The Automatic Library Tracking Database

Mark Fahey

National Institute for Computational Sciences

Scientific Computing Group Lead

May 24, 2010

Cray User Group
May 24-27, 2010

Contributors

- Ryan Blake Hitchcock
- Patrick Lu
- Nick Jones
- Bilel Hadri

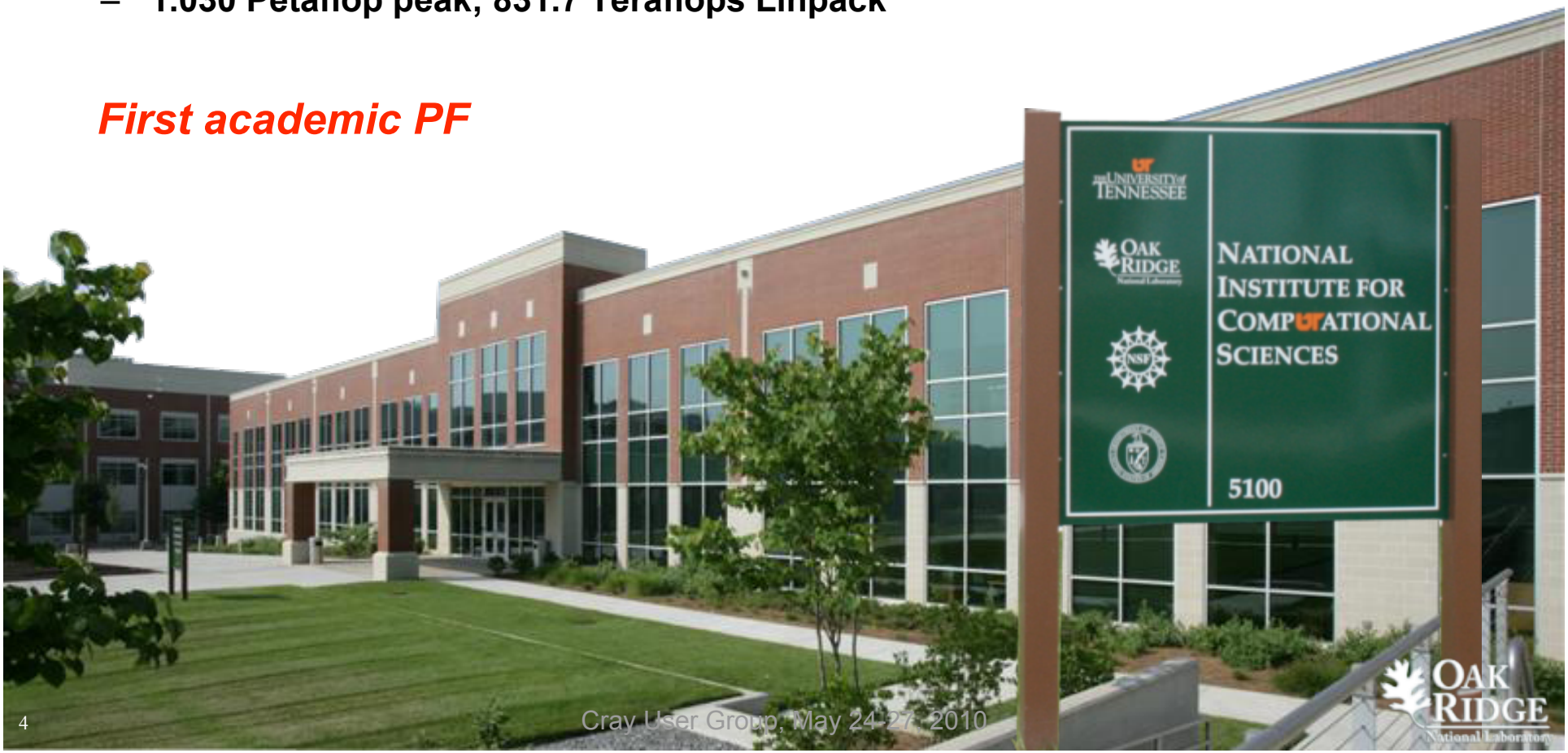
Outline

- NICS/OLCF
- Motivation for tracking library use
- Design/Implementation
- Results
- Conclusions

National Institute for Computational Sciences
University of Tennessee

- NICS is the latest NSF HPC center
- Kraken #3 on Top 500
  - 1.030 Petaflop peak; 831.7 Teraflops Linpack

First academic PF



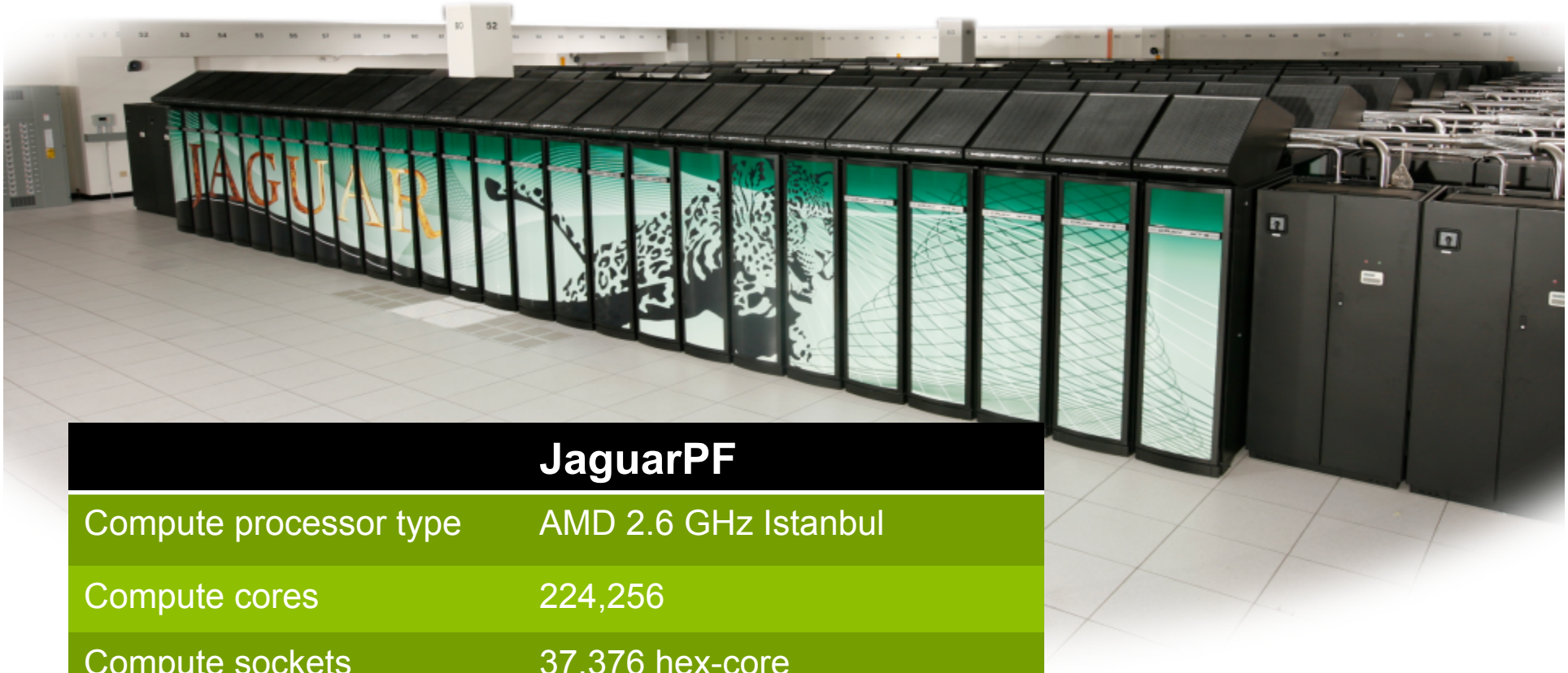
Kraken XT5

	Kraken
Compute processor type	AMD 2.6 GHz Istanbul
Compute cores	99,072
Compute sockets	16,512 hex-core
Compute nodes	8,256
Memory per node	16 GB (1.33 GB/core)
Total memory	129 TB

Oak Ridge Leadership Computing Facility

- JaguarPF #1 on Top 500
  - 2.331 Petaflops peak, 1.759 Petaflops Linpack
- Center (40,000 ft^2)

JaguarPF XT5



	JaguarPF
Compute processor type	AMD 2.6 GHz Istanbul
Compute cores	224,256
Compute sockets	37,376 hex-core
Compute nodes	18,688
Memory per node	16 GB (1.33 GB/core)
Total memory	362 TB

Motivation

- Issues
  - Centers support >100 software packages
  - Supporting multiple compilers (>=3)
  - Multiple versions of each library
- Want to
  - have the software users need; “stay ahead” of user requests
  - change default versions as needed
  - clean up; keep list of software presented to users reasonable
- How do
  - we know when to change defaults (to newer versions)
  - we know when we can get rid of old versions
  - we find out who is using
    - deprecated software?
    - software with bugs?
    - software funded by NSF/DOE?

Software maintained on Kraken

```
------------------------------------------------- /sw/xt/modulefiles -------------------------------------------------
CTSSV4                          fftw/3.1.2                      java-jre/1.5.0.06               pspline/1.0
DefApps                         fftw/3.1.2-dualcore             lammps/Jan10                    python/2.5.2
MiscApps                        fftw/3.3_alpha                  lammps/Mar09(default)           python/2.6.4(default)
abinit/6.0.2                    fpmpi/1.1                       lammps/Oct09                    python/3.1.1
altd/1.0                        fpmpi_papi/1.1                  lapack/3.1.1(default)           q-espresso/4.1.2
amber/10                        fsplit/1.0                      lapack/3.1.1-dualcore           qbox/1.47
amber/9                         gamess/2008Mar04                lapack/3.1.1-fPIC               qbox/1.50
ambertools/1.3                  gamess/2009Jan12                libart/2.3.19                   qt/4.3.4
apache-ant/1.6.5                gdlib/2.0.35                    marmot/2.3.0                    qt/4.5.2
aprun-wrapper/0.1               gempak/5.11.4                   mercurial/1.3                   ruby/1.9.1
apwrap/0.1(default)             ghostscript/8.64(default)       metis/4.0.1                     scalasca/1.1
apwrap/0.2                      gimp/2.6.4                      mpe2/1.0.6                      scalasca/1.2(default)
arpack/2008.03.11               git/1.6.4.3                     mpip/3.1.2                      sprng/2.0b
atk/1.24.0                      glib/2.18.3                     mumps/4.7.3_par                 srb-client/3.4.1-r3(default)
atlas/3.8.3                     globalarrays/4.1.1              mumps/4.9.2_par                 subversion/1.4.6(default)
atlas/3.8.3-fPIC-dualcore       globus/4.0.8                    namd/2.6                        subversion/1.5.0
aztec/2.1                       gmake/3.81                      namd/2.7b1                      subversion/1.6.9
blas/ref(default)               gnuplot/4.2.6(default)          namd/2.7b1-09Jul21              sundials/2.3.0
blas/ref-dualcore               gptl/3.5(default)               namd/2.7b2                      superlu/3.1
bugget/2.0                      grace/5.1.21                    nano/2.0.9                      superlu/4.0
cairo/1.8.6                     grads/2.0.a7.1                  ncl/5.0.0                       superlu_dist/2.3
casino/2.5                      gromacs/4.0.5                   ncl/5.0.0_source                swig/1.3.36
cdo/1.3.2                       gromacs/4.0.7(default)          nco/3.9.9                       szip/2.1
cdo/1.4.1                       gromacs/4.0.7_fprelaxed         nco/4.0.0                       tau/2.19(default)
charm++/6.1.3                   gsl/1.13                        ncview/1.93g                    tg-policy/0.2-r1
cmake/2.6.4(default)            gsl/1.13-dualcore               nedit/5.5                       tginfo/1.1.0-r1
cmake/2.8.0                     gtk/2.14.6                      netcdf/3.6.2                    tgusage/3.0-r2(default)
condor/7.0.4-r1                 gv/3.6.8                        netcdf/3.6.3                    tiff/3.8.2
cpmd/3.13.2                     hdf4/4.2r4                      netcdf/4.1                      tkdiff/4.1.4
desmond/2.2.7.3_dbl             hdf5/1.6.10                     netcdf-parallel/4.1             totalview/8.6.0-1
desmond/2.2.7.3_sngl            hdf5/1.8.3                      numpy/1.3.0                     totalview/8.7.0-1(default)
espresso/2.1.0gnu(default)      hdf5/1.8.4                      nwchem/5.1                      trilinos/10.0.2
espresso/2.1.0pgi               hdf5-parallel/1.6.10            p-netcdf/1.0.3                  trilinos/9.0.3
espresso/2.1.2j-gnu             hdf5-parallel/1.8.3             p-netcdf/1.1.1                  udunits/1.12.9
espresso/2.1.2j-pgi             hdf5-parallel/1.8.4             pacman/3.26-r1                  udunits/2.1.13
ferret/6.1                      hypre/2.0.0                     pango/1.20.5                    umfpack/5.1.1
fftpack/5-r4i4                  imagemagick/6.5.3               parmetis/3.1                    umfpack/5.4.0
fftpack/5-r8i4                  imagemagick/6.6.1(default)      petsc/2.3.3-debug               upc/2.8.0
fftpack/5-r8i8                  iobuf/beta                      petsc-complex/2.3.3-debug       valgrind/3.4.1
fftw/2.1.5                      java-jdk/1.5.0.06               pgplot/5.2                      valgrind/3.5.0
fftw/2.1.5-dualcore_            java-jdk/1.6.0.06               pixman/0.13.2                   vim/7.2(default)
```

NICS

Objective

- Track libraries that are linked into executables
- Track executables run (and by inference) how often are the libraries used?
  - Of course, not necessarily true

Assumptions/Requirements

- Must support statically linked executables
  - Shared library support desirable as well
- Have as little impact on user as possible
  - Lightweight solution
    - No runtime increase
    - Only link time and job launch have marginal increase in time
  - Do not change user experience
    - Linker and job launcher work as expected
- Tracking libraries
  - Not function calls
- Only libraries actually linked into executable

Design

- Wrap binutils “ld” and job launcher “aprun”
  - This allows us to track libraries at link time
  - This allows us to track executables that we can tie back to the actually link and thus the libraries
- ld - Intercept link line
  - Update tags table
  - Create altd.o to link into executable
  - Call real linker (with tracemap option)
  - Use output from tracemap to find libraries linked into executable
  - Update linkline table
  - (Could stop here)
- aprun- Intercept job launcher
  - Pull information from altd section header in executable
  - Update jobs table
  - Call real job launcher

altd.o

- **Assembly code inserted into binaries**

```
.section .altd
.asciz "ALTD_Link_Info"

.byte 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
.asciz "Version:0.7:"
.asciz "Machine:athena:"
.asciz "Tag_id:38:"
.asciz "Year:2009:"
.byte 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
.asciz "ALTD_Link_Info_End"
```

MySQL database

- 3 tables: tags, linkline, and jobs
  - Tags – entry for every link executed
    - ld wrapper does 2 steps
      - First pass, entry added to include user name, date stamp
      - On the final pass of the ld wrapper, previous entry is updated with the linkline table "id"
    - This gives first count of library usage => # times used in link
  - Linkline – entry for each unique link line
    - Inserted if new on 2nd pass of ld wrapper
  - Jobs – entry for each executable launched
    - The "tag id" and "build machine" is pulled from the binary and stored
    - This table gives us another way to count library "usage"
      - Usage => how many times code was run

tags table

tag_id	linkline_id	username	exit_code	link_date
91126	14437	user1	0	2010-04-28
91127	0	user2	-1	2010-04-28
91128	14435	user3	0	2010-04-28
91129	6835	user2	0	2010-04-28
91130	14438	user4	0	2010-04-28
91131	14439	user1	0	2010-04-28
91132	14439	user1	0	2010-04-28

linkline table

linkline_id	linkline
14437	../bin/cg.B.4 /usr/lib/../lib64/crt1.o /usr/lib/../lib64/crti.o /opt/gcc/4.4.2/snos/lib/gcc/x86_64-suse-linux/4.4.2/crtbeginT.o /sw/xt/tau/2.19/cnl2.2_gnu4.4.1/tau-2.19/craycnl/lib/libTauMpi-gnu-mpi-pdt.a /sw/xt/tau/2.19/cnl2.2_gnu4.4.1/tau-2.19/craycnl/lib/libtau-gnu-mpi-pdt.a /usr/lib/../lib64/libpthread.a /opt/cray/mpt/4.0.1/xt/seastar/mpich2-gnu/lib/libmpich.a /opt/cray/pmi/1.0-1.0000.7628.10.2.ss/lib64/libpmi.a /usr/lib/alps/libalpslli.a /usr/lib/alps/libalpsutil.a /opt/xt-pe/2.2.41A/lib/snos64/libportals.a […. gcc 4.4.2 libraries …] /usr/lib/../lib64/libc.a /usr/lib/../lib64/crtn.o
14438	highmass3d.Linux.CC.ex /usr/lib64/crt1.o /usr/lib64/crti.o /opt/pgi/9.0.4/linux86-64/9.0-4/lib/trace_init.o /usr/lib64/gcc/x86_64-suse-linux/4.1.2/crtbeginT.o /sw/xt/hypre/2.0.0/cnl2.2_pgi9.0.1/lib//libHYPRE.a /opt/cray/pmi/1.0-1.0000.7628.10.2.ss/lib64/libpmi.a /usr/lib/alps/libalpslli.a /usr/lib/alps/libalpsutil.a /opt/xt-pe/2.2.41A/lib/snos64/libportals.a /usr/lib64/libpthread.a /usr/lib64/libm.a /usr/local/lib/libmpich.a […. pgi 9.0.4 libraries …] /usr/lib64/librt.a /usr/lib64/libpthread.a /usr/lib64/libm.a /usr/lib64/gcc/x86_64-suse-linux/4.1.2/libgcc_eh.a /usr/lib64/libc.a /usr/lib64/gcc/x86_64-suse-linux/4.1.2/crtend.o /usr/lib64/crtn.o
14439	probeTest /usr/lib/../lib64/crt1.o /usr/lib/../lib64/crti.o /opt/gcc/4.4.2/snos/lib/gcc/x86_64-suse-linux/4.4.2/crtbeginT.o /opt/cray/mpt/4.0.1/xt/seastar/mpich2-gnu/lib/libmpich.a /opt/cray/pmi/1.0-1.0000.7628.10.2.ss/lib64/libpmi.a /usr/lib/alps/libalpslli.a /usr/lib/alps/libalpsutil.a /opt/xt-pe/2.2.41A/lib/snos64/libportals.a /usr/lib/../lib64/libpthread.a […. gcc 4.4.2 libraries …] /usr/lib/../lib64/libc.a /usr/lib/../lib64/crtn.o

jobs table

run_inc	tag_id	executable	username	run_date	job_launch_id	build_machine
144091	91126	/nics/b/home/user1/NPB3.3/bin/cg.B.4	user1	2010-04-28	548346	kraken
144099	91131	/nics/b/home/user1/probeTest	user1	2010-04-28	548357	kraken
144102	91132	/nics/b/home/user1/probeTest	user1	2010-04-28	548357	kraken
144179	91128	/lustre/scratch/user3/CH4/vasp_vtst.x	user3	2010-04-28	548444	kraken
144192	91128	/lustre/scratch/user3/CH4/vasp_vtst.x	user3	2010-04-28	548488	kraken
144356	91128	/lustre/scratch/user5/src/CH4/vasp_vtst.x	user5	2010-04-29	548638	kraken

linking_inc	linkline
14437	../bin/cg.B.4 /usr/lib/../lib64/crt1.o /usr/lib/../lib64/crti.o /opt/gcc/4.4.2/snos/lib/gcc/x86_64-suse-linux/4.4.2/crtbeginT.o /sw/xt/tau/2.19/cnl2.2_gnu4.4.1/tau-2.19/craycnl/lib/libTauMpi-gnu-mpi-pdt.a /sw/xt/tau/2.19/cnl2.2_gnu4.4.1/tau-2.19/craycnl/lib/libtau-gnu-mpi-pdt.a /usr/lib/../lib64/libpthread.a /opt/cray/mpt/4.0.1/xt/seastar/mpich2-gnu/lib/libmpich.a /opt/cray/pmi/1.0-1.0000.7628.10.2.ss/lib64/libpmi.a /usr/lib/alps/libalpslli.a /usr/lib/alps/libalpsutil.a /opt/xt-pe/2.2.41A/lib/snos64/libportals.a /opt/gcc/4.4.2/snos/lib/gcc/x86_64-suse-linux/4.4.2/libgfortranbegin.a /opt/gcc/4.4.2/snos/lib/gcc/x86_64-suse-linux/4.4.2/libgcc.a /opt/gcc/4.4.2/snos/lib/gcc/x86_64-suse-linux/4.4.2/libgcc_eh.a /usr/lib/../lib64/libc.a /opt/gcc/4.4.2/snos/lib/gcc/x86_64-suse-linux/4.4.2/crtend.o /usr/lib/../lib64/crtn.o
14438	highmass3d.Linux.CC.ex /usr/lib64/crt1.o /usr/lib64/crti.o /opt/pgi/9.0.4/linux86-64/9.0-4/lib/trace_init.o /usr/lib64/gcc/x86_64-suse-linux/4.1.2/crtbeginT.o /sw/xt/hypre/2.0.0/cnl2.2_pgi9.0.1/lib//libHYPRE.a /opt/cray/pmi/1.0-1.0000.7628.10.2.ss/lib64/libpmi.a /usr/lib/alps/libalpslli.a /usr/lib/alps/libalpsutil.a /opt/xt-pe/2.2.41A/lib/snos64/libportals.a /usr/lib64/libpthread.a /usr/lib64/libm.a /usr/local/lib/libmpich.a /opt/pgi/9.0.4/linux86-64/9.0-4/lib/libstd.a /opt/pgi/9.0.4/linux86-64/9.0-4/lib/libC.a /opt/pgi/9.0.4/linux86-64/9.0-4/lib/libpgf90.a /opt/pgi/9.0.4/linux86-64/9.0-4/lib/libpgc.a /usr/lib64/librt.a /usr/lib64/libpthread.a /usr/lib64/libm.a /usr/lib64/gcc/x86_64-suse-linux/4.1.2/libgcc_eh.a /usr/lib64/libc.a /usr/lib64/gcc/x86_64-suse-linux/4.1.2/crtend.o /usr/lib64/crtn.o
14439	probeTest /usr/lib/../lib64/crt1.o /usr/lib/../lib64/crti.o /opt/gcc/4.4.2/snos/lib/gcc/x86_64-suse-linux/4.4.2/crtbeginT.o /opt/cray/mpt/4.0.1/xt/seastar/mpich2-gnu/lib/libmpich.a /opt/cray/pmi/1.0-1.0000.7628.10.2.ss/lib64/libpmi.a /usr/lib/alps/libalpslli.a /usr/lib/alps/libalpsutil.a /opt/xt-pe/2.2.41A/lib/snos64/libportals.a /usr/lib/../lib64/libpthread.a /opt/gcc/4.4.2/snos/lib/gcc/x86_64-suse-linux/4.4.2/libgcc_eh.a /usr/lib/../lib64/libc.a /opt/gcc/4.4.2/snos/lib/gcc/x86_64-suse-linux/4.4.2/crtend.o /usr/lib/../lib64/crtn.o

a) Linkline table

tag_id	linkline_id	username	exit_code	link_date
91126	14437	user1	0	2010-04-28
91127	0	user2	-1	2010-04-28
91128	14435	user3	0	2010-04-28
91129	6835	user2	0	2010-04-28
91130	14438	user4	0	2010-04-28
91131	14439	user1	0	2010-04-28
91132	14439	user1	0	2010-04-28

b) tag_id table

run_inc	tag_id	executable	username	run_date	job_launch_id	build_machine
144091	91126	/nics/b/home/user1/NPB3.3/bin/cg.B.4	user1	2010-04-28	548346	kraken
144099	91131	/nics/b/home/user1/probeTest	user1	2010-04-28	548357	kraken
144102	91132	/nics/b/home/user1/probeTest	user1	2010-04-28	548357	kraken
144179	91128	/lustre/scratch/user3/CH4/vasp_vtst.x	user3	2010-04-28	548444	kraken
144192	91128	/lustre/scratch/user3/CH4/vasp_vtst.x	user3	2010-04-28	548488	kraken

c) job_id table

Results

- **Most used libraries provided by Cray**

Rank	Kraken	JaguarPF
1	CrayPAT/5.0	CrayPAT/4.x
2	Libsci/10.4	PETSc/3.0
3	PETSc/3.0	PAPI/3.6
4	FFTW/3.2	ACML/4.2
5	HDF5/1.8	HDF5/1.8

3 months of Kraken data, JaguarPF data is for all of 2009

Results

- **Most used libraries provided by centers**

Rank	Kraken	JaguarPF
1	SPRNG/2.0b	SZIP/2.1
2	PETSc/2.3	HDF5/1.6
3	Iobuf/beta	Trilinos/9
4	TAU/2.19	PSPLINE/1.0
5	SZIP/2.1	NetCDF/3.6

3 months of Kraken data, JaguarPF data is for all of 2009

Results

- **Most used applications on Kraken (last 3 months)**

ALTD

Rank	Library	# instances
1	interpo**	60,032
2	namd*	8,389
3	amber*	5,784
4	chimera	4,000
5	mpiblast	2,917

From Torque job scripts

Rank	Library	# instances
1	arps	11,844
2	amber	6,789
3	namd	6,450
4	chimera	4,473
…		
8	mpiblast	2,919

Absolute number of executions, not CPU hours!
And only "launched jobs".

* Counting both center-provided and user-built applications

** Compiled on athena and run on Kraken

- Typically job script mining counts more because includes staff and matches strings that can appear in multiple places; and ALTD will miss some early after being turned on
- ALTD counted more for namd because we catch it each time it is launched, the scripts searching for namd in job scripts can't tell if it is inside a loop.

Results

- **Least used libraries on JaguarPF for 2009**

0 Usage Libraries
fftpack

0 Usage Libraries +Version
tau/2.17
hdf5 (various parallel versions)
fftw/3.2 (locally built)
acml/4.0.1

Clearly, supporting fftpack can stop
Old versions of tau and acml, for example, can be removed.
Locally built hdf5 and fftw/3 libraries are not being used because there is a Cray analogue!

Miscellaneous

- If a library is unused (or used very little)
  - How do we really know if we can stop support
    - Maybe the users “went away” for awhile
    - Need long duration and “recent” usage views
- Found we can’t just ignore all .o files
  - Iobuf – IO buffering library is a .o

Installation details

- Written in Python, original version in C
- Actual mode of interception
  - Modulefiles (prepend PATH)
  - Move/rename ld and aprun
  - Tied into admin's "aprun wrapper" as an aprun-prologue
    - See Matt Ezell's talk on Tuesday at 3:30
- Built in ability to turn tracking on/off with env vars
  - Per person if desired
- Gets complicated with tools like Totalview
  - Either "fix" Totalview or unload ALTD
    - Modified Totalview on JaguarPF
  - Unload ALTD modulefile on Kraken

Conclusions

- In production and tracking usage
  - We don't *really* know if the libraries were used
  - We *do know* they were linked into the application
- Almost unnoticed by users
  - One or two hiccups along the way, but were addressed quickly
- Mining the data is hard
  - Even with mostly consistent software installations, many exceptions when looking for patterns
- Can start making decisions about software support based on real usage
  - 1. Stop providing FFTPACK and an old version of ACML, TAU
  - 2. Users linking with Cray provided libraries
- Will be preparing a release of ALTD soon