Application Workloads on the Jaguar Cray XT5 System

Wayne Joubert
Oak Ridge Leadership Computing Facility
Oak Ridge National Laboratory
Oak Ridge, TN USA
joubert at ornl.gov

Shiquan Su
National Institute for Computational Sciences
Oak Ridge, TN USA
shiquansu at hotmail.com

***Abstract*—In this study we investigate computational workloads for the Jaguar system during its tenure as a 2.3 petaflop system at Oak Ridge National Laboratory. The study is based on a comprehensive analysis of MOAB and ALPS job logs over this period. We consider Jaguar utilization over time, usage patterns by science domain, most heavily used applications and their usage patterns, and execution characteristics of selected heavily-used applications. Implications of these findings for future HPC systems are also considered.**

Keywords-HPC, workload, applications, science, Cray, ORNL, metrics, petascale, scaling

I. INTRODUCTION

Oak Ridge National Laboratory (ORNL) has a long lineage of involvement in state-of-the-art high performance computing (HPC) dating back to the early 1950s. In recent years, the Oak Ridge Leadership Computing Facility (OLCF) has deployed a series of HPC systems of increasing computational power, from a 6.4 TF Cray X1 in 2004, a 18.5 Cray X1e upgrade in 2005, a 26 TF Cray XT3 in 2005, a 54 TF dual-core system in 2006, an upgrade to a 119 TF Cray XT4 in 2007, an upgrade to 263 TF in 2008, a 1.375 PF Cray XT5 in 2008, and an upgrade to 2.3 PF in 2009. The Jaguar system has recently undergone an upgrade to a 3.3 PF XK6 system in anticipation of NVIDIA GPUS to be installed in late 2012 to form the 20 PF Titan system. Later, ORNL plans to site the OLCF-4 system of 100-250 PF in 2016 and the OLCF-5 exascale platform in the 2018-2020 period. The continued growth in computational capabilities is driven by key priorities of the U.S. Department of Energy (DOE) to deploy computational tools enabling researchers to analyze, model, simulate and predict complex phenomena in order to solve urgent challenges in national and homeland security, energy security, economic competitiveness, health care and environmental protection.

The effectiveness of these systems is ultimately tied to the quality of the science application codes that are run on these systems. Assessing and assuring the efficiency and scalability of the applications used to produce science is a vital priority as hardware continues to grow in computational power and complexity.

The purpose of this study is to investigate the usage of science applications on the Cray Jaguar XT5 system during its two-year tenure as a 2.3 PF system at ORNL. The study is based on the corpus of MOAB and ALPS log data accumulated over this period of time. The objective of this analysis is to give a better understanding of how Jaguar is being used as a representative petascale system to produce science and also to better inform the discussion of requirements for future multi-petascale and exascale systems.

The remainder of this study is organized as follows. After discussing the study approach, we begin with a high-level view of Jaguar utilization over time. We then analyze utilization by science area. Following this we investigate usage by science application and then study in further detail the usage patterns of the specific applications that have been heavily used on Jaguar.

II. APPROACH

The system used for this study is the Jaguar Cray XT5 platform located at Oak Ridge National Laboratory. This system contains 18,688 compute nodes, with each compute node containing two hex-core 2.6 GHz AMD Opteron 2345 Istanbul processors and 16 GB of DDR2-800 memory. Each node contains a SeaStar 2+ router with peak bandwidth of 57.6 GB/s. The full system contains 224,256 processing cores, 300 TB of memory, and a peak performance of 2.3 petaflops.

Jaguar usage is aimed at capability computing, to run science problems too large to be run on smaller-scale HPC systems. As such, Jaguar's usage policy emphasizes applications and jobs that use a large fraction of the system.

The data for this study are taken from a period of approximately two years, from November 2009 through September 2011. Two primary sources of data are used. First, the MOAB scheduler [22], which underlies the PBS job queuing system [25] and reserves compute nodes for executing an application, stores entries in a database with information on every PBS job launched on the system. Second, the ALPS scheduler [1] stores entries for every execution of an "aprun" command which is used to launch an executable code within a PBS job. By matching the entries of these two databases, it is possible to extract detailed data for every execution instance of an application, including job size and duration, user and project, and filename of the executable code.

Jaguar does not have any automated method for identifying the science application name associated with each executable file that is run. Thus, a manual process was

used, utilizing data from multiple sources, including a secondary database containing the link commands used to build each parallel executable file built on Jaguar, a cross-reference process based on the user name and project associated with each job execution, and in some cases interviews with developers and users to clarify the underlying application being used. This process was used to identify and validate the application names associated with the most heavily-run codes on Jaguar, forming the basis of the application analysis given below.

III. System Utilization

Figure 1 shows Jaguar usage over the twenty-three month reporting period. The theoretical peak usage per month is shown as the total available uptime assuming no outages. From this, the scheduled uptime is obtained by excluding time for scheduled outages, actual uptime excludes all outages, job scheduler shows time spent executing jobs as reported by the MOAB scheduler, and analyzed application utilization is time spent in "aprun" execution launches as reported by ALPS logs that can be successfully matched against MOAB logs. Over this period, actual uptime was in excess of 90% of the theoretical maximum, and 83% of this time was spent executing user jobs. These figures indicate high usage of Jaguar compared to typical values for large HPC systems. Furthermore, the analyzed application utilization is 86% of the job scheduler utilization, indicating that the data used for studying application usage is representative of the total time actually used.

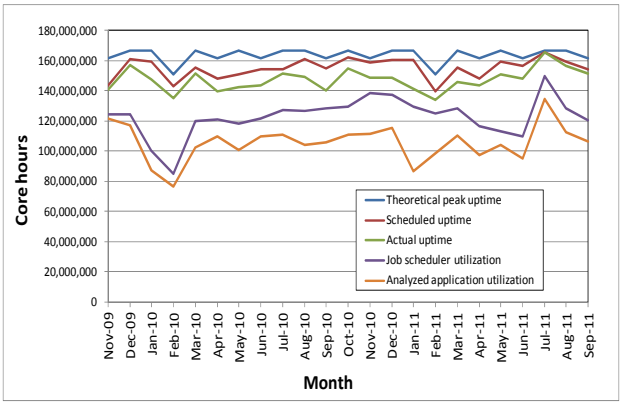

Figure 1. System usage over time.

IV. Utilization by Science Domain

Figure 2 shows the proportion of core-hours spent for each science domain on Jaguar. Of the twenty-three science domains shown, the top five science domains consume half of the total core-hours used: chemistry, fusion, computer science, materials and astrophysics. Furthermore, the top nine science domains consume 75% of the total. Jaguar supports a diverse portfolio of science domains with a few heavy-usage science domains. General-purpose systems such as Jaguar that support diverse science codes, models and algorithms must deploy well-balanced hardware to perform well for the hot spots of the targeted algorithms.

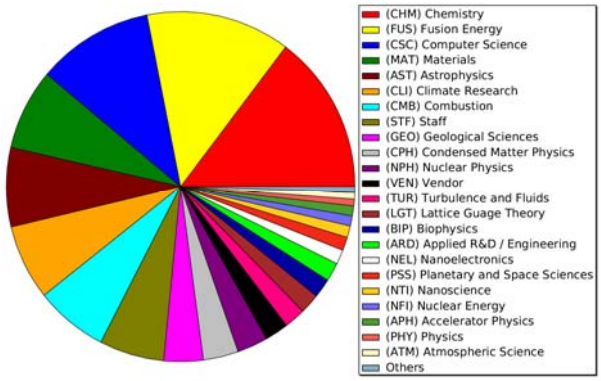

Figure 2. Usage by science domain.

Figure 3 shows usage over time for each science domain. Depending on science need, some science areas demonstrate a relatively uniform consumption of resources over time, while others show a more punctuated usage pattern, whether due to the science workflow pattern or due to the calendar year schedule by which much of the computer time on Jaguar is awarded.

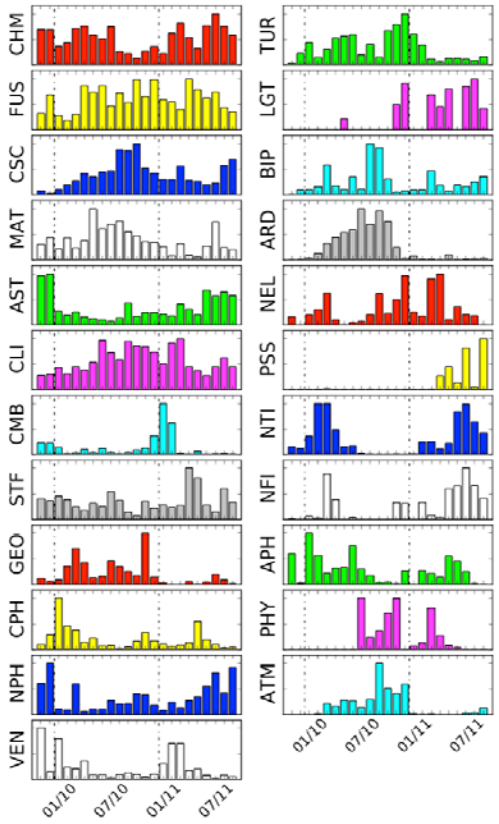

Figure 3. Science domain usage over time.

V. Utilization by Application

We now consider usage of applications as derived from analysis of the ALPS log data. The approach here is to

classify the application runs during this time period by several characteristics, based on the number of core-hours spent, for each choice of a given quantity of interest. Figure 4 shows core-hours spent in each application in a cumulative graph ranging from the most heavily-used applications to the less-used ones. The top 20 applications consume 50% of the total core-hour usage, and the top 50 applications consume 80% of the total. Thus the graph has a "short tail." This is in agreement with how computer time is awarded on Jaguar, emphasizing a comparatively small number of high-impact projects. This is strategic, since it is difficult to deploy substantial developer support across a very large portfolio of projects, as leadership systems become increasingly complex and challenging to program efficiently.

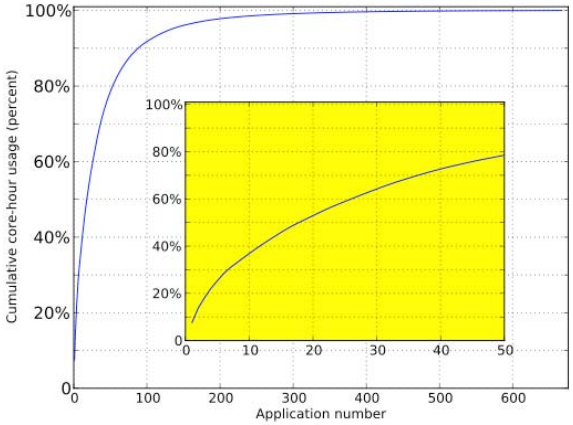

Figure 4. Core-hour usage by application.

Figure 5 shows how Jaguar core-hours are used in terms of job size. Henceforth, by "job" we refer to an instance of running an application using the "aprun" command. A full 43% of core-hours are spent in jobs run on 20% or more of the full system, while 15% of core-hours are spent in truly large jobs that run on 60% or more of the full system. Thus Jaguar is highly used for leadership-class jobs, but it is also well-used across the full spectrum of job sizes.

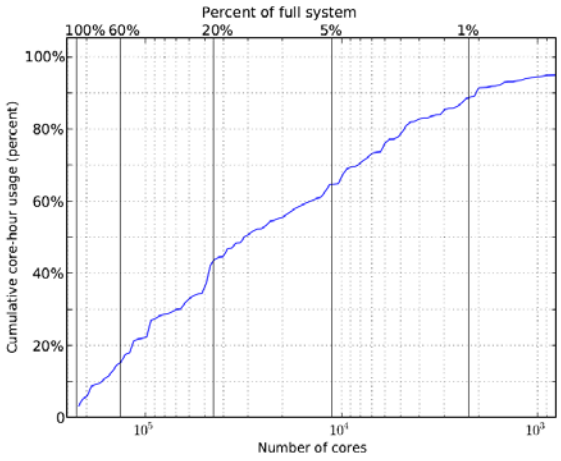

Figure 5. Core-hour usage by job size.

Figure 6 shows the core-hour usage for jobs of a given duration. During the reported period, the Jaguar scheduler set a 24 hour maximum time limit on user jobs. A total of 88% of core-hours were spent on jobs of 12 hours or less, and 50% of core-hours were spent on jobs of 6 hours or less. This should be compared to a MTTF of 65 hours for the period, and average system or node failure every 35 hours. The data suggest users are effectively running jobs within constraints of scheduler time limits and substantially below system failure rates. The capability of applications to run for short durations may be of growing importance as future systems are expected to have increasing failure rates as the number of parts for high-end systems increases.

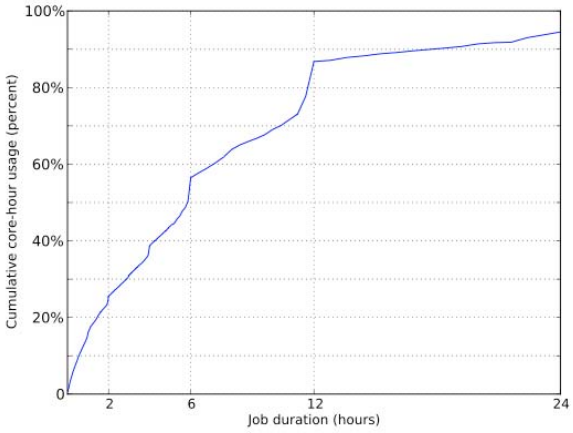

Figure 6. Core-hour usage by job duration.

Figure 7 is a scatter plot showing the relationship between job size and job duration. Jobs are binned by job size, and for each bin the weighted average job duration is plotted. Here, all averages are weighted by the core-hour consumption to tie the statistics more directly to Jaguar resource consumption. The incidence of data values is fairly well-distributed across the range. Small jobs are limited by the scheduler to run no more than 12 hours, whereas very large jobs are unlikely to run for a very long time due to the high core-hour resource cost to run such jobs.

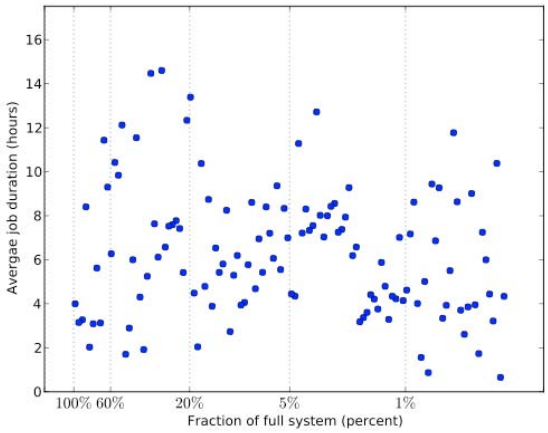

Figure 7. Job size vs. job duration.

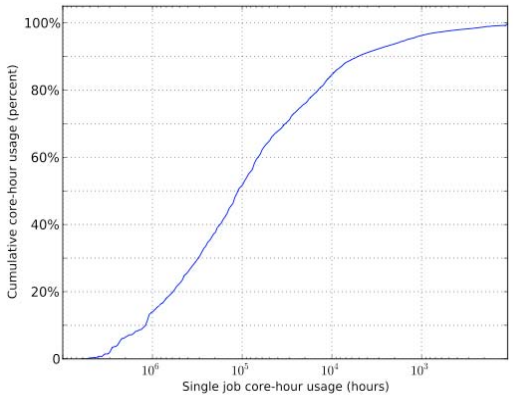

Figure 8. Core-hour usage by job core-hours.

Figure 8 shows the number of core-hours spent in jobs of a given core-hour usage value. The qualitative features are similar to those of Figure 5, with uniform usage in the mid-range of jobs consuming 10,000 to 1 million core-hours, while usage for very small and very large core-hour usage jobs is less.

In summary, over the time period studied, Jaguar workload consisted primarily of a comparatively small number of applications, with execution of jobs across the entire range of job sizes including leadership-class jobs, run successfully within the constraints of system uptimes and scheduler limits.

VI. Utilization for Selected Applications

To get a better understanding of the usage pattern of specific applications, we now restrict to a study of twenty-two selected applications taken from the top-used codes as well as several other strategically important codes. These applications account for 50% of the entire analyzed application runtime on Jaguar. A list of these applications is given in Table I.

TABLE I. Selected Applications

Application	Primary Science Domain	Description
NWCHEM	Chemistry	large scale molecular simulations
S3D	Combustion	direct numerical simulation of turbulent combustion
XGC	Fusion Energy	particle-in-cell modeling of tokamak fusion plasmas
CCSM	Climate Research	climate system modeling
CASINO	Condensed Matter Physics	quantum Monte Carlo electronic structure calculations
VPIC	Fusion Energy	3-D relativistic, electromagnetic, particle-in-cell simulation
VASP	Materials	ab-initio quantum mechanical molecular dynamics
MFDn	Nuclear Physics	a Many Fermion Dynamics code
LSMS	Materials	Wang-Landau electronic structure c multiple scattering
GenASiS	Astrophysics	AMR neutrino radiation magneto-hydrodynamics
MADNESS	Chemistry	adaptive multi-resolution simulation by multi-wavelet bases
GTC	Fusion Energy	gyrokinetic toroidal momentum and electron heat transport
OMEN	Nanoelectronics	multi-dimensional quantum transport solver
Denovo	Nuclear Energy	3-D discrete ordinates radiation transport
CP2K	Chemistry	atomistic and molecular simulations
CHIMERA	Astrophysics	modeling the evolution of core collapse supernovae
DCA++	Materials	many-body problem solver with quantum Monte Carlo
LAMMPS	Chemistry	molecular dynamics simulation
DNS	Fluids and Turbulence	direct numerical simulation for fluids and turbulence
PFLOTRAN	Geological Sciences	multi-phase, multi-component reactive flow and transport
CAM	Climate Research	global atmosphere models
QMCPACK	Materials	diffusive quantum Monte Carlo simulations

Many of these codes are well-known and widely used on multiple HPC platforms. Figure 9 shows the number of Jaguar projects and science domains using each of these applications. On average, each code is used by four projects and two science domains. The code with the top number of science domains is VASP, followed by LAMMPS and NWCHEM. Broadly-used codes offer more opportunity for leveraging code improvements to many target user communities.

Figure 10 shows the core-hour usage of each of the selected applications. Of the 2.3 billion core-hours studied, NWCHEM is the top user with 7.5% of the total, followed by S3D with 6.3%.

The scalability characteristics of the applications are presented in Figure 11. For each application, three values are presented: the largest core count ever run, the core count for the single job consuming the most core-hours, and the weighted average core count over the reporting period. Over half of the codes have been run at least once at 90% or more of the full system, though typical usage often less. Over half the codes are run on average at 20% of the full system or more. Thus, many of the codes are scaling up to use significant fractions of the full system.

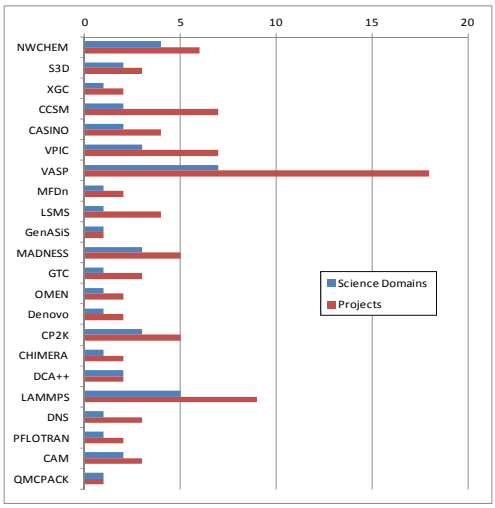

Figure 9. Selected applications community usage.

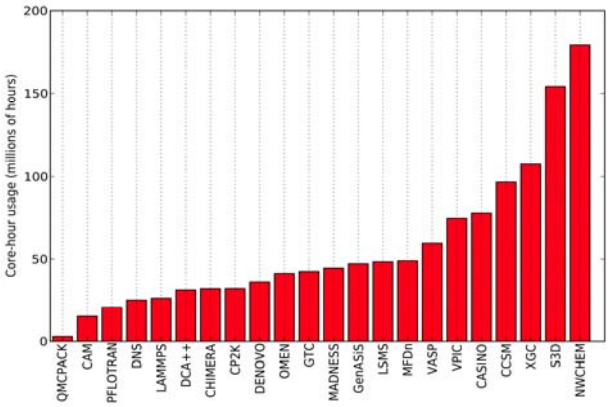

Figure 10. Selected applications core-hour usage.

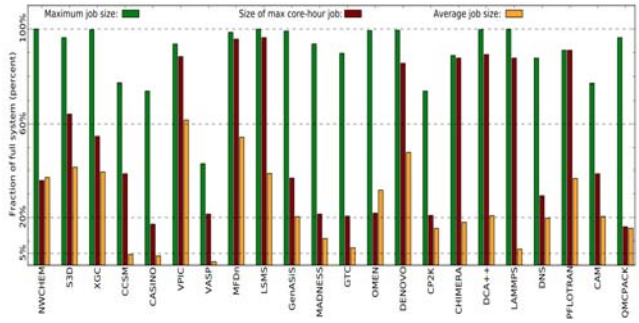

Figure 11. Selected applications scalability characteristics.

The relationship between average job size and total core-hour usage is shown in Figure 12. The top three applications as well as numerous others are commonly used at high core counts, while some others are typically used at lower core counts. It is advantageous for the most heavily-used codes to be as scalable as possible as leadership-class HPC systems continue to scale up in size.

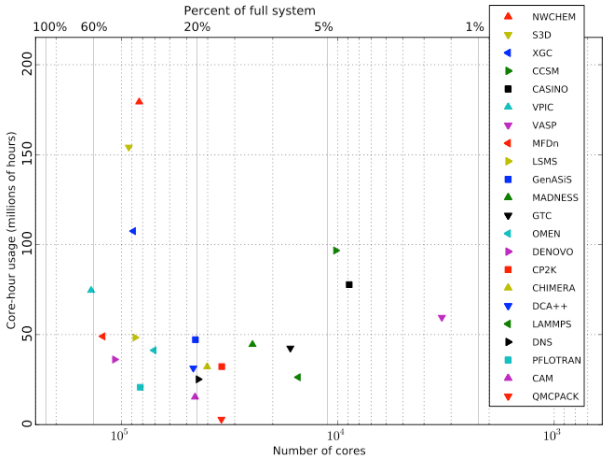

Figure 12. Average job size vs. core-hour usage.

Figure 13 gives a more detailed view of scaling behavior by representing the fraction of core-hours for each application spent in several job size brackets, including <1% of total system size, 1-5%, 5-20%, 20-60% and >60%. The usage patterns here are diverse, with some codes heavily used at high core counts and others emphasizing lower core counts. Some codes such as DCA++ and QMCPACK are highly scalable and generate science results at higher core counts but are also heavily used at lower core counts. Codes executing typically at lower core counts might be candidates for greater effort to improve scalability characteristics going forward.

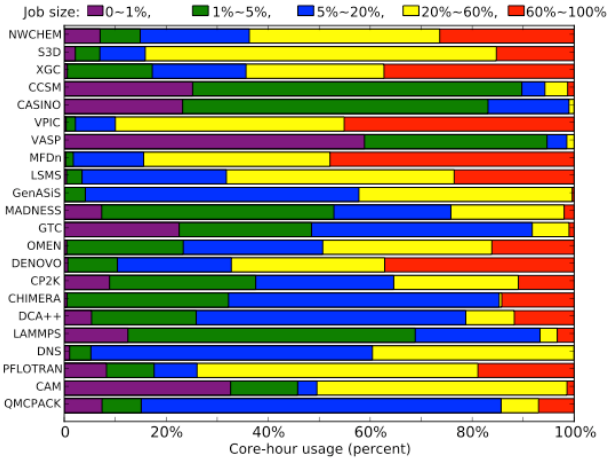

Figure 13. Usage for selected core-count ranges.

The job duration characteristics of applications are shown in Figure 14. For each application the figure shows the

longest case ever run, the duration of the job with largest core-hours, and the weighted average duration. Though one-third of applications have run for the maximum length of 24 hours, half of the codes run on average six hours or less. Furthermore, some codes such as VPIC and S3D which run for long durations on average are not required to do so since they have checkpoint/restart capabilities.

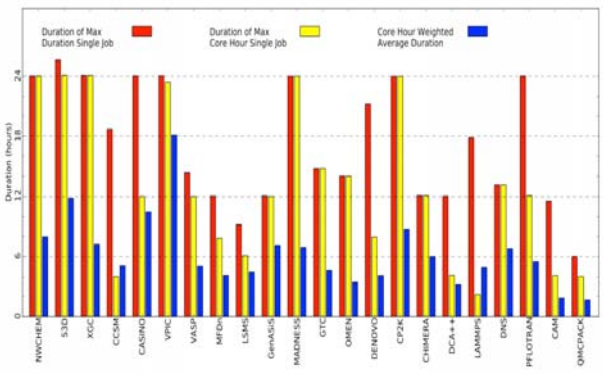

Figure 14. Selected applications job duration characteristics.

The core counts at which each of these applications runs for long durations are demonstrated by the heat map in Figure 15. The brightness of the respective square shows the average duration of jobs run with that application at that core count range. S3D, XGC, VPIC and PFLOTRAN have long-running jobs at high core counts. Jobs running for long duration at high core counts are most susceptible to single-node failures and thus a possible concern if failure rates increase on future systems due to increased numbers of components.

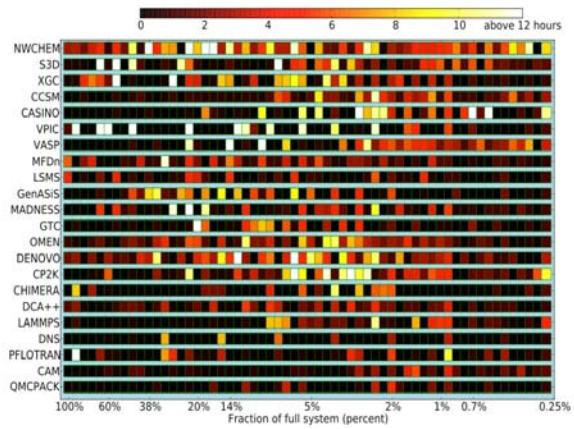

Figure 15. Selected applications job size of long jobs.

The workflow patterns illustrating how these applications are used are shown in Figure 16, which shows the number of application launches via an invocation of the "aprun" command for each code. Of the 4.4 million total application launches tracked, the figure shows the number of application launches for each application as well as the number of launches having core count at least half the weighted average job size for that application, the latter being a statistic that filters out an excess of small jobs. The most frequently run codes are VASP and LAMMPS. Log data suggests that frequently run jobs are often submitted by an automated process which may be part of a larger experiment, which might be considered another form of parallelism. At the other extreme, VPIC runs a much smaller number of very large-scale jobs.

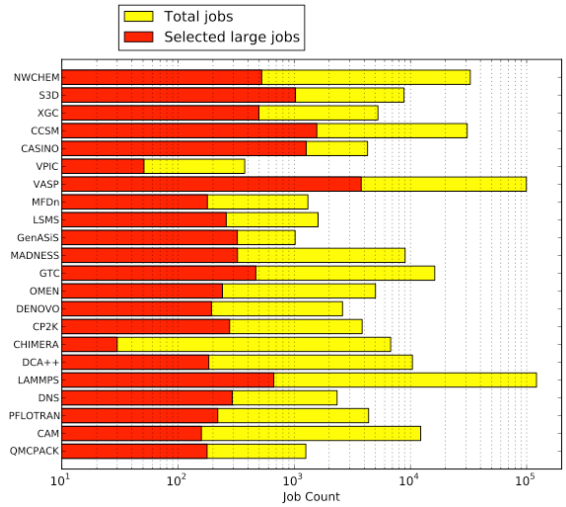

Figure 16. Selected applications number of application launches.

VII. Conclusions

We draw the following conclusions.

- Jaguar has had a productive run as the world's first petascale system for open science.
- The workload of Jaguar has been diverse, whether by science domain, average job sizes, typical job durations or usage patterns.
- Usage is dominated by a relatively small number of applications.
- Many application codes are scaling to a large percentage of the machine, and workflow patterns show that application teams are running their codes effectively within the constraints of system uptime characteristics.
- There is more work to be done. Applications are at various points in the scale-up curve, and some applications require concentrated effort to improve their performance and scaling behaviors, particularly for cross-cutting applications used across many science domains and projects.

Acknowledgment

The authors thank ORNL staff who provided input to this study, including Buddy Bland, Mark Fahey, Chris Fuson, Al Geist, Mitch Griffith, Rebecca Hartman-Baker, Joshua Hursey, Ricky Kendall, Don Maxwell, Bronson Messer, Maggie Miller, Hai Ah Nam, Jack Wells and Julia White.

This research used resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of

the U.S. Department of Energy under Contract No. DE-AC0500OR22725. This document describes research performed under contract number DE-AC0500OR22750 between the U.S. Department of Energy and Oak Ridge Associated Universities.

REFERENCES

[1] Michael Karo, "ALPS: Application Level Placement Scheduler," Cray User Group Meeting, 2006, http://www.adaptivecomputing.com/products/moab-hpc-suite-basic.php.

[2] ---, "The opportunities and Challenges of Exascale Computing: Summary Report of the Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee," Fall 2010, http://science.energy.gov/~/media/ascr/ascac/pdf/reports/Exascale_subcommittee_report.pdf.

[3] David H. Bailey, "Computing: The Third Mode of Scientific Discovery," presented at University of Newcastle, Australia, August 2009, http://crd.lbl.gov/~dhbailey/dhbtalks/dhb-grinnell.pdf.

[4] Community Earth System Model 1.0: CAM Documentation, http://www.cesm.ucar.edu/models/cesm1.0/cam.

[5] HECToR: CASINO, http://www.hector.ac.uk/cse/distributedcse/reports/casino.

[6] CESM Project, http://www.cesm.ucar.edu.

[7] Chimera Collaboration, http://astrodev.phys.utk.edu/chimera/doku.php.

[8] CP2K Project Homepage, http://cp2k.berlios.de.

[9] Thomas C. Schulthess, "The DCA++ Story,", Leadership Computing Facility Seminar, Oak Ridge National Laboratory, 1/30/2009, http://www.nccs.gov/wp-content/training/seminar_series/dcaStory.pdf.

[10] Thomas M. Evans, "Denovo: A New Parallel Discrete Transport Code for Radiation Shielding Applications," "http://info.ornl.gov/sites/publications/Files/Pub22424.pdf.

[11] DNS, Georgia Institute of Technology, P. K. Yeung, http://soliton.ae.gatech.edu/people/pyeung.

[12] C. Y. Cardall, A. O. Razoumov, E. Endive, E. J. Lentz, A. Mezzacappa, "Toward five-dimensional core-collapse supernova simulations," Journal of Physics: COnference Series 16 (2005), pp. 390-394, http://iopscience.iop.org/1742-6596/16/1/053/pdf/1742-6596_16_1_053.pdf.

[13] Scott A. Klasky, "Gyrokinetic Particle Simulations of Fusion Plasmas," http://computing.ornl.gov/SC08/documents/pdfs/Klasky_GTC.pdf.

[14] Peter Kogge, ed., *ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems*, DARPA, 2008, http://users.ece.gatech.edu/~mrichard/ExascaleComputingStudyReports/ECS_reports.htm.

[15] Peter Kogge, "Update on Current Trends and Roadblocks," 2011 Workshop on Architectures I: Exascale and Beyond: Gaps in Research, Gaps in our Thinking, ORAU, 2011, http://www.orau.gov/archI2011/presentations/koggep.pdf.

[16] "DOE Leadership Computing INCITE Program," http://www.doeleadershipcomputing.org/incite-program.

[17] 2011 INCITE Awards, http://science.energy.gov/~/media/ascr/pdf/incite/docs/2011_incite_factsheets.pdf.

[18] LAMMPS Molecular Dynamics Simulator, http://lammps.sandia.gov.

[19] M. Eisenbach, C. G. Zhou, D. M. Nicholson, G. Brown, T. C. Schulthess, "Thermodynamics of magnetic systems from first principles: WL-LSMS," http://computing.ornl.gov/workshops/scidac2010/papers/materials_m_eisenbach.pdf.

[20] MADNESS: Multiresolution Adaptive Numerical Environment for Scientific Simulation, http://code.google.com/p/m-a-d-n-e-s-s.

[21] Philip Sternberg, "Progress in MFDn: Mathematics and Computer Science," http://unedf.org/content/PackForest2008_talks/Day1/mfd_unedf.pdf.

[22] MOAB HPC Suite, http://www.adaptivecomputing.com/products/moab-hpc-suite-basic.php.

[23] NWChem: Delivering High-Performance Computational Chemistry to Science, http://www.nwchem-sw.org.

[24] Mathieu Luisier and Gerhard Klimeck, "Numerical Strategies Toward Peta-Scale Simulations of Nanoelectronics Devices," Parallel Computing 36 (2010), pp. 117-128, http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1513&context=nanopub.

[25] Portable Batch System, http://en.wikipedia.org/wiki/Portable_Batch_System.

[26] PFLOTRAN: Modeling Multiscale-Multiphase;Multicomponent Subsurface Reactive Flows Using Advanced Computing, http://ees.lanl.gov/pflotran.

[27] QMCPACK Quantum Monte Carlo package, http://code.google.com/p/qmcpack.

[28] David Lignell, C. S. Yoo, Jacqueline Chen, Ramanan Sankaran and Mark R. Fahey, "S3D: Petascale Combustion Science, Performance, and Optimization," Cray Scaling Workshop, Oak Ridge National Laboratory, July 30-31 2007, http://www.nccs.gov/wp-content/training/scaling_workshop_pdfs/ornl_scaling_ws_2007.pdf.

[29] Titan, Oak Ridge Leadership Computing Facility, Oak Ridge National Laboratory, http://www.olcf.ornl.gov/titan.

[30] TOP500, http://top500.org.

[31] VASP Group, http://cms.mpi.univie.ac.at/vasp.

[32] Brian J. Albright and Guy Dimonte, "Multi-scale Simulations: VPIC," Applied Physics Division, Los Alamos National Laboratory, 10/17/2007, http://www.lanl.gov/orgs/hpc/roadrunner/rrinfo/RR%20webPDFs/rr3_bja6.pdf.

[33] XGC Documentation, http://w3.physics.lehigh.edu/~xgc.