NCCS Acceptance Tests and Test Harness

NATIONAL CENTER
FOR COMPUTATIONAL SCIENCES

presented by

Arnold Tharrington

National Center for Computational Science – Scientific Computing Division

Oak Ridge National Laboratory

PO Box 2008 MS6008

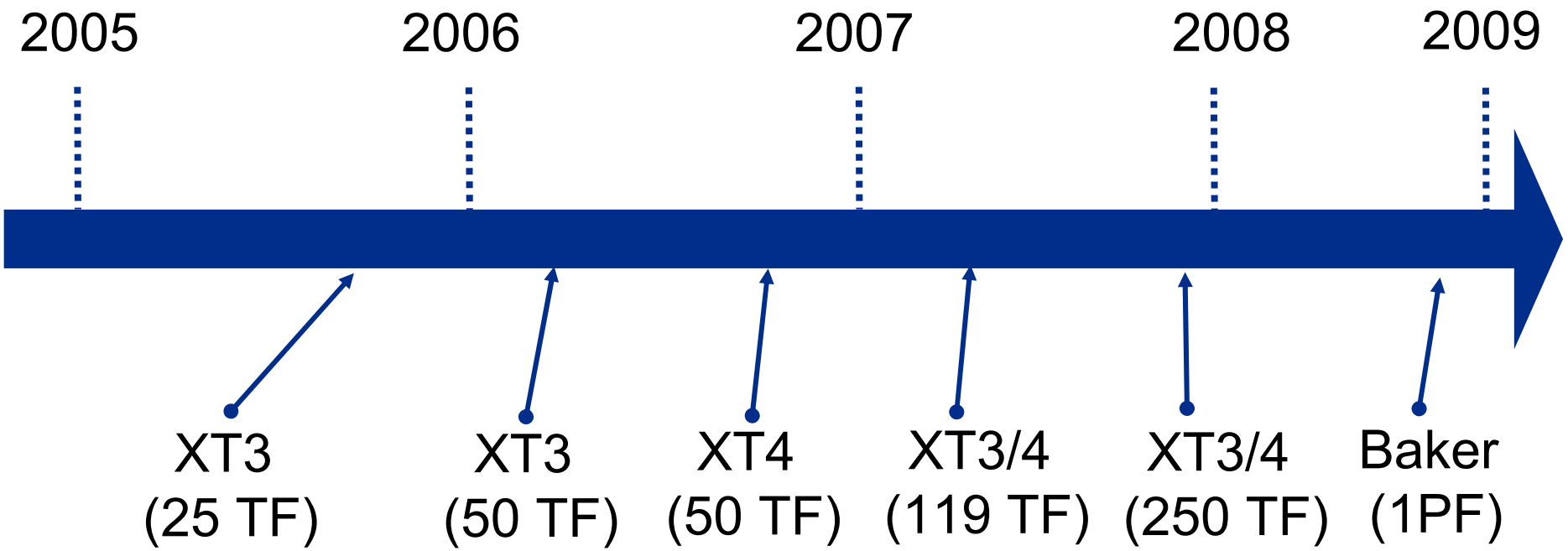
Oak Ridge, TN 37831-6008

Oak Ridge National Laboratory
U.S. Department of Energy

Introduction

- NCCS Roadmap to Petaflop
- Overview of NCCS Jaguar Acceptance Testing
- Description of NCCS Test Harness
- Conclusions and Future Direction of Test Harness

Aggressive Roadmap to Petaflop



Hardware Environment

Jaguar 50T Upgrade	
Type	**Quantity**
Compute Nodes	5,212 AMD Opteron Dual-Core Processors
Compute Threads	10,424
Memory per node	2 GB
Global Disk Space	120 TB
Global Disk Bandwidth	14 GB/s
External Network I/O connections	1 Gb/s Ethernet x 38 and 10 Gb/s Ethernet x 2
Login Nodes	8

Hardware Environment (contd.)

Jaguar 100T Upgrade	
Type	**Quantity**
Compute Nodes	6,296 AMD Opteron Dual-Core Processors
Compute Threads	12,592
Memory per node	2 GB
Global Disk Space	767 TB
Global Disk Bandwidth	41 GB/s
External Network I/O connections	1 Gb/s Ethernet x38 and 10 Gb/s Ethernet x 10
Login Nodes	20

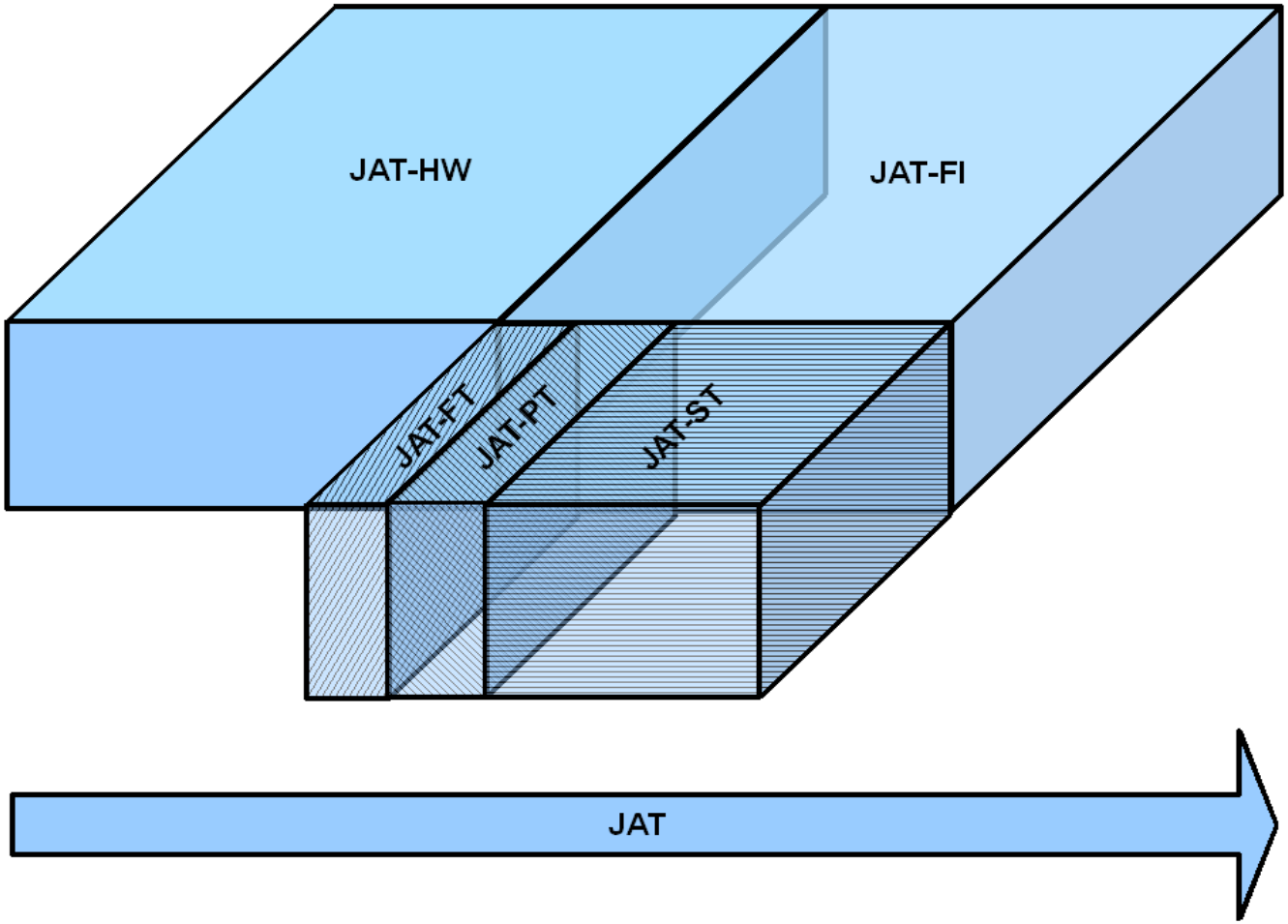
Hardware Environment (contd.)

Jaguar 250T Upgrade	
Type	**Quantity**
Compute Nodes	6,296 AMD Opteron Quad-Core Processors
Compute Threads	25,184
Memory per node	8 GB
Global Disk Space	767 TB
Global Disk Bandwidth	41 GB/s
External Network I/O connections	1 Gb/s Ethernet x N and 10 Gb/s Ethernet x 10
Login Nodes	20

Hardware Environment (contd.)

Baker 1PF System	
Type	**Quantity**
Compute Nodes	22,400 AMD Opteron Quad-Core Processors
Compute Threads	89,600
Memory per node	8 or 16 GB
Global Disk Space	5 to 15 PB
Global Disk Bandwidth	240 GB/s (procured separately)
External Network I/O connections	1 Gb/s Ethernet x 20 and 10 Gb/s Ethernet x 8
Login Nodes	20

General Organization of Jaguar Acceptance Test



Applications

- MPI Tests
  - **Intel MPI Benchmarks (IMB 2.3) formerly known as the Pallus**
  - **Purple Presta 1.2**
  - **MPICH Test Suite**
- Library Functionality Tests
  - **Scalapack**
  - **HDF5**
  - **NetCDF**
  - **FFTW2, FFTW3**

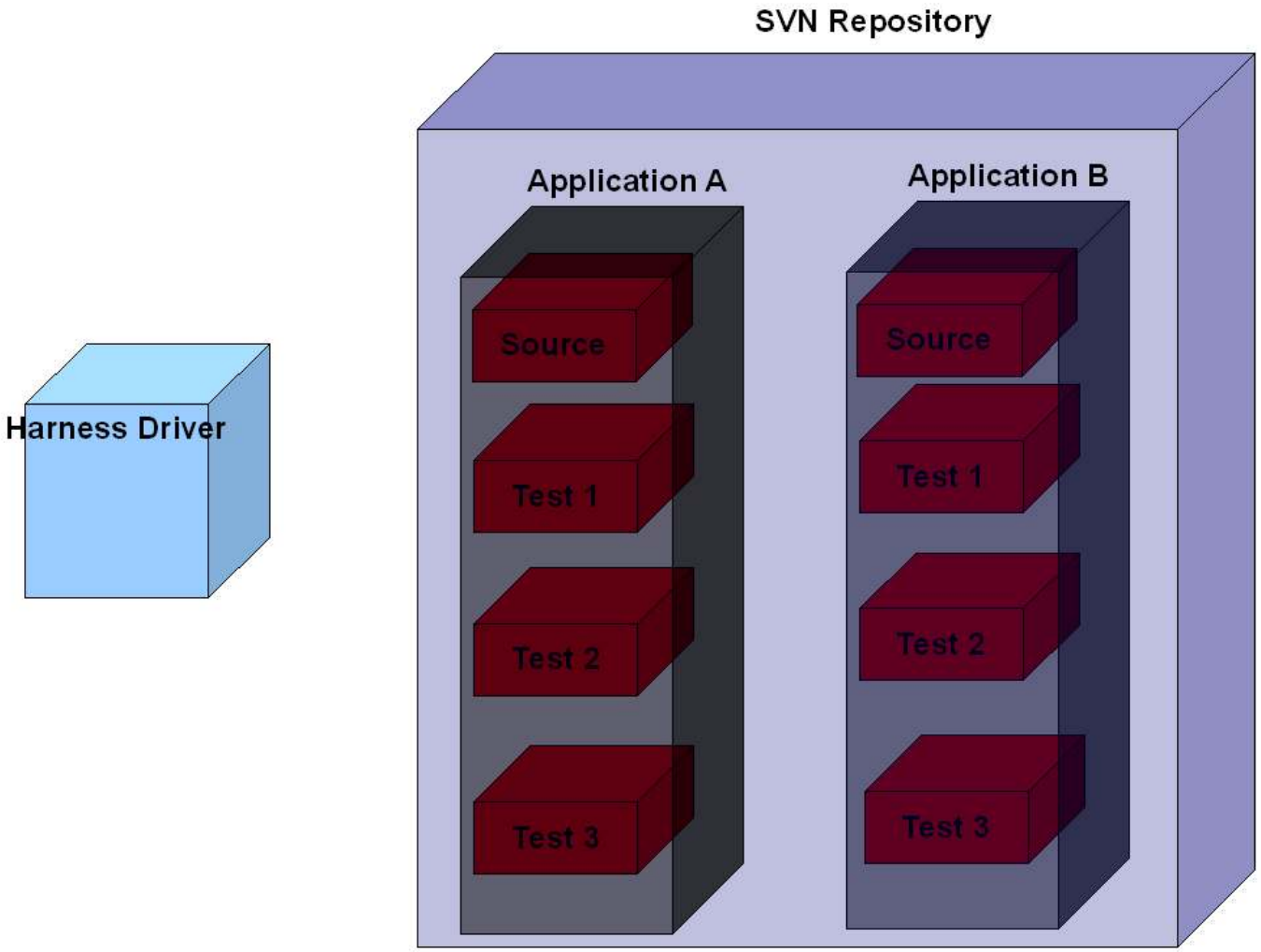
Applications (contd.)

- Scientific Applications
  - **Gyrokinetic Toroidal Code (GTC)**
  - **S3D**
  - **CCSM**
  - **POP**
  - **LSMS**
  - **VH1**
  - **AORSA**
- Torque/Moab Test
  - **Kickstart**

Jaguar Acceptance Test-Final Integration: Logistical Barriers

- Typically between 50 and 100 tests for each JAT-FI part.
- The JAT-FI parts run for sustained times
  - Critical to frequently analyze the results so as to ascertain the pass rate and detect any problems as soon as possible.
- The job load will be such that it reasonably simulates a production environment
- All test results must be archived for auditing
- Single point of contact for running the acceptance tests

Harness Design



Running Harness

- Make Input File
- Define appropriate user environmental variables
- Load harness module
- Run the command “runtests.py”

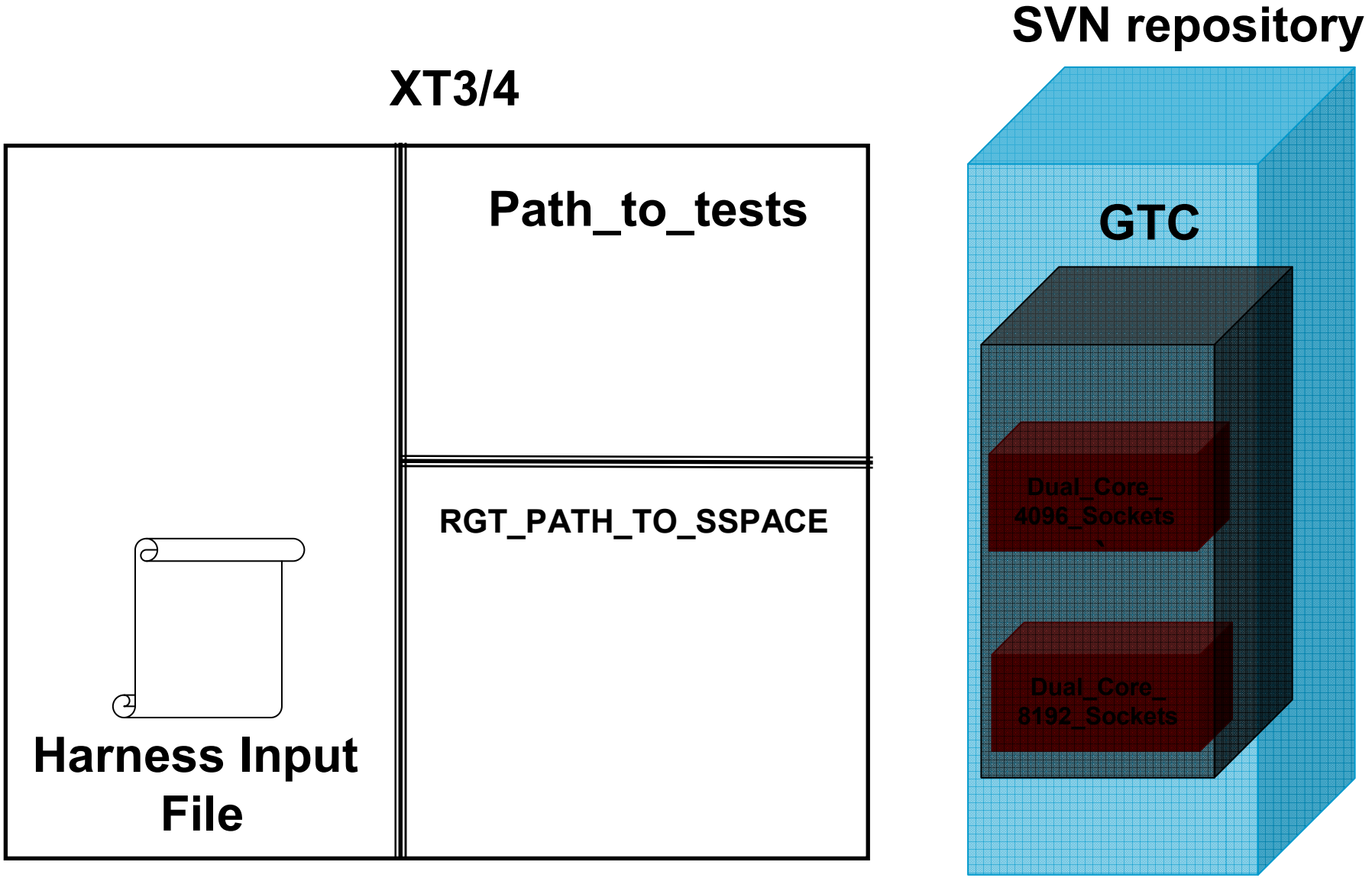
Harness Input File

```
###############################################################################
#  Set the path to the top level of the application directory.                #
#                                                                             #
###############################################################################
Path_to_tests = /lustre/scr144/arnoldt/trial_applications_6

###############################################################################
#      Name of Applications       Name of Tests                               #
#                                                                             #
###############################################################################
Test = IMB_v2.3                   Dual_Core_4096_Sockets

###############################################################################
#      Name of Applications       Name of Tests                               #
#                                                                             #
###############################################################################
Test = GTC                        Dual_Core_4096_Sockets
Test = GTC                        Dual_Core_8192_Sockets

###############################################################################
#      Name of Applications       Name of Tests                               #
#                                                                             #
###############################################################################
Test = MPICH-test_v1.1                Test1

###############################################################################
#  The task to the harness can perform on the tests.                          #
#                                                                             #
# Simply uncomment the appropiate tasks(s).                                   #
###############################################################################

################################################
# Checks of the test from the SVN repository. #
################################################
#Harness_task = check_out_tests

################################################
# Starts the test                             #
################################################
#Harness_task = start_tests

################################################
# Display the status of the tests.            #
################################################
#Harness_task = display_status

################################################
# Stops the tests.                            #
################################################
#Harness_task = stop_tests
```

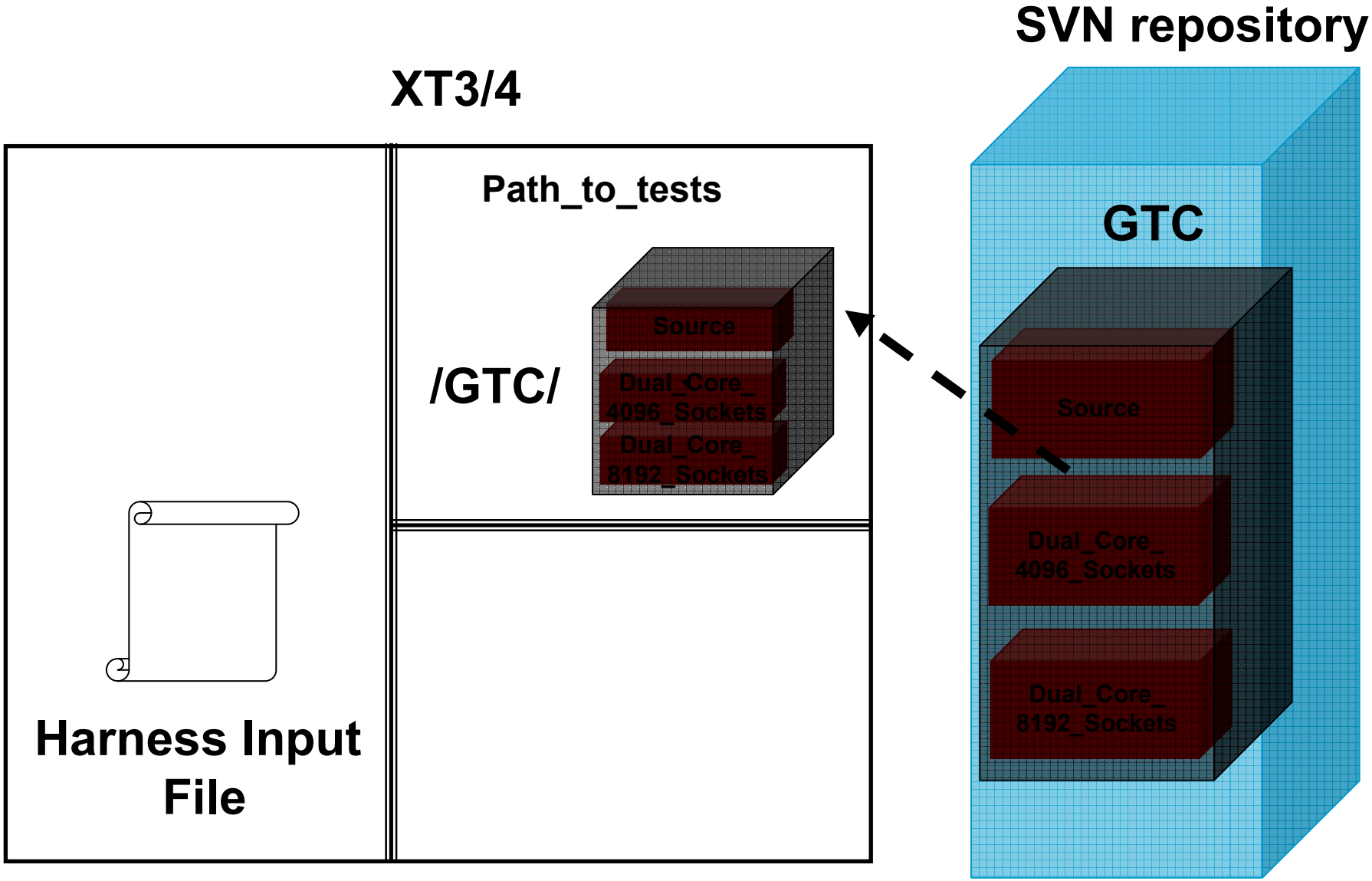
Harness Environmental Variables

```
#! /usr/bin/env bash

#
# Author: Arnold Tharrington
# Email: arnoldt@ornl.gov
# National Center of Computational Science, Scientifc Computing Group.
#

#
# This file defines and sets user specific environmental variables for the test
# harness.
#


# Add or modify as needed to suit your environment.

   #-----------------------------------------------------------------
   # Absoulte path to scratch space location.                      -
   #-----------------------------------------------------------------
   RGT_PATH_TO_SSPACE='/lustre/scr144/arnoldt/trial_scratch_6'
   export RGT_PATH_TO_SSPACE

   #-----------------------------------------------------------------
   # PBS job account id.                                           -
   #-----------------------------------------------------------------
   RGT_PBS_JOB_ACCNT_ID='stf006bf'
   export RGT_PBS_JOB_ACCNT_ID

   #-----------------------------------------------------------------
   # Set the path to this file                                     -
   #-----------------------------------------------------------------
   RGT_ENVIRONMENTAL_FILE='/spin/home/arnoldt/nccs_xt7
_stability_test/AllTests/rgt_environmental_variables.bash.x'
   export RGT_ENVIRONMENTAL_FILE

   #-----------------------------------------------------------------
   # Name of nccs test harness module to load                      -
   #-----------------------------------------------------------------
   RGT_NCCS_TEST_HARNESS_MODULE='nccs_test_harness/0.2'
   export RGT_NCCS_TEST_HARNESS_MODULE
```

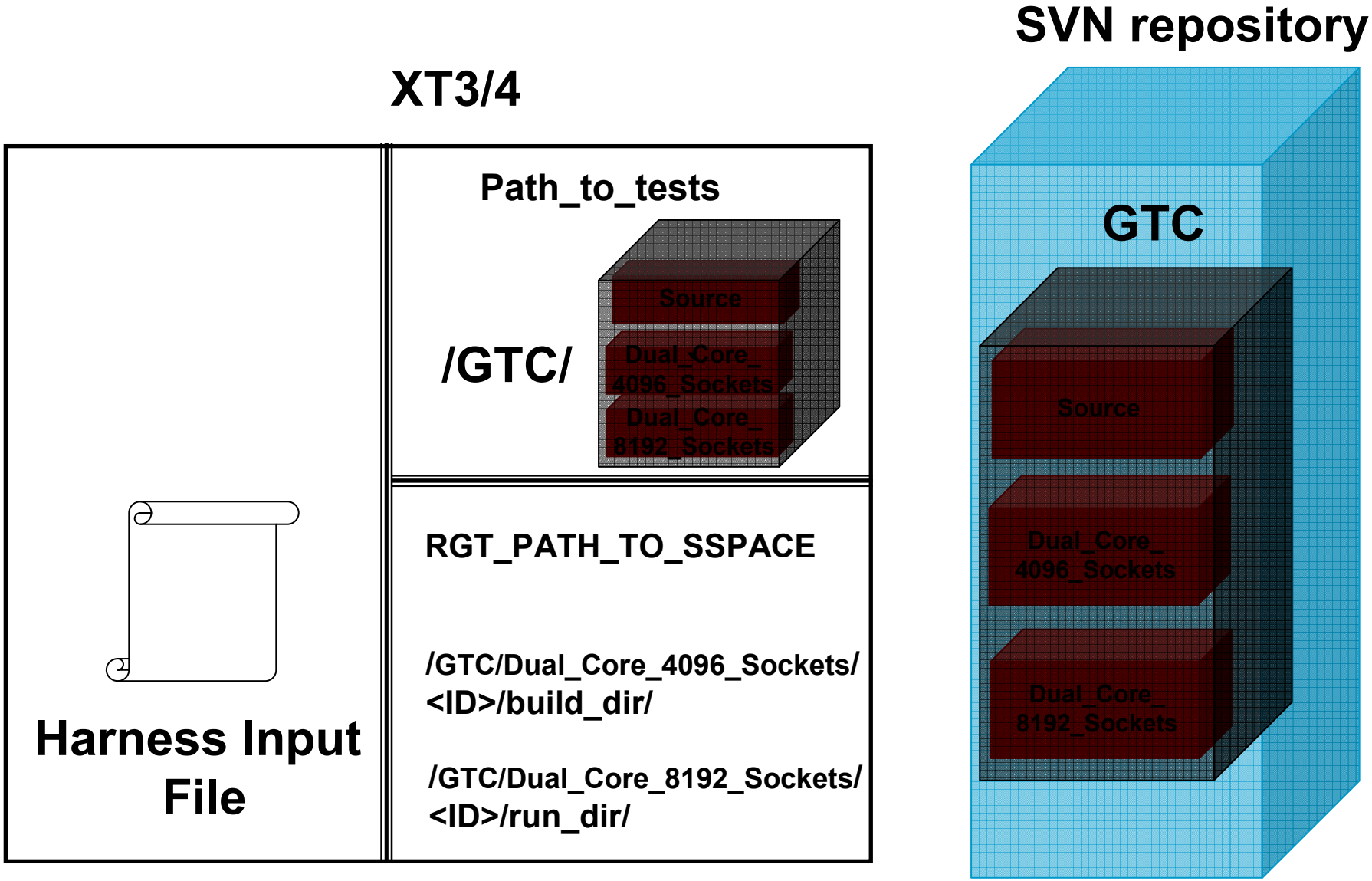
Harness in Action



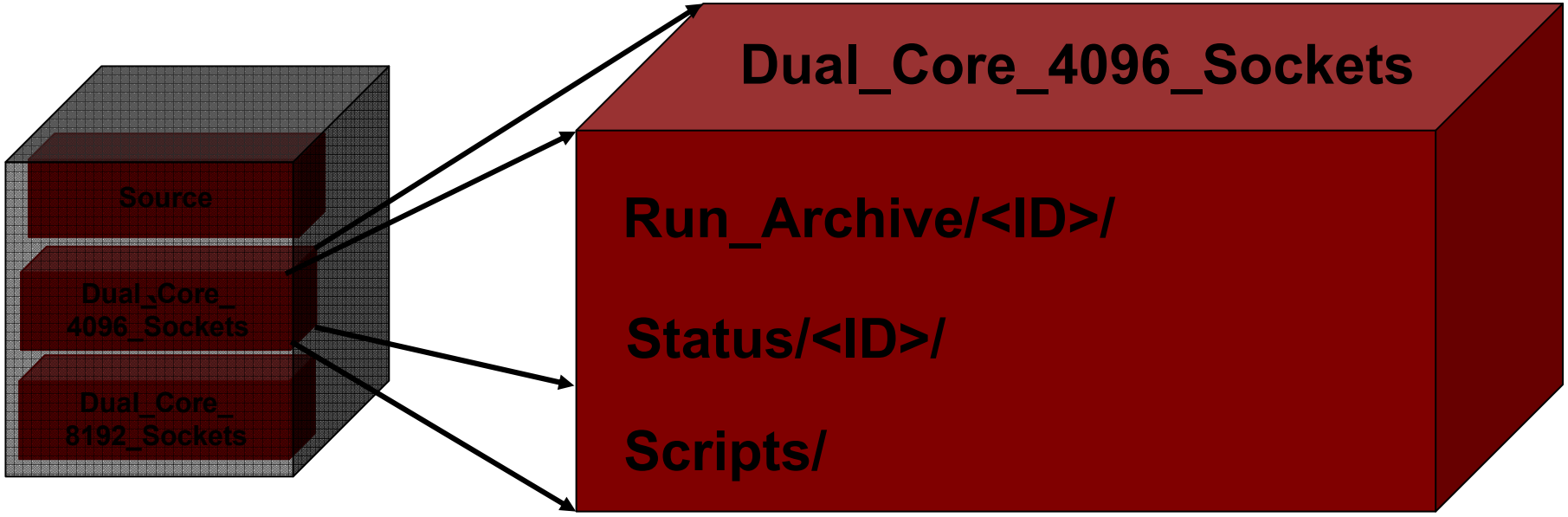
Harness in Action (Checkout a Test)



Harness in Action (Running a Test)



Harness in Action (Running a test contd.)



build_me.x -i <ID> -p <path_to_unique_location>

submit_me.x -i <ID> -p <path_to_unique_location>

check_me.x -i <ID> -p <path_to_unique_location>

Harness in Action (Running a test contd.)

build_me.x -i <ID> -p <path_to_unique_location>

Returns the status of the build.

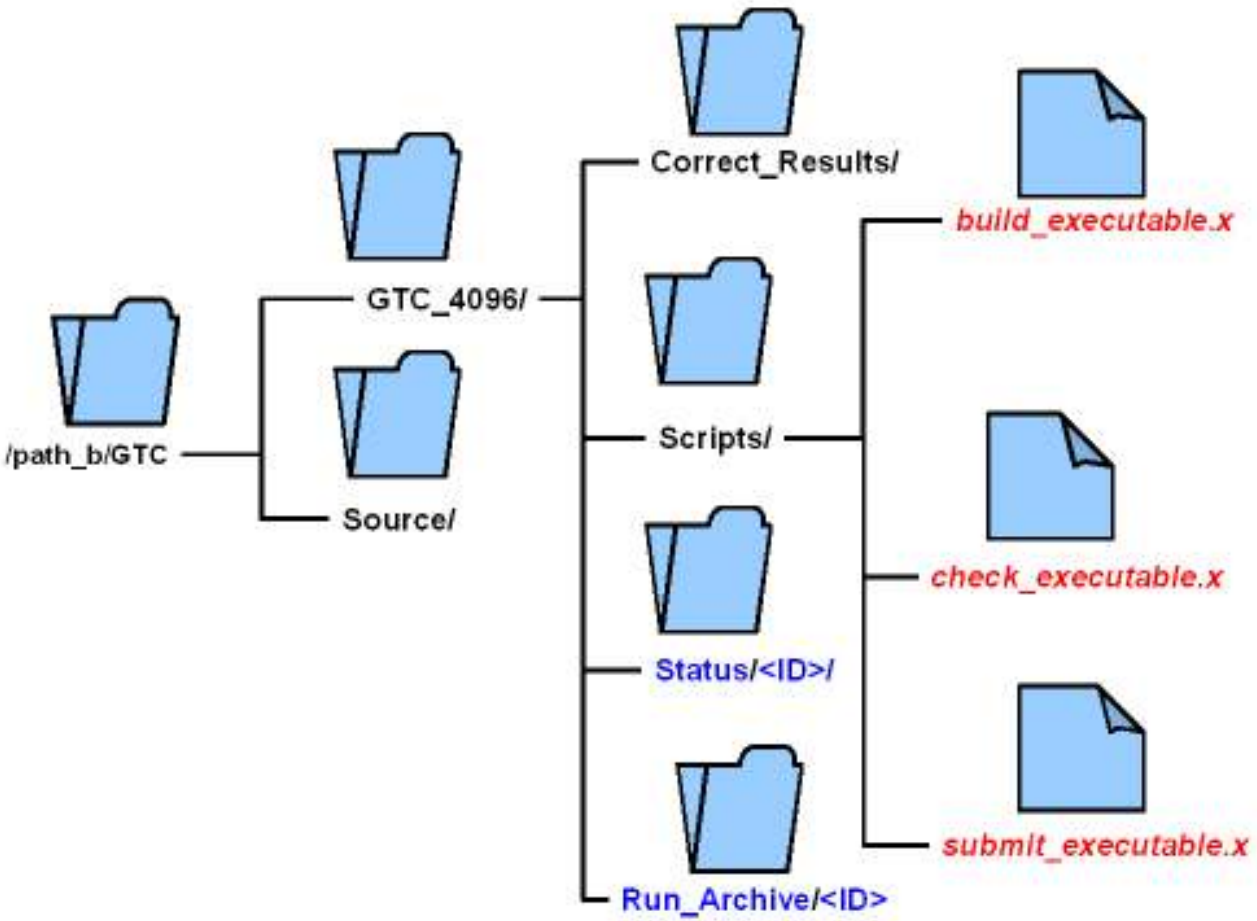
submit_me.x -i <ID> -p <path_to_unique_location>

-Returns the status of the submit.
-All jobs are run through batch
-Batch script calls check_me.x
-Batch script calls harness driver to run next instance of test

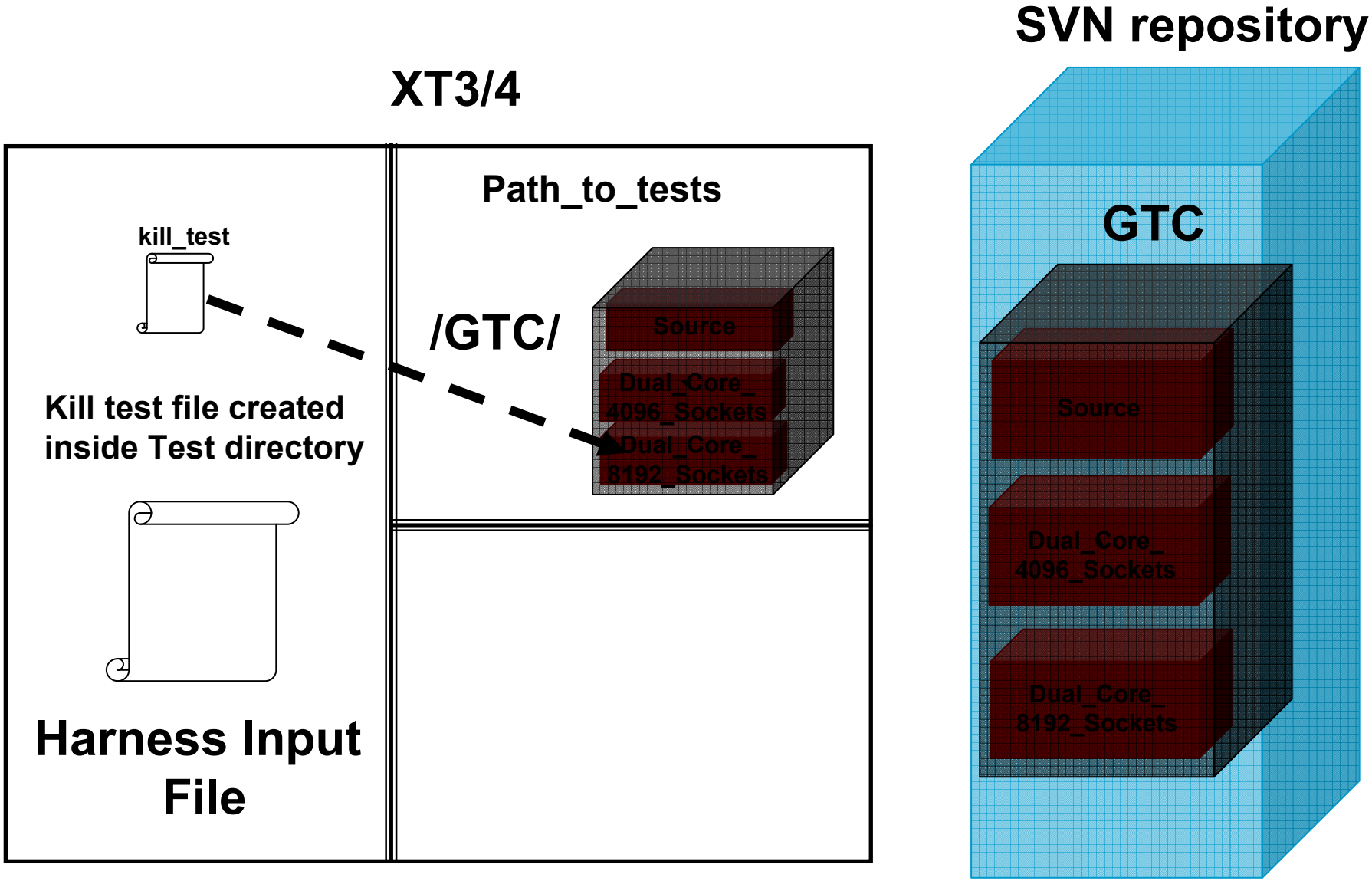
check_me.x -i <ID> -p <path_to_unique_location>

Append results of test to “rgt_status.txt”

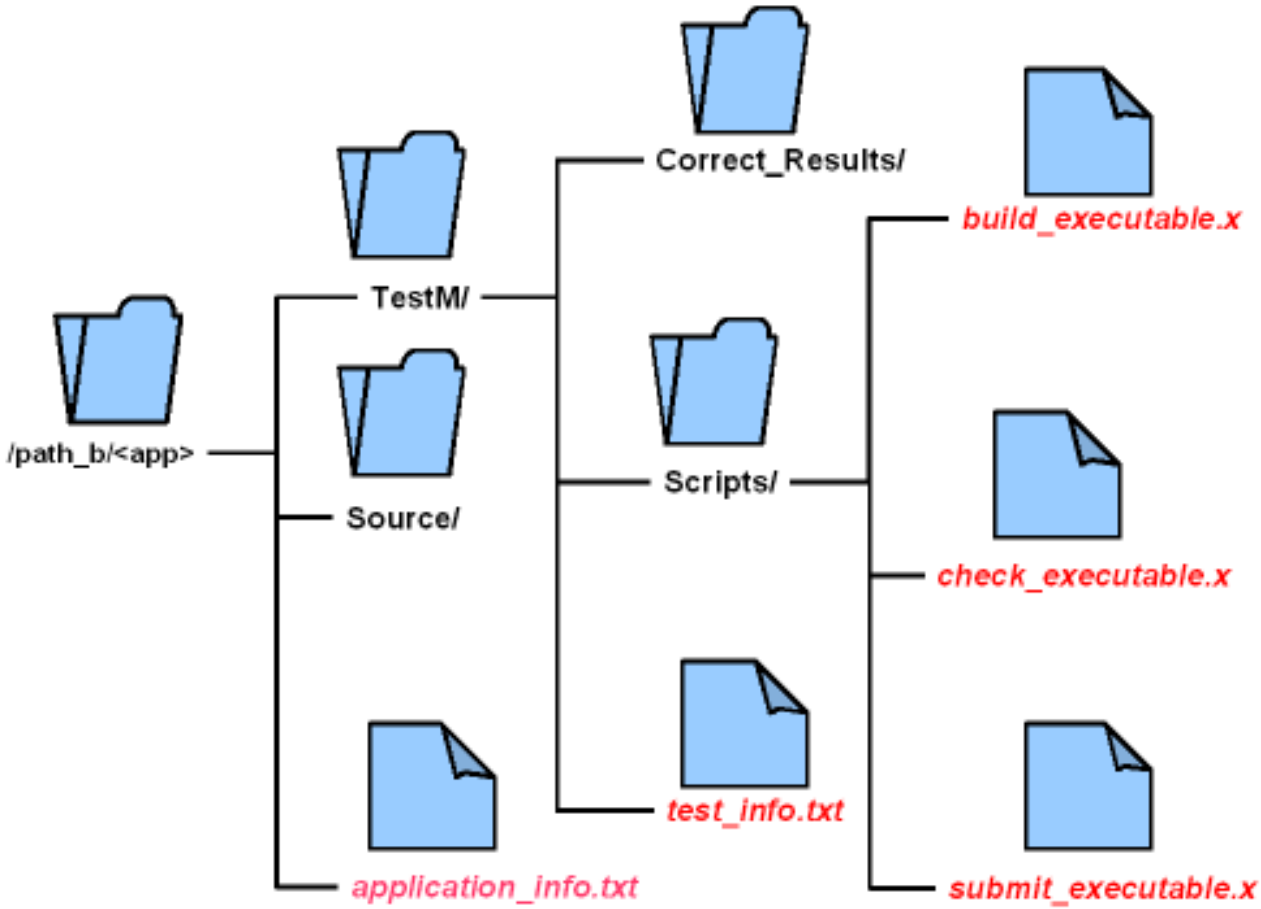
Directory Layout of Harness Run



Harness in Action (Stopping a Test)



Implementing a test



Conclusions and Future Directions

- Harness was of great value in running acceptance test
- Initial penalty due to long startup.
- Lustre file system sometimes causes jobs to exceed batch wall times.

- Need more Stressful Tests in Harness
- Need a wider variety of Tests in Harness
- Need to improve error recovery abilities

Acknowledgements

- Buddy Bland, Ann Baker, Don Maxwell, and Ricky Kendal
  - **Drafted for this project**
- NCCS Scientific Computing Group
  - **Many design iterations**
  - **Implementation of tests**