Software infrastructure and remote sites

Petr Chaloupka
Nuclear Physics Institute ASCR, Prague

STAR regional meeting

Dubna, Russia 11/21/2003

Where to go for help and informations

- **Main SOFI and help pages:**

  `www.star.bnl.gov/STAR/comp/sofi/ , .../STAR/comp/train/`

- **RCF pages**

  `www.rhic.bnl.gov/RCF`

- **Hypernews fora.**
  - STARSOFT and SOFI forum.
  - Fast help from people who know the answer.

- **RCF Trouble Ticket System**
  - For serious problems with RCF like:

    bad NFS disks, strangely behaving nodes, login problems, forgotten passwords.
  - Nothing Jerome or any other person from STAR can do about it.
  - Accessible trough web at: `www.rhic.bnl.gov/RCF/Organization/Contacts.shtml`

Overview of the RHIC computing facility

- **General Computing environment**
  - For interactive computing tasks not related to data analysis.
  - IMAP mail server `rcf.rhic.bnl.gov`
  - Interactive, Sun based server `rcf2.rhic.bnl.gov`
- **Data analysis facility**
  - For data mining and related tasks.
  - NFS Sun servers.
  - StorageTek tape libraries managed by HPSS.
  - Solaris machines `rmine601-605`.
  - Large farm of Intel based Linux machines:
    - Reconstruction Farm (CRS) - for reconstruction, not available to public.
    - Analysis Farm (CAS) - for analysis of reconstructed data .

      STAR experiment can use ONLY nodes `rcas6xxx` !

Acessing RCF from outside of BNL

- **Through gateways only.**
  - Since September 2003, improved security with Kerberos 5 authentication system.
  - Allows to have only single password for all systems (including AFS).

- **For remote work and data analysis:**
  - Using **`ssh`**.

    `rssh.rhic.bnl.gov`
  - From here log into other machines inside BNL: rcas, rmine, rcf2, ...
  - Or use `rterm` utility - automatically selects optimal rcas machine.

- **For data transfers:**

  `rftpexp.rhic.bnl.gov`
  - Various possible ways how to transfer data in and out of BNL ...

Transferring data from and to BNL

- **Through rftpexp.rhic.bnl.gov**
- **Supported programs:**
  - **`scp`** - encrypts data => slow. Should be used to transfer small (1-10KB) file of sensitive data:

    ```
    scp local_file username@rftpexp.rhic.bnl.gov:remote_file
    username@rftpexp.rhic.bnl.gov:remote_file local_file
    ```

  - **`sftp`** - works under ssh2, also encrypts data.
  - Ordinary ftp is not supported for security reasons.
  - **`bbftp`** - big block ftp-like program. Supports parallel tcp streams - significant increase in speed. Does NOT encrypt data. Suitable for large nonsensitive data.
  - **`rsync`** through ssh. Encrypts data. Compares source and target, and transfers only differences. Good for updating.

    ```
    rsync 'options' -e ssh 'username'@rftpexp.rhic.bnl.gov:'sourcefile'
          'destination'
    ```

Data storage and location

- **NFS**
  - Home directories and local software equipment
  - pwg and scratch disks at `/star/dataxx`

    `AAAREADME` file on each disks tells it's purpose - your data could be DELETED, if on wrong disk !

- **HPSS - High Performance Storage System**
  - High volume tape storage, mainly physics data.
  - Users are encouraged to archive here their data.
  - Production and raw data are not directly accessible for ordinary users.
  - They can be retrieved to NFS area by "data carousel" utility.

- **AFS - Andrew File System**
  - Allows file sharing over the Internet, providing security and caching.
  - Mainly STAR software and libraries => allows working on remote sites.
  - Each user has his space in the AFS area in `/afs/rhic/star/users/'username'`.

Retrieving data from HPSS

- **First, you have to find them**
  - FileCatalog
  - Growing importance of the catalog.

    We have lots of data, but there is WAY MORE to come!
    See Jerome's presentation from 2003 collaboration meeting.
    Also: `www.star.bnl.gov/STAR/comp/sofi/FileCatalog/`

- **Retrievng from HPSS**
  - Using **data carousel** script.
  - Basic use:

    ```
    hpss_user.pl 'source file' 'target directory'
    hpss_user.pl -f file.list
    ```

  - Here is a good description:

    `www.star.bnl.gov/STAR/comp/sofi/carousel/data_carousel.html`
  - The input file list can easily be generated by a script at STAR web page, also the results of the retrieving can be found there (next slide).

Web-based tools and utilities

There is a lot of useful tools at STAR pages:

- **RunLog Browser:** `online.star.bnl.gov/RunLog2003/`
- **Production Data Browser:** `www.star.bnl.gov/devcgi/dbDataSetQuery.pl`
- **MC Data Browser:** `www.star.bnl.gov/devcgi/dbMCData.pl`
- **Offline Software Guide:** `www.star.bnl.gov/cgi-bin/prod/swguide.pl`
  - Searching through offline software.
- **Data carousel input file generator:**

  `www.star.bnl.gov/cgi-bin/jerome/genlist.cgi`
  - Easy way to generate input file for data carousel.
- **HPSS account status monitor:**

  `www.star.bnl.gov/cgi-bin/jerome/display_accnt.cgi`
  - Shows status of requests submitted to HPSS.
- **Rcas machines status:** `www.star.bnl.gov/cgi-bin/nova/showMachines.pl`

Orienting in library versions

- **After logging into rcas machine:**

```
----------------------
The present release assignment:
              SL00m    (SL00m)    ROOT_LEVEL 2.25.03
              SL01i    (SL01i)    ROOT_LEVEL 3.02.00 AuAu 130GeV production
              SL01j    (SL01j)    ROOT_LEVEL 3.02.00 updated SL01i with SIMU tag
              SL02c    (SL02c)    ROOT_LEVEL 3.02.07 AuAu200 GeV production
              SL02d    (SL02d)    ROOT_LEVEL 3.02.07 AuAu200 GeV production
              SL02e    (SL02e)    ROOT_LEVEL 3.02.07 AuAu and pp 200GeV production
              SL02g    (SL02g)    ROOT_LEVEL 3.02.07 FTPC side AuAu200 production
              SL02h    (SL02h)    ROOT_LEVEL 3.02.07 only tag exists
              SL02i    (SL02i)    ROOT_LEVEL 3.02.07 daq100 test,dAu200 MC
              SL03a    (SL03a)    ROOT_LEVEL 3.03.09 real dAu 200GeV production
              SL03b    (SL03b)    ROOT_LEVEL 3.03.09 dau200 MC hijing
              SL03d    (SL03d)    ROOT_LEVEL 3.03.09
      old-> SL03e    (SL03e)    ROOT_LEVEL 3.05.04 ppMinBias, ppTrans-1,ppLong-1
      pro-> SL03f    (SL03f)    ROOT_LEVEL 3.05.04 ppMinBias 2001/2002 SVT rerun
      new-> SL03h    (SL03h)    ROOT_LEVEL 3.10.01 dAu and pp data reproduction
      dev-> DEV                 ROOT_LEVEL 3.10.01
     .dev-> .DEV                ROOT_LEVEL 3.99.99
-----------------------------------------------
```

name

version ID

- **Select library:**
  - **`stardev, starnew, starpro, starold`**
  - or change to any other library by **`starver 'version ID'`**
- **You can check by:**

  echo $STAR_LEVEL

Orienting in library versions

Four main versions of libraries:

- **DEV**
  Contains latest development code.
  Upgraded and tested daily.
  Very likely contains bugs => don't use unless absolutely necessary.
- **NEW**
  Relatively stable version.
  Usually couple weeks old.
- **PRO**
  Current production version.
  Fully tested.
  Recommended for use in analysis.
- **OLD**
  Preceding production version.

Description of current releases - please, remember

`www.star.bnl.gov/STAR/comp/ofl/software_releases.html`

Using CVS

- **Newest code is stored in CVS repository.**
  - Three main branches: StRoot, Online, Offline.
- **The repository is in the AFS area**
  - Accessible from any computer with AFS client - need AFS token.
- **Code can be checked in and out at any time**
  - It's stronly recommended to save into Online and Offline branches any important utilities connected with online and offline data production.
  - Check in (upload) files in StRoot is restricted.
- **More information can be found at**

  `www.star.bnl.gov/STAR/comp/train/tut/UsingCvs.html`
- **Basic use:**
  - Check out file: `cvs co 'file or directory name'`
  - Update files in current directory: `cvs update`
  - Option: `-n` only test, don't write

    `-r 'ver'` specify version

Compiling with CONS

- **cons is Perl based replacement for make.**
- **Tutorial:**
  - `http://www.star.bnl.gov/STAR/comp/train/cons/ConsInSTAR.html`
- **Standard use:**
  - `cons ...` Compiles everything in the current directory.
  - `cons +[pattern]...` Compiles all modules matching the pattern.
- **Cons uses parameters - these can be overwritten on the command line.**

  Parameters used for debugging:
  - NODEBUG=yes, set debug flag to -O2 (default -g);
  - DEBUG=value, set debug flag to "value";
- **Strange problems during compilation? Check your library version.**
  - Check last library version used for compilation of your program by:

    `STAR_LEVELS -g`

Simple Example

```
[rcas6007] ~/> mkdir test
[rcas6007] ~/> cd test
[rcas6007] ~/test/> cvs co StRoot/StSvtCalibMaker
cvs checkout: Updating StRoot/StSvtCalibMaker
U StRoot/StSvtCalibMaker/StSvtDriftVelocityMaker.cxx
U StRoot/StSvtCalibMaker/StSvtDriftVelocityMaker.h
U StRoot/StSvtCalibMaker/StSvtPedMaker.cxx
U StRoot/StSvtCalibMaker/StSvtPedMaker.h
[rcas6007] ~/test/> starpro
[rcas6007] ~/test/> cons
```

Create directory.

Retrieve source code from CVS

Change into PRO library
Compile

```
g++ -g -shared -Wl,-Bdynamic -o .rh80_gcc32/obj/StRoot/StSvtCalibMaker/StSvt
CalibMaker.so .rh80_gcc32/obj/StRoot/StSvtCalibMaker/StSvtDriftVelocityMaker
.o .rh80_gcc32/obj/StRoot/StSvtCalibMaker/StSvtPedMaker.o .rh80_gcc32/obj/St
Root/StSvtCalibMaker/StSvtCalibMaker_Cint.o -L/afs/rhic/star/packages/DEV/.r
h80_gcc32/lib
Install .rh80_gcc32/obj/StRoot/StSvtCalibMaker/StSvtCalibMaker.so as .rh80_g
cc32/lib/StSvtCalibMaker.so
[rcas6007] ~/test/>
```

Compilation went well and result, library StSvtCalibMaker.so**. is placed in**

./.rh80_gcc32/lib/StSvtCalibMaker.so

PART II

Remote sites

(setup and possible optimization)

Motivation for working remotely

- **Working in the STAR software environment**
  - using StChain, root4star, staf ...
  - Working with different libraries ...old, new, dev.
- **Running many short sessions during a day (testing and debugging).**
  - need for quick repetitive program editing, compilation and running.
  - need for reasonably quick viewing of results.
- **Use of local computing power.**
- **Speed and stability of the Internet connection to BNL is crucial!**
  - Often insufficient outside of US.

In the beginning

Two main things are necessary for a remote site:

- **To have an access to STAR libraries and software**
  - Usually through the AFS.
- **To setup the STAR environment**
  - Set up environment variables during login.

AFS

Distributed file system over TCP/IP, based on client/server model.

- **Sharing of files over the Internet.**
  - Provides security- authentication and Access Control List.
  - Caching - less vulnerable to Internet problems.
  - All STAR files in RHIC cell: /afs/rhic.bnl.gov
- **When setting up the site.**
  - Free software from `www.openafs.org`
  - All that is needed is the AFS client.
  - AFS module is loaded into Linux kernel.
  - File `/usr/vice/etc/ThisCell has to contain rhic.bnl.gov.`
  - In file `/usr/vice/etc/CellServDB` is list of available AFS cell servers. This list has to contain the `rhic.bnl.gov` cell! If RHIC servers change this file has to be updated! It can be copied from any rcas machine.
  - Configure your cache in `/usr/vice/etc/cacheinfo`. This influences the performance.
  - Make sure your AFS can see through your firewall.

Environment setup

- **The login environment is set by login scripts** `.login, .cshrc`
  - They can be downloaded from BNL:

```
cp /afs/rhic/star/group/templates/cshrc ~/.cshrc
cp /afs/rhic/star/group/templates/login ~/.login
```

- **It's preferable to make own copy of group directory**
  - Copy `/afs/rhic/star/group and` make your `GROUP_DIR` point to it's new location.
  - It's possible, now, to freely change the variable set in `group_env.csh`
  - The most important variables:

    `AFS_RHIC` : top rhic directory path [default = /afs/rhic]

    `OPTSTAR` : Base directory for updates of /usr/XXX or /usr/local/XXX [default = either /opt/star if exists or $XOPTSTAR]. The entire tree $XOPTSTAR may be installed on your local cluster.

    CVSROOT : the STAR CVS repository
- **More informations at:** `www.star.bnl.gov/STAR/comp/train/tut/EnvSetup.html`
- **Subscribe to STAR Offsite facilities HyperNews!**

Possibilities when working remotely

- **Running at PDSF**
  - Great for big batch jobs, not for continuous editing and rerunning.
- **RCAS machines**
  - long response time => impossible to efficiently edit and run programs remotely in BNL.
  - Possible to edit locally, then copy to rcas and run there - very uncomfortable, usually need to copy back the results.
  - RCAS machines are slow.

- **Using local computer with AFS**
  - AFS - long time for loading data into cache (mainly for the first time).
  - MySQL database - loading data in every run (large fraction of time for small programs).
  - Sensitive to problems with Internet connection.
- **Local copy of STAR software and MySQL mirror**

What does the "local copy" mean?

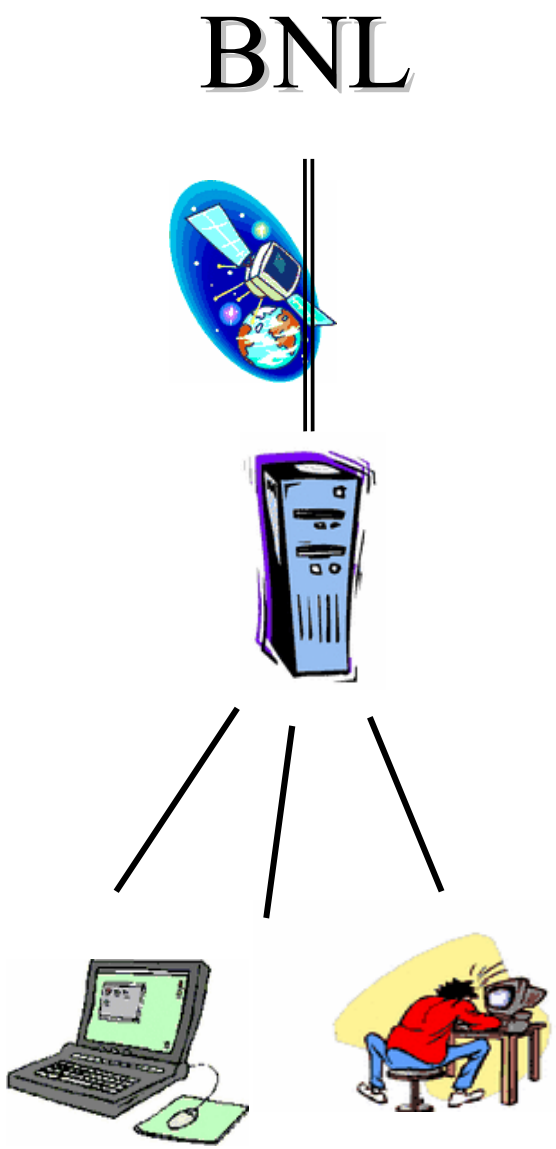
This setup was done in Prague, based on the similar setup done by Marcelo Munhoz at Sao Paulo University.

- **Physical copy of STAR software on local disk**
  - STAR software, which is normally in shared AFS directory `/afs` is copied on local disk: `/localAfsDir`.
  - This includes:
    - root4star: `/localAfsDir/star/ROOT`
    - libraries: `/localAfsDir/star/packages/SL*../dev,/pro`
    - Cernlib: `/localAfsDir/asis/share/cern`

- **STAR environment**
  - Using local copy of STAR login and setup scripts: `/localAfsDir/star/group`
  - Local `.login` is original STAR login altered to set `$AFS_RHIC to /localAfsDir`

- **Access to newest code trough CVS**
  - Altering login script to keep `$CVSROOT` pointing to `/afs/rhic/star/packages/repository`
  - It's still possible to obtain the newest code by `cvs`.

Hardware setup at Prague



- **Internet connection to BNL**
  - Sufficient for large overnight data transfers (10GB/hour using bbftp)
  - Fluctuation in speed
- **Main computing machine**
  - Double processor Pentium IV 2GHz
  - 1 GB memory, 240 GB RAID hard disk.
  - RedHat Linux 7.3
- **Local Internet connection**
  - 100 Mbit/s, stable
- **Local computers**
  - Mainly used as terminals

Why local MySQL database?

- **Connecting and downloading from database far away takes large fraction of time when running short sessions.**

- **Little comparison:**
  Simple script to simulate 5 SVT events -->

  - **With remote database**
    more than **10 minutes**.

  - **With local database**
    about **2 minutes**.

- **Now imagine rerunning it 20 times.**

```
{ //making 5 simulated SVT events
 LoadLibraries();

 StChain *chain  = new StChain("myMainChain");

 //_____________________________________________________
 // GEANT maker
 St_geant_Maker       *geantMk  = new St_geant_Maker("geant");
 char* MainFile = "data/cf1197_05_5940evts.fzd";
 geantMk->SetInputFile(MainFile);

 //_____________________________________________________
  // DB maker
 dbaseMk = new St_db_Maker("db","MySQL:StarDb");
 dbaseMk-> SetDateTime(20021115,000000);

 //_____________________________________________________
 // SVT Db maker
 svtDbMk  = new StSvtDbMaker("svtDb");

 //_____________________________________________________
 // Simulation Maker
 simMkr = new StSvtSimulationMaker("SvtSimu");

 Int_t iInit = chain->Init();
 if (iInit) chain->Fatal(iInit,"on init");

 //
 // Event loop - 5 events
 //
 nevent=5;
 int istat=0,i=1;
 EventLoop: if (i <= nevents && istat!=2)
 {
   chain->Clear();
   istat=chain->Make(i);

   goto EventLoop;
  }
 chain->Finish();
}
```

MySQL server

Get green To Go from M. DePhillips or J. Porter before starting to install!

- **MySQL - software under GNU GPL license**
  - Get correct version from www.mysql.com Must be the same version as in BNL.
- **Note when installing:**
  - The server must be running as slave to the BNL server

  => automatic data update.
  - Has to be set to use GMT time.
- **Server lookup**
  - `~/dbServers.xml` contains the address of your new local server.
  - If local server is down, programs will automatically access other servers.

Pros and cons

- **Pros**
  - Local computers can be comfortably used for computing within STAR software frame work.
  - Quickly accessible database with almost no required maintenance.
  - Significant increase in work speed and efficiency.
  - Insensitivity to Internet speed and quality fluctuations.

- **Cons**
  - Libraries may not be completely up to date.
  - Need of updating - can be automatized using `cron`.
  - Care has to taken of AFS system dependent directories (`@sys` variable).