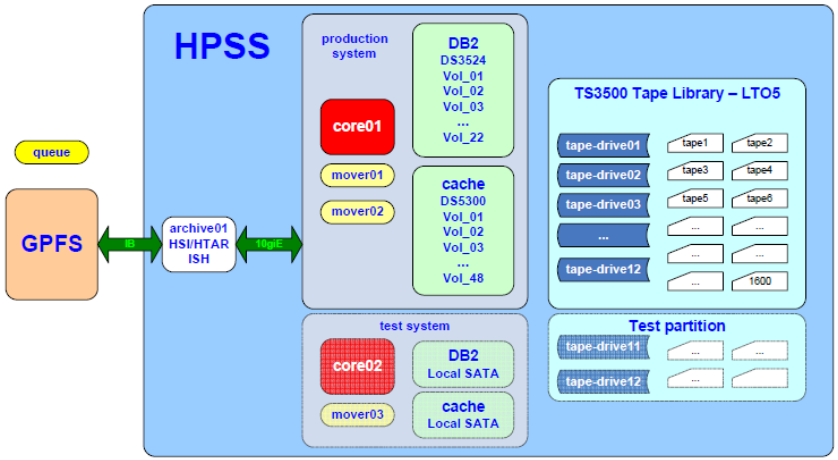
Most HPC systems, have *quotas* in storage resources (SciNet: 1M files, 20TB)

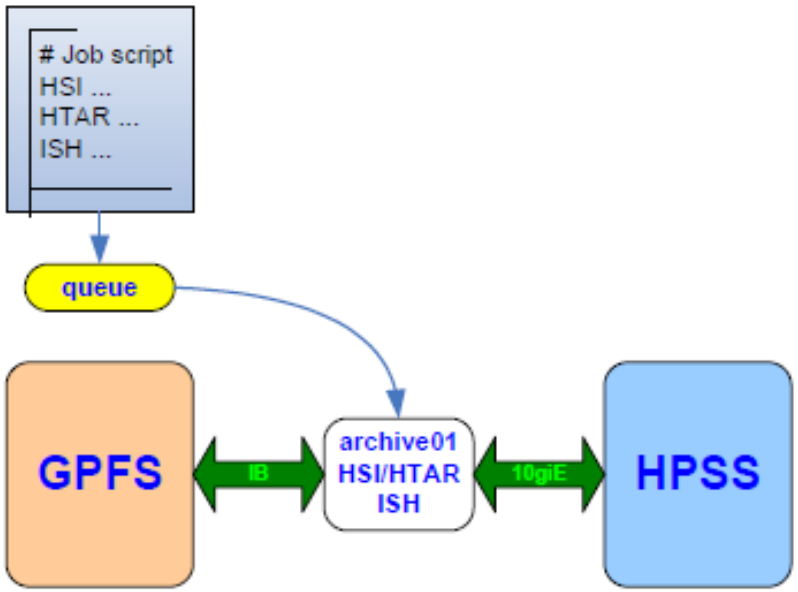
- Minimize use of filesystem commands like `ls -l` and `du`.
- Regularly check your disk usage using /scinet/gpc/bin6/diskUsage.
- Warning signs which should prompt careful consideration:
  - More than 100,000 files in your space
  - Average file size less than 100 MB
- Remember to distinguish: Analysis, Required and By-Product data.

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

HPSS, tapes & archival resources

- most HPC systems, include a High Performance Storage System
- tape-backed hierarchical storage system that provides a significant storage resource





SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Moving data between different machines

- To move your data between machines, `scp` will do.
- Transfer will be faster if you have compressed and tarred files.
- Still, scp is not always the fastest (single stream), may time out, etc.
- An easier and more robust way is to use Globus.
- See
  `https://www.computecanada.ca/research-portal/globus-portal`
  `https://globus.computecanada.ca/SignIn`

SciNet
ADVANCED RESEARCH COMPUTING at the UNIVERSITY OF TORONTO

Moving data between different machines: Globus