Comparison of Scheduling Policies and Workloads on the NCCS and NICS XT4 Systems at Oak Ridge National Laboratory

Troy Baer
HPC System Administrator
National Institute for Computational Sciences
University of Tennessee

Don Maxwell
Senior HPC System Administrator
National Center for Computational Sciences
Oak Ridge National Laboratory

Overview

- Introduction
- System Descriptions
  - Hardware & Software
  - Batch Environment
    - Queue Structures
    - Job Prioritization
    - Quality of Service Levels
    - Other Scheduling Policies
- Allocation Processes
- Workload Analysis
  - Overall Utilization
  - Breakdown by Job Size
  - Quantifying User Experience
  - Application Usage
- Conclusions and Future Work

Introduction

- **Oak Ridge National Laboratory is home to two supercomputing centers:**
  - **National Center for Computational Sciences**
    - **Founded in 1992.**
    - **DoE Leadership Computing Facility**
  - **National Institute for Computational Science**
    - **Joint project between ORNL and University of Tennessee, founded in 2007.**
    - **NSF Petascale Track 2B awardee**
- **Both centers have Cray XT4 systems**
  - **Jaguar (NCCS)**
  - **Kraken (NICS)**
- **Both systems have the goal of running as many big jobs as possible**

System Hardware and Software

Jaguar

- **84 cabinets**
- **7,832 compute nodes (31,328 cores)**
- **Quad-core Opteron @ 2.1 GHz**
- **61.19 TB of RAM**
- **700 TB of disk**
- **CLE 2.0**

Kraken

- **40 cabinets**
- **4,508 compute nodes (18,032 cores)**
- **Quad-core Opterons @ 2.3 GHz**
- **17.61 TB of RAM**
- **450 TB of disk**
- **CLE 2.0**

Batch Environment

- **Both Jaguar and Kraken use TORQUE as their batch system, with Moab as the scheduler.**
- **Moab has a number of advanced features, including a "native resource manager" interface for connecting to e.g. ALPS.**
- **While the software is the same on the two systems, there are significant differences in how it is configured on the two systems.**

Jaguar Queue Structure

- **`dataxfer`**
  - **Size = 0**
  - **Max time = 24 hrs.**
- **`batch`**
  - **Max time = 24 hrs.**
- **`debug`**
  - **Max time = 4 hrs.**
- **Additional walltime limits for smaller jobs (size<1024) imposed by TORQUE submit filter**

Kraken Queue Structure

- **`dataxfer`**
  - **Size = 0**
  - **Max time = 24 hrs.**
- **`small`**
  - **0 >= size >= 512**
  - **Max time = 12 hrs.**
- **`longsmall`**
  - **0 <= size <= 512**
  - **Max time = 60 hrs.**
- **`medium`**
  - **512 < size <= 2048**
  - **Max time = 24 hrs.**
- **`large`**
  - **2048 < size <= 8192**
  - **Max time = 24 hrs.**
- **`capability`**
  - **8192 < size <=18032**
  - **Max time = 48 hrs.**

Job Prioritization

Jaguar

- **Priority thought of in units of "days", equivalent to one day of queue wait time**
- **Components:**
  - **QoS, assigned based mainly on job size**
  - **Queue wait time**
  - **Fair share targets assigned to QoS**

Kraken

- **Priority units are arbitrary**
- **Components:**
  - **Job size**
  - **Queue wait time**
  - **Expansion factor (ratio of queue time plus run time to run time)**

Quality of Service Levels on Jaguar

- **sizezero**
  - size = 0
  - +90 days priority.
  - Max 10 jobs/user.
- **smallmaxrun**
  - 0 < size <= 256
  - 20% fair share target.
  - Max 2 jobs/user.
- **nonldrship**
  - 256 < size <= 6000
  - 20% fair share target.
- **ldrship**
  - 6000 < size <= 17000
  - +8 days priority.
  - 80% fair share target.
- **topprio**
  - size > 17000
  - +10 days priority.
  - 80% fair share target.

Quality of Service Levels on Kraken

- **`sizezero`**
  - **size=0**
  - **Queue time target of 00:00:01.**
  - **Priority grows exponentially after queue time target is passed.**
- **`negbal`**
  - **Applied to jobs from projects with negative balances.**
  - **-100000 priority.**
  - **Additional penalties (e.g. disabling backfill or a small fair share target) have been discussed as well.**

Other Scheduling Policies on Kraken

- **`longsmall` jobs limited to 1,600 cores total.**
- **Only 1 capability is eligible to run at any given time.**

Allocation Processes

Jaguar

- **DoE INCITE**
- **Made annually**
- **Allocations can last multiple years**
- **Applications must be able to use a "significant fraction" of LCF systems at ORNL and/or ANL**

Kraken

- **NSF/Teragrid TRAC**
- **Made quarterly**
- **Allocations last one year (i.e. "use it or lose it")**
- **No major requirement on application scalability**

Workload Analysis

- **TORQUE accounting records parsed and loaded into a database.**
- **Job scripts also captured and stored in DB.**
  - **On Kraken, this happens automatically.**
  - **On Jaguar, the `aprun` parts of scripts are reconstructed using another database.**
- **Period of interest is the 4th quarter of 2008.**
  - **Both XT4 machines in production and allocated.**
  - **XT5 successor systems not yet generally available.**
- **To be able to compare apples to apples, size breakdowns are normalized by the size of each machine.**

Overall Utilization for 4Q2008

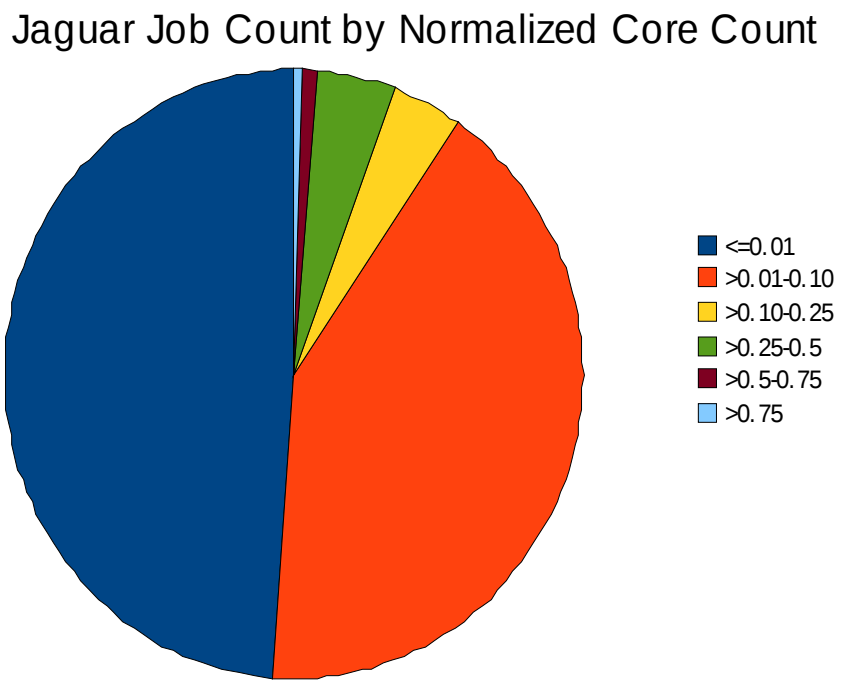
Jaguar

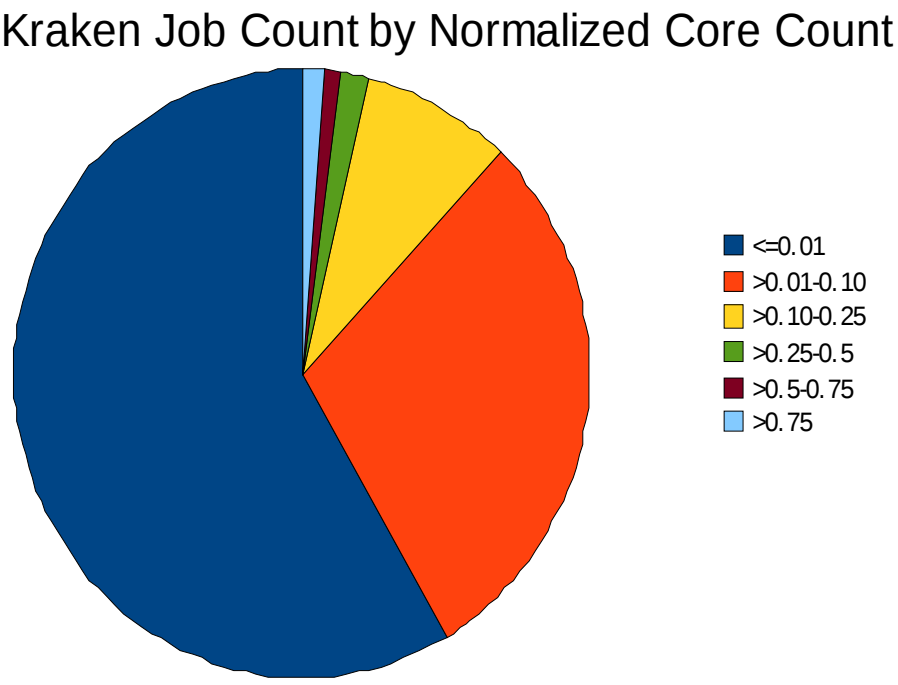
- **46,023 jobs run.**
- **54.46 million CPU-hours consumed.**
- **89.7% average utilization.**
- **300 active users.**
- **142 active projects.**

Kraken

- **15,744 jobs run.**
- **21.00 million CPU-hours consumed.**
- **57.0% average utilization.**
- **116 active users.**
- **40 active projects.**

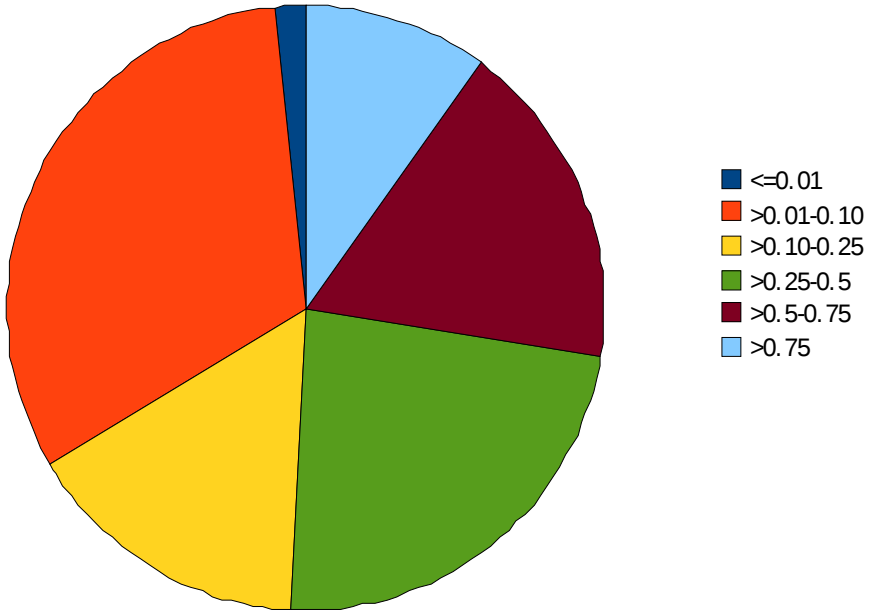
Breakdown by Job Size -- Count



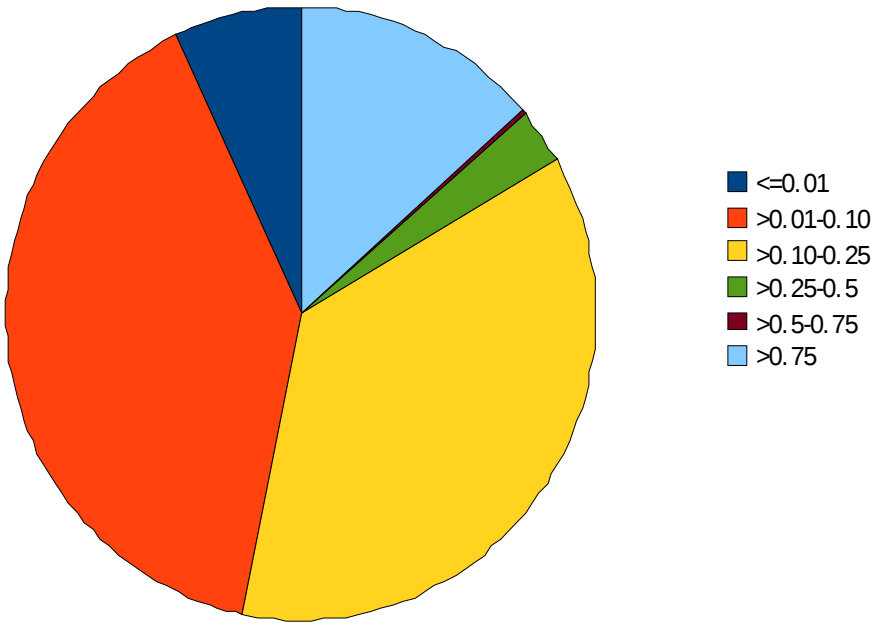


Breakdown by Job Size – CPU Hours

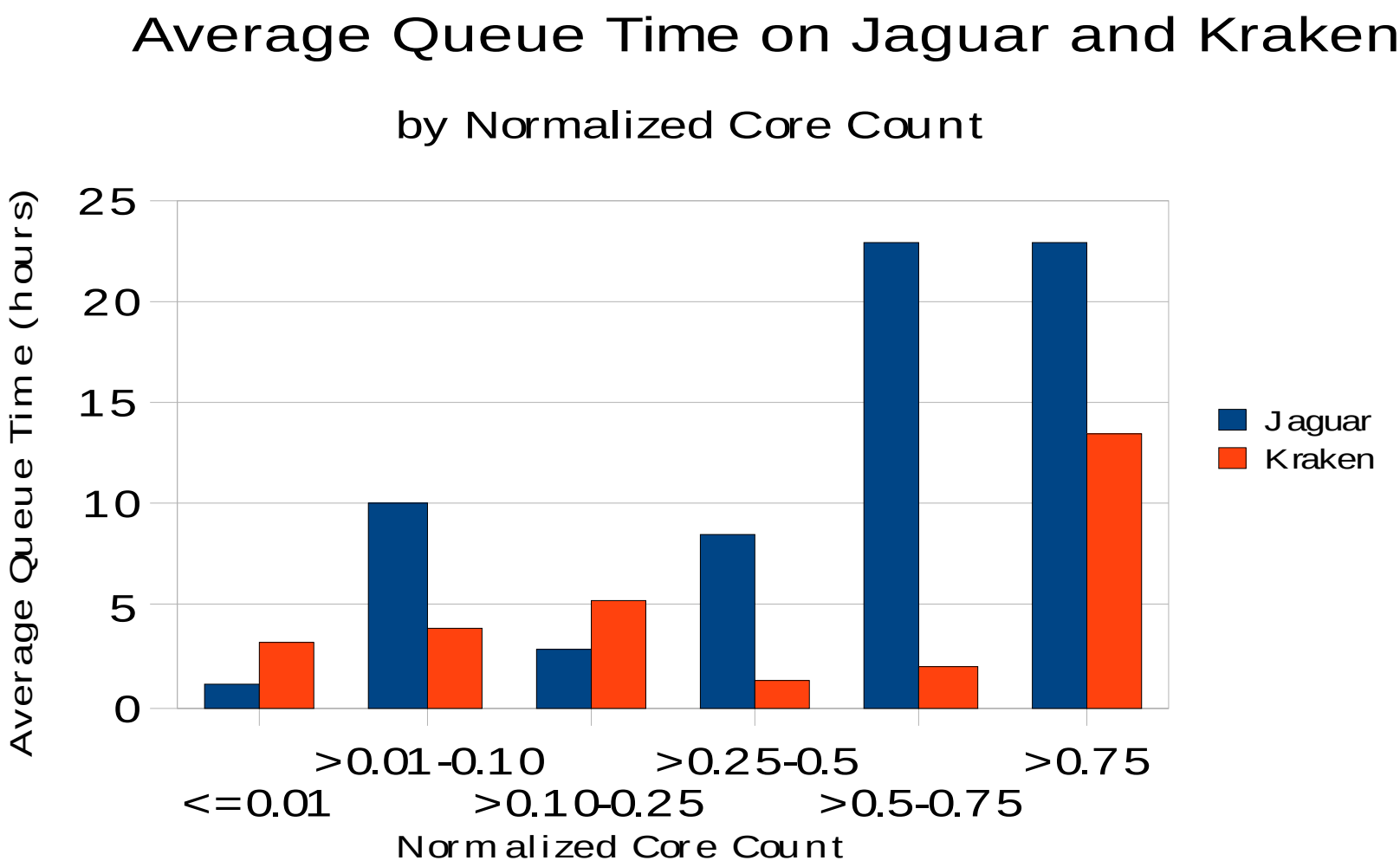
Jaguar CPU Hours by Normalized Core Count



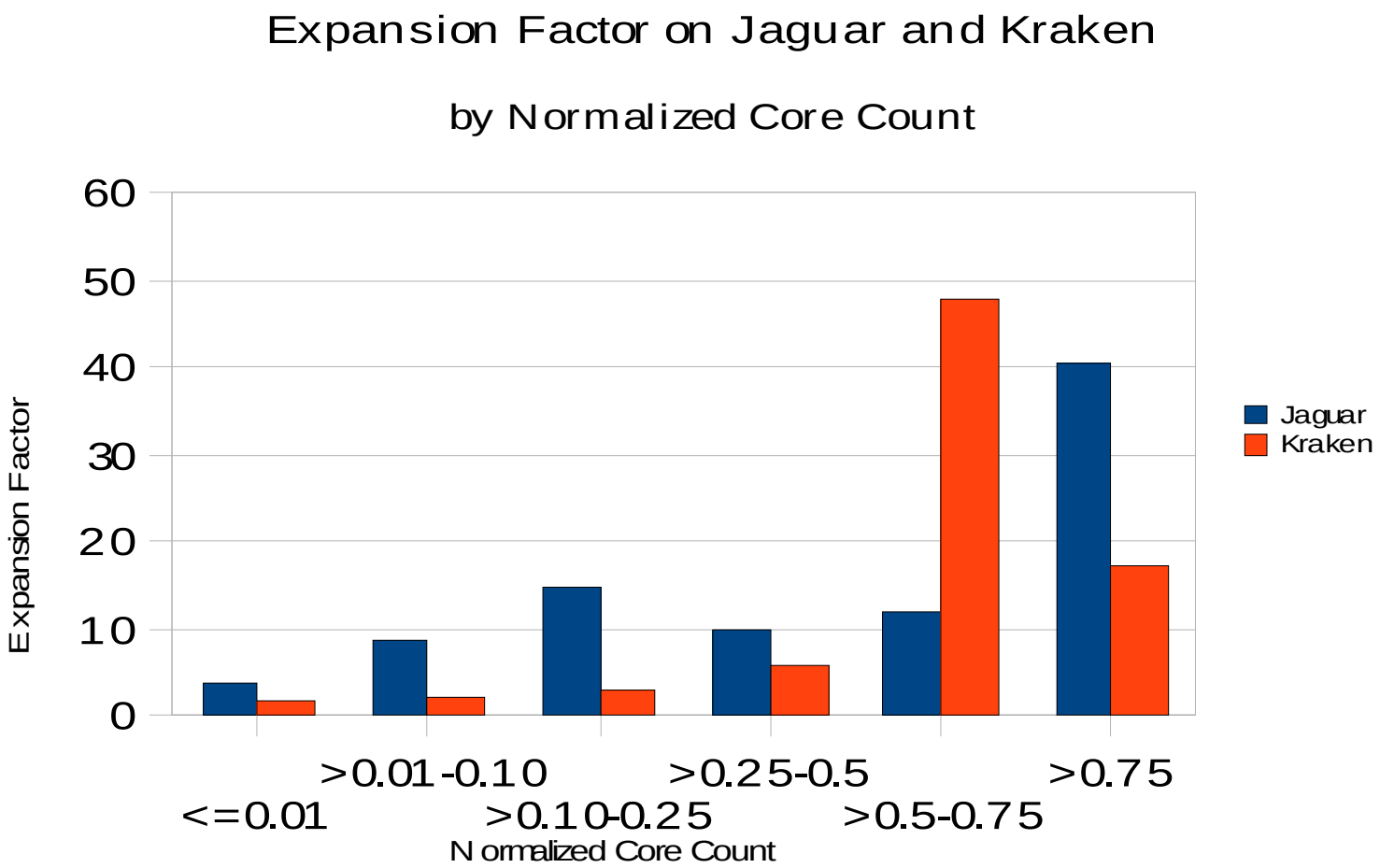
Kraken CPU Hours by Normalized Core Count



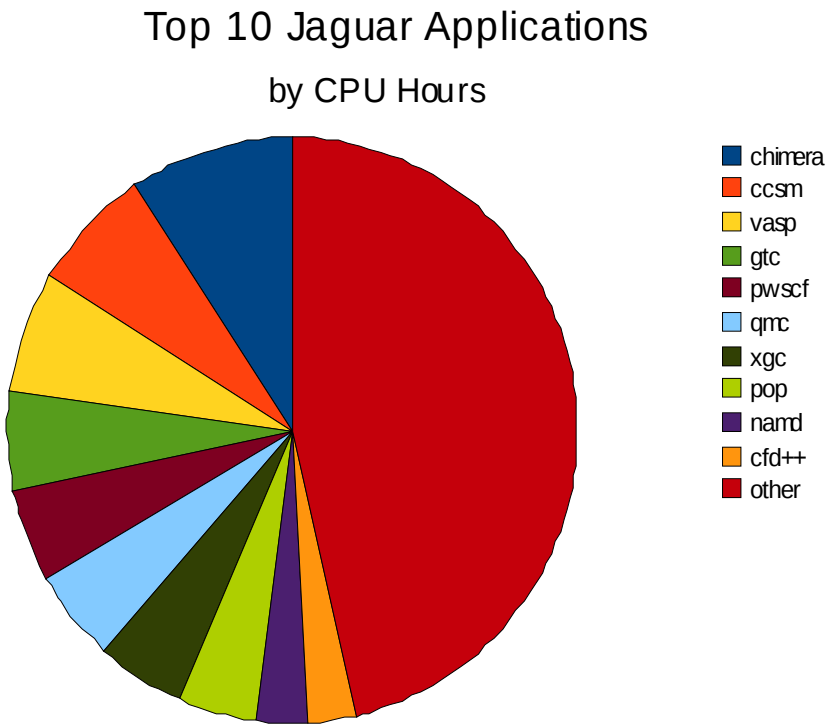
Quantifying User Experience

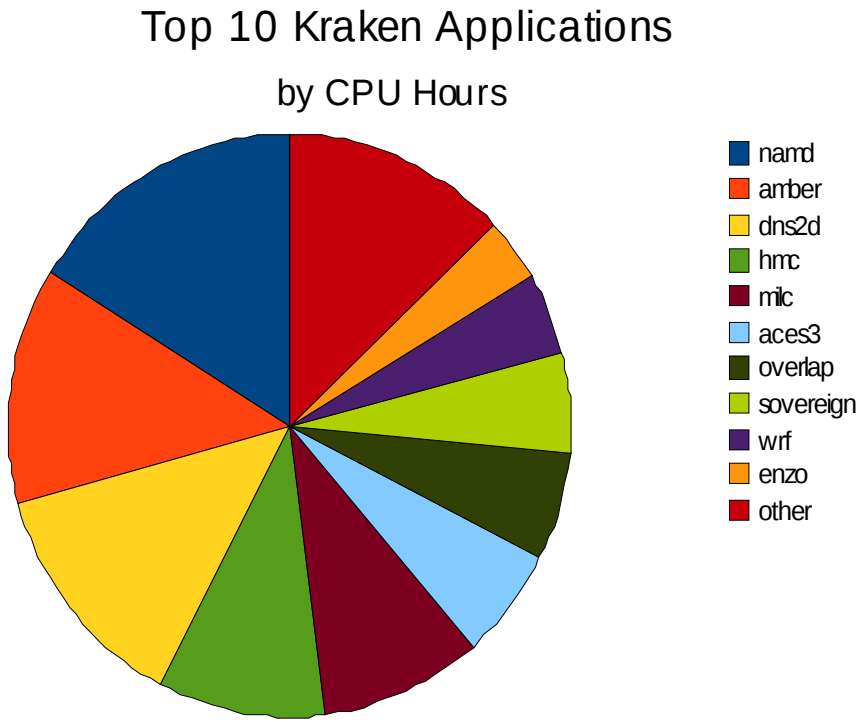


Quantifying User Experience (con't.)



Application Usage





Conclusions

- **Jaguar and Kraken actually do a lot of the same things using different mechanisms**
- **Both systems achieve their goal of running the big jobs**
  - **For Jaguar, this consists mostly of jobs using 10% or more of the system each, with a strong bias toward jobs using 25% or more.**
  - **For Kraken, this is a more bimodal distribution with many small jobs (<25% of the system) and a significant number of whole-system jobs with no much in between.**
  - **Difference is largely due by how the systems are allocated.**

Future Work

Future Work (con't)

- **XT5 systems will require some changes due to sheer scale.**
- **Better understanding of queue time**
  - **Resource availability and policy components.**
  - **Some times these overlap (e.g. standing reservations).**
- **Fair share on Kraken?**
  - **On per-project basis, based on allocation balance.**
- **More complex queue structure on Jaguar?**
  - **Centralize where walltime limits are defined.**
  - **Would simplify TORQUE submit filter.**