Scholar: A Campus HPC Resource to Enable Computational Literacy

Michael E. Baldwin*, Xiao Zhu†, Preston M. Smith†, Stephen Lien Harrell†, Robert Skeel‡, Amiya Maji†
{baldwin, zhu472, psmith, sharrell, rskeel, amaji}@purdue.edu
Department of Earth, Atmospheric, and Planetary Science*, Research Computing†, Department of Computer Science‡
Purdue University, West Lafayette, Indiana, USA

***Abstract*—To teach the computational science necessary to prepare STEM students for positions in both research and industry, faculty need HPC resources specifically tailored for their classrooms. Scholar was developed as a large-scale computing tool that faculty can use in their classrooms to teach HPC as well as scientific principles and experimentation. In this paper, we discuss the pedagogical need for a campus-wide HPC teaching resource and outline how such a resource was implemented at Purdue University.**

I. Introduction

Comprehensive cyberinfrastructure (CI) has radically transformed the way scientific discoveries can take place. CI-powered methods and tools are applied to a wide range of disciplines to enable new exciting research topics and create new opportunities for multidisciplinary collaborations. Efficient computing resources permit exploration of complex phenomena that can never be observed or replicated by experiment. Recently, data-intensive science has emerged as, considered by many, the fourth paradigm of scientific discoveries [10]. Scientific instruments, simulations and sensor networks are generating more data than ever before and the rate of increase is accelerating rapidly. Visualization technology contributes to the understanding of data across disciplines by opening up insight into the complex relationships that exist within the data. For instance, today's scientists and engineers combine massive amounts of complex data from various resources with cutting-edge data exploration and analysis techniques to come up with new solutions and research methods, e.g. in climate change and genetic research.

Many campuses have established advanced cyberinfrastructure which encompasses high-performance computing and storage systems and fast networks among other resources for their research enterprises. While more and more scientific and engineering domains utilize computational tools to advance our knowledge and understanding of the world, an emphasis on teaching and training the next generation of researchers and workers to use these tools is lacking on university campuses. Through interactions with faculty within and outside our own institution, it is clear that many students in various science and engineering domains lack the computing-related skills, or CI literacy, needed to pursue scientific discovery and exploration in the 21st century.

EduHPC2016; Salt Lake City, Utah, USA; November 2016
978-1-5090-3827-5/16/$31.00 ©2016 IEEE

II. Pedagogical Requirements

Scientists at Purdue University have a variety of pedagogical goals requiring high-performance computing to students. These targets range from domain sciences to teaching computing and HPC itself.

A. Use in the domain sciences

1) Atmospheric Science: The field of atmospheric science has relied on high-performance computing throughout the history of the electronic computer, since Charney and colleagues executed the first successful numerical weather prediction model on the ENIAC in the late 1940s.[5] [13] Numerical weather prediction models have become increasingly sophisticated as computing power has grown over time, today providing 100s of ensemble forecasts across the globe at high temporal frequency and spatial resolution. Sophisticated Earth-system models are executed to simulate many centuries in order to allow for understanding of historical and future changes in the global climate. Atmospheric data volumes have been growing at an explosive pace, especially during the recent decades as radar and satellite technology has evolved significantly. Sophisticated visualization techniques have been utilized over several decades to help atmospheric scientists analyze and understand these data from observations and simulations. Since many sectors of the global economy are sensitive to weather conditions and changes in the climate, valuable insight can be obtained through the combination of weather/climate data with information from other sectors, such as agriculture, transportation, finance, and utilities [6]. Data-driven computing and scientific visualization are expected to become even more critical going forward as data volumes, velocity, and variety continue to expand in the future. Given this critical connection between atmospheric science and high-performance computing, students coming into the field of atmospheric science need significant education in computing in order to succeed. However, these students are often intimidated by high-performance computing. They do not consider themselves as programmers, and often struggle in traditional computer science programming courses. While students are typically comfortable using digital technology, they do not have confidence in creating new technology or feel that they are capable of developing applications for their own data analysis, simulation, or visualization purposes. Many students do not have direct access to high-performance computing

resources and associated tools. These challenges must be overcome in order to prepare atmospheric science students to contribute to data-driven decision-making across numerous sectors of the economy.

Computer literacy is at the core of the field of atmospheric science. In order to serve the needs of atmospheric scientist students, high-performance computing must be incorporated throughout the curriculum at both the undergraduate and graduate levels. At Purdue, we have been working to address these issue by developing several courses that incorporate scientific computing that have been taught at various levels, from graduate-level courses to sophomore-level introductory labs. In these courses, we have used Scholar to provide students with access to high-performance computing resources and scientific computing tools, such as model simulations, data analysis, and visualization.

EAPS 59100 - High-performance Computing in Atmospheric Science We have offered several versions of this course over the past few years, mainly related to Purdue's entry into the Student Cluster Competition, a part of the annual Supercomputing conference. For example, in one version of this course, undergraduate computer science majors as well as graduate students in EAPS worked together on large group projects that involved high-performance computing hardware and developing state-of- the-art analysis tools that an atmospheric scientist would find quite useful. Topics covered include the science and numerical methods behind atmosphere and ocean models, molecular dynamics simulation, Bayesian inference of phylogeny, and cosmological simulation to practical issues of optimization of code on a variety of computing hardware systems.

The scientific applications were built and executed by the students on a small cluster system, similar to the Scholar system.

EAPS 59100 - Physical Parameterizations in NWP This graduate-level, multidisciplinary course was developed to address the need of graduate students to gain understanding of the physical processes used in state-of- the-art numerical weather prediction and climate models. These parameterizations are included in sophisticated atmospheric models to account for such physical processes as: radiation, clouds, precipitation, turbulent mixing, surface fluxes of heat, moisture, and momentum. Students are provided with hands-on experience using numerical models in a high-performance computing environment to allow increased understanding of how these physical parameterizations operate.

EAS 43400 - Weather Analysis and Forecasting This is a senior-level course where the students apply advanced topics related to weather analysis and forecasting. The primary steps in the "forecast process" are emphasized, which include: analysis of the current 3-D state of the atmosphere, forecasting techniques, communication of forecast information, and evaluation of forecast performance. Students in this course utilize weather data visualization software on a regular basis to analyze and diagnose weather systems, calculating a variety of variables to understand the behavior of large-scale weather systems. The students also run simulations using sophisticated numerical weather prediction models on high-performance computing resources, such as Scholar, in order to obtain valuable experience with these weather prediction tools.

EAS 23000 - Laboratory in Atmospheric Science The objective of this lab is to gain an understanding of the processes in the atmosphere and how scientists use computers to analyze weather data. This course also includes research experiences for sophomore-level students, where the collect and analyze data to study the accuracy of weather forecast information from different sources. Through the use of Scholar computing resources students analyze results from numerical model simulations, use data analysis tools to identify and evaluate different forces within the Earth-Atmosphere system, synthesize and present the results of their research. One of the main expected benefits of this course is that development of the students' understanding of the nature of science will be accelerated, and their computer literacy skills will be enhanced.

2) Additional Domain Science Usage: **Animal Sciences** Third-generation DNA sequencing technology has been adapted for a variety of assays that can be used by many life sciences disciplines. The broad range of tools for analysis of sequence all require advanced computing due to the size of most eukaryotic genomes and amount of data that can be generated. However, there is a substantial skill gap because many of the life sciences disciplines that can use these new methods have not traditionally needed computations skills beyond the personal computer. By adopting cluster use in his course project for the last three years, one professor from Animal Sciences has experienced the dramatic changes that would help fill the gap. Students learned to write BASH scripts for data retrieval and the Tuxedo suite of programs (Tophat/Cufflinks)[21]. They also made DNA alignment tracks that can be visualized on the major genome browsers. Finally, they used the cummeRbund [7] package to visualize and interpret the gene expression results. Committing powerful computing resources for instruction and using them in life sciences courses will greatly improve students' computing skills and will enable the upcoming scientists to more fully utilize the genomic resources and technologies that are available.

Business Data Analytics Business data analytics is transforming the way businesses operate on a day to day basis. From analysis of streaming data to targeted marketing strategies, businesses have a need to quickly analyze and make decisions on data they have collected. With the massive amount of information available, business data analytics, can be better framed as data science for business, it is the field that allows businesses to best utilize and analyze their data. Traditional HPC systems use batch mode as the basis for most things, in business field interactivity is needed because development and real-time adjustments become crucial to success. The ability of a computing resource, to interactively teach students R via RStudio Server, visualization via Shiny Server and Tableau, and big data systems such as Hadoop and Spark, gives students an advantage to make real time

changes and exposure to HPC systems which will help them find success post graduation.

Environmental Engineering Critical infrastructure systems underlie the economic prosperity of every society. These systems include energy systems, water systems, transportation, healthcare, information and communications technology, security, and financial services. Ensuring resiliency of these highly complex, interactive and interdependent systems is of utmost importance, due to the essential services that they provide to our society. By harnessing the ever-increasing volumes of data recorded by the users and operators of these critical infrastructure and leveraging advanced machines learning algorithms, researchers are able to make data-driven decisions that enhance the resiliency of these systems. Students learn and evaluate impacts of climate extremes and climate change on the reliability and resiliency of complex system. However, without a powerful resource dedicated to teaching, students cannot run the analysis with real-world data.

Food Science Computer-aided modeling and data analysis is not a conventional weapon for food scientists, but it is increasingly expanding in food research. When food researchers have to extract the size of food nanoparticles from Dynamic Laser Scattering data, they start to use complex inverse algorithms having a heavy computational component. Given the complexity of food materials, the solution is not trivial and parallel computer must be associated to the algorithms used. Application of molecular modeling in food biomaterials is more recent but it has served to elucidate self-assembling structures involving food molecules such as proteins, starch and fatty acids. [19] Professors Campanella and Corvalan in Agricultural and Biological Engineering and Food Science Departments have recently developed a course (ABE/FS 591, Numerical Methods for Biological Sciences) that covers many numerical tools that are unfamiliar to food researchers. The biggest challenge for them is that their students do normally not possess the necessary scientific computing skills.

Fluid Dynamics Powerful computational resources allow obtaining very rapid computer-aided solution to problems commonly found in engineering such as integration of stiff ODEs, inversion of large systems of linear equations. While advanced computing technology, including GPGPU, is generally employed in his research field, majority of the students, surprisingly, lack necessary computing literacy in their academic studies. The main cause for the lag is not the difficulty of the subject, per se, but the lack of exposure to such task from the early stages of their education. Often, training in computational tools is integrated within existing courses and given only marginal attention. Fully dedicated training on computational methods computing resources can significantly improve this situation. Being able to exploit computational resources in an informed and efficient way is rapidly becoming part of the common skill set of a student that wants to be competitive in both academic and industrial work marketplace.

B. Use in the computing sciences

CS 501 - Computing for Science and Engineering

Computer Science 501, Computing for Science and Engineering, is a Purdue service course, typically taken by first-semester graduate students. It is primarily a second course in programming that exposes the students to concepts and tools likely to be useful in their research, and coursework.

The assignments consist of relatively short pieces of code; the goal is to expose the students to a number of techniques and tools that they might find useful now or later. Students are encouraged to do the programming assignments on the same computers that they expect to use for their research. Efforts are made to accommodate Linux, OS X, and Windows.

To keep course logistics manageable, particular software tools are specified. Constructing a correct working program is the prime consideration, so the emphasis is on using a language requiring moderate programming effort.

Python, with Numpy[22] and Matplotlib [11] extensions, is chosen for its broad applicability and popularity. At the same time, performance is so often an issue, so the use of a compiled programming language is advocated for creating dynamic libraries of computational kernels. C is chosen due to its universality.

Parallel computation is often needed. MPI and OpenMP remain the most popular tools for doing this. Students write C code using OpenMP. However, coding MPI library calls in C is cumbersome. The Python extension mpi4py provides a faithful rendition of MPI library but with less drudgery, so this is used to expose students to MPI.

In practice, it can be difficult to install the required software, particularly implementations of MPI. For this reason, all students are given access to the Scholar cluster at Purdue. This system provides an Anaconda [1] package that includes Python 3.5, Numpy, Matplotlib, and mpi4py. Also available is conveniently packaged software for writing Code using OpenMP and MPI.

An additional benefit of the Scholar cluster is to demonstrate the use of a Linux cluster, which is a primary means of performing demanding computations, and at Purdue many of the clusters are practically clones of Scholar. Some time is given in lecture demonstrating the basics of the bash shell, remote access, and the PBS queueing system.

CS 348 - Database Information Systems

Undergraduate and graduate courses in Computer Science have used Scholar since Fall 2015. A virtualized on-premise cloud environment has helped instructors to build Cloudera (a big-data framework) clusters almost at will and provision them according to class size. Using this paradigm, computer science students can develop big data software in ways that are not possible on batch HPC systems.

The ability to run virtual machines allows different courses to run separate Operating Systems and Software. For example, Computer Science courses can use Linux virtual machines for programming, while Earth and Atmospheric Sciences courses can use Windows virtual machines to run ArcGIS simulations) all on the same shared hardware, with customized clusters for individual courses created at the click of a button. Moreover, the hardware resources can be re-provisioned for different

courses throughout the year while preserving the data from previous courses.

1) Other Computing Science uses: **Data Science** Statistics has applications in almost every field. Big data is among the most promising research trends of the decade, drawing attention from every segment of the market and society. This trend is only going to become more important as time goes by, which has rapidly changed the way we teaching statistics and data analysis. Purdue operates an NSF-funded program connecting academics, research, professional development and residential life, designed to help statistics students overcome the "sophomore slump." He also developed a data analysis course for new undergraduate students. Each year, over 100 undergraduate students are trained about key concepts in data analysis, right at the start of their undergraduate years. They learn about R, visualization, SQL, UNIX, bash shell, XML, and about computing in an external, massive cluster environment. Instruction-dedicated cyberinfrastructure will also enable Purdue to introduce an all-new Data Science undergraduate major at Purdue that could accommodate many hundreds of undergraduate students.

Scientific Visualization Data visualization presents complex information in an easy-to-understand format that enables users to understand, use, communicate, and take action. Visualization plays increasingly important role in informed decision-making, data analysis, providing explanations of complex data sets, detection of trends and patterns, and storytelling. Many students and some faculty are, however, not aware of the data visualization process, its value and purpose or the benefits of visualization in academia, research, and industry. In order to prepare the next generation of researchers and scientists to make transformative and innovative discoveries in a data driven world, exposure to the process, tools and techniques of data visualization must begin early at the undergraduate level. Not surprisingly, Purdue is going to offer a new approved undergraduate major in data visualization in spring 2017. Having access to resources that combines interactive visualization and large-scale data capacity is critical for the success of visualuzation students.

C. Use in Applied HPC Education

In addition to the need for a teaching resource in the classroom there is growing need in extra curricular activities and applied courses. Integrating HPC into these project-based activities is crucial in order to introduce students to HPC in a setting that allows for exploration. These types of activities can inform students' career decisions and professional interests as much as any classroom setting. The Scholar resource has supported many of these activities such as the Student Cluster Competition[9], Hyperloop competition[2] and Formula SAE [15].

III. The Scholar Resource

A. Community Cluster Program at Purdue

Since 2008, Purdue University has operated a research cyberinfrastructure built around"community clusters" [14] in which faculty investors partner with central IT to build shared computing systems. Faculty are freed from inefficiencies of building their own HPC systems, and As McCartney [14] describes, the institution gets a

> ... business model to support collaborative funding that empowers faculty and promotes their research while fully leveraging the institution's investment in facilities and IT staffing.

An investing faculty member is provided with a dedicated queue with access to as many nodes as he has purchased, and is able to access unused nodes purchased by other partners. Additionally, each user is assigned their own large parallel scratch space, given access to a tape archive, and can use the Research Data Depot [17] storage system for sharing data and applications.

As the community cluster model grew to become a key resource for faculty at Purdue, many sought to use their nodes in their teaching.

B. Scholar HPC Resource

Built in 2012 to address this missing link, the Scholar system leverages the community cluster cyberinfrastructure to provide access to high-performance computing for instruction.

For many instructors, this system makes it feasible for the first time to incorporate hands-on high-performance computing experiences into courses as Scholar is integrated with the campus academic framework, tuned for use in classes, and centrally supported, all of which lowers the barrier to entry and makes it easy to use by the instructors and students alike.

Early use was driven by interests from faculty in Purdue's Earth, Atmospheric, and Planetary Sciences department to provide hands-on experience to students in running weather modeling with realistic settings as they learn meteorology.

Access to Scholar is closely integrated with Purdue's student systems, to lower the barrier to use. To use Scholar in a course, an instructor simply has to provide a course number in a request, and the course's entire roster of students is granted access to the Scholar system, and automatically removed at the end of the semester.

C. System Configuration

Scholar has the capability to support teaching and student learning in a wide range of scientific domains, with a strong focus on traditional, batch oriented HPC. Specifically, Scholar leverages the infrastructure of the "Rice" community cluster. Rice is a 576-node HPC system, built with 2.6 GHz Intel Xeon E5-2660 processors. Each node has 64 GB of RAM and is interconnected with 56Gb FDR Infiniband.

By also providing access to the "Hathi" Hadoop cluster system, Scholar enables big data, data analytics and scientific visualization education in the classroom. The overall architecture of the system, software, and campus cyberinfrastructure is depicted in Figure 1.

- Large-Memory, Interactive Nodes: 4 large-memory nodes with 512GB of memory are designed for data-intensive,

interactive data analysis applications requiring large shared memory per node.

- Compute Nodes: There are 16 general-purpose compute nodes with 2 10-core Intel Haswell processors, 64GB of memory, and FDR Infiniband interconnect.
- File Systems: Students using Scholar each have 100TB of space on a Lustre parallel file system for high-performance scratch storage. Additionally, instructors may leverage the Research Data Depot for shared resources.

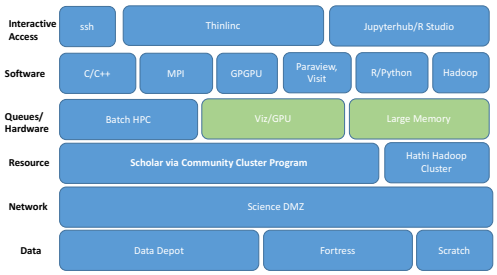

Fig. 1. Scholar System Architecture

D. Flexible User Environment

Scholar provides a diverse and flexible user environment to meet the needs of various classes. Tables I and II list frequently used software tools and applications in different scientific domains. Because Scholar shares the common infrastructure of community clusters, it benefits from continually updated application packages, compilers, communications libraries, tools, and math libraries. Software specific to classes are also installed upon an instructor's request.

Each course has the option of getting space on the Research Data Depot, where the instructor can control access permissions for the whole class through a simple web interface.The instructor may use this space to share large data sets for classroom use, installing software for student use, and for saving student output.

E. Interactive Capabilities

Although the use batch resources is typical within scientific computing, interactive use of computing plays an important role within the classroom. Interactivity and immediate feedback help professors create an interactive learning environment and increase classroom engagement. Based on feedback across campus, limited interactive usability is the primary reason for instructors not to incorporate the use of computing resource into teaching and student learning activities.

Beginning in Fall 2016, Scholar has added three software packages to provide interactivity to scientific computing: Thinlinc, [4] Rstudio Server [18] and Jupyterhub [16]. These packages align with the needs of faculty who have expressed interest in having an interactive educational scientific computing resource. As Scholar evolves, it is expected that other packages will be deployed to further diversify the capability of the resource and the scientific communities it serves.

TABLE I
SOFTWARE TOOLS ON SCHOLAR

Operating Systems, Compilers, Programming Languages	
Operating System	Red Hat Enterprise Linux 6
Compilers	Intel Parallel Studio XE 2016
Message Passing Libraries	Intel MPI, OpenMPI
OpenMP API	Vendor-supplied compilers
Other Language Support	Java, Python, Perl, TCL, R
Data Management Tools	
Data Components	HDF5, HDF4, NetCDF
Data Analysis	NCL, NCO
Databases	MySQL, PostgreSQL, MongoDB
System Resource Management	
Job Scheduling	Torque, Moab
User Environment	Modules (lmod)
Productivity Tools	
Debugging and Profiling	Intel Vtune Amplifier, Totalview
Version Control	Subversion, Git, Purdue Github
Workflow	DAGMan,Pegasus,Swift,Makeflow

TABLE II
APPLICATION SOFTWARE ON SCHOLAR

Biology/Bioinformatics	MrBayes, Bowtie, BWA, Abyss
Data Analytics	Hadoop, Spark, R,MATLAB
Chemistry	GAMESS, NWChem,Gaussian
Molecular Dynamics	NAMD, LAMMPS, GROMACS
Engineering	FLUENT, OpenFOAM, Abaqus, Comsol
Math and Statistics	Intel MKL, FFTW, PETSc
Environmental Sciences	CESM, WRF, SWAT, HRLDAS
Visualization	Visit, Paraview, VTK, Avizo

F. Big Data Technologies

Hadoop and Spark are two popular big data frameworks that are frequently requested by instructors. Scholar leverages Purdue's "Hathi" Hadoop [20] cluster to accommodate classes with various software requirements. Additionally, Scholar supports interactive Apache Hadoop YARN and Spark on the batch compute nodes. Students can submit and run MapReduce jobs using YARN cluster manager, monitor job progress, and view job logs and histories through the Hadoop web interfaces.

In batch mode, we will implement Hadoop YARN and Spark frameworks on the regular computing nodes by taking advantage of software modules myHadoop [12] and pbs-spark-submit [3], respectively. In addition, popular R and Hadoop Integrated Programming Environment (RHIPE)[8] is supported to provide a rapid development, debug, and deployment cycle for interactive teaching and class projects. The combination of RHIPE and RStudio allows the students to leverage the power of parallel, distributed computing from a traditional R programming environment, without worrying about the technical details of underlying implementation.

IV. IMPACT

Since 2012, adoption of Scholar has been growing steadily, with instructors from many departments incorporating it into their curricula, including Computer Science, Aeronautical and Astronautical Engineering and Animal Sciences. Usage statistics, including CPU hours and number of jobs, show a trend

TABLE III
COURSES USING SCHOLAR PER SEMESTER

Semster	Courses
Fall 2013	2
Spring 2014	4
Fall 2014	5
Spring 2015	8
Fall 2015	10
Spring 2016	16

of consistently increasing demand for use of the cluster in teaching scientific computing. Figure 2, Table III.

Since its inception, Scholar has shown impact on student development as future researchers. Animal Sciences Professor Christopher Bidwell used Scholar for an animal functional genomics course over three semesters and he shared some interesting story: "The students learn a lot about their own computing skills and they begin to think differently about research possibilities. They are not as intimidated by projects with more demanding analysis. In each semester, there have always been one or two students that really get into the computing work and the experience has changed their minds about the types of graduate training options they consider."

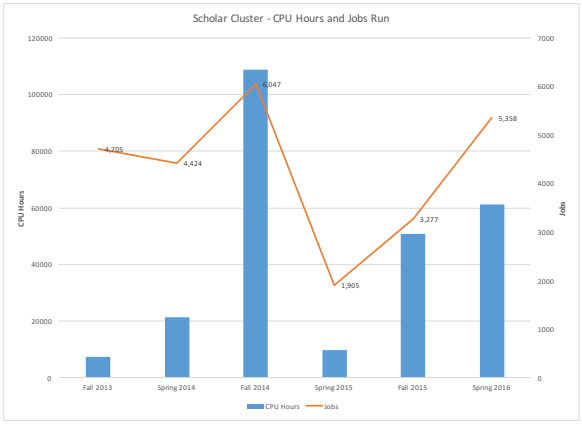

Fig. 2. Scholar Usage - CPU Hours and Jobs per Semester

V. CONCLUSIONS

The Scholar HPC resource has proven to be an effective teaching tool for courses with computational literacy components. We have seen a steady growth in the number of courses using scholar, growing from two courses in its first semester to 16 unique courses using the resource in Spring of 2016. We observe upward trends in the number of jobs run and hours used and are adapting new modes of system use to broaden its availability to more than just batch high-performance computing.

Integration with campus cyberinfrastructure creates a path to production use of those systems and is a key advantage of the Scholar system. Students may use the system for class and progress into using the Purdue Community Clusters in their research careers. Finally, using campus authentication and integrating with learning management systems serves an important function of lowering the barrier of entry for courses that can be augmented with high-performance and scientific computing.

A. Future Work

Future work for the Scholar system will involve broadening its applicability to more than just basic HPC. Addition of batch nodes with large quantities of memory could be designated for data-intensive applications requiring large shared memory per node. Nodes configured with advanced GPUs will bring the ability of remote visualization and GPU computing to Purdue's teaching community, with applications from chemistry, molecular dynamics, and computer graphics.

Future architectures, such as advanced GPUs or next-generation coprocessors could be added to enable instruction on many-core programming, and broadening access to public and private cloud systems for instructors is also on the roadmap.

Finally, we forsee future collaboration between IT and faculty to develop metrics to measure the impact of the scholar system on students' learning.

VI. ACKNOWLEDGMENTS

Thanks to faculty partners and Scholar team members Alina Alexeenko, Gladys Andino, Walid Aref, Chris Bidwell, Vetria Byrd, Osvaldo Campanella, William Cleveland, Benjamin Cotton, Carlos Covalan, Lev Gorenstein, Michael Gribskov, Roshi Nateghi, Carlo Scalo, Lyuidmila Slipchenko, Carol Song, Mark Ward, and Boyu Zhang.

REFERENCES

[1] Anaconda Software Distribution. Computer software. https://continuum.io, 2016. [Online; accessed 02-September-2016].

[2] A. Alexeenko. Purdue University Hyperloop Team. http://www.purdue.edu/hyperloop/, 2015. [Online; accessed 02-September-2016].

[3] T. Baer, P. Peltz, J. Yin, and E. Begoli. Integrating apache spark into pbs-based hpc environments. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, XSEDE '15, pages 34:1–34:7, New York, NY, USA, 2015. ACM.

[4] Cendio. ThinLinc - a remote desktop server. https://www.cendio.com/thinlinc/what-is-thinlinc, 2015. [Online; accessed 02-September-2016].

[5] J. G. Charney, R. Fjörtoft, and J. v. Neumann. Numerical integration of the barotropic vorticity equation. *Tellus*, 2(4):237–254, 1950.

[6] J. L. Demuth, J. K. Lazo, and R. E. Morss. Exploring variations in people's sources, uses, and perceptions of weather forecasts. *Weather, Climate, and Society*, 3(3):177–192, 2011.

[7] L. Goff, C. Trapnell, and D. Kelley. cummerbund: Analysis, exploration, manipulation, and visualization of cufflinks high-throughput sequencing data. *R package version*, 2(0), 2012.

[8] S. Guha. Computing environment for the statistical analysis of large and complex data. 2010.

[9] S. Harrell, H. Nam, V. Vergara, K. Keville, and D. Kamalic. Student Cluster Competition: A Multi-disciplinary Undergraduate HPC Educational Tool. In *EduHPC '15 Proceedings of the Workshop on Education for High-Performance Computing*, 2015.

[10] T. Hey, S. Tansley, K. M. Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.

[11] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[12] S. Krishnan, M. Tatineni, and C. Baru. myhadoop-hadoop-on-demand on traditional hpc resources. *San Diego Supercomputer Center Technical Report TR-2011-2, University of California, San Diego*, 2011.

[13] P. Lynch. The origins of computer weather prediction and climate modeling. *Journal of Computational Physics*, 227(7):3431–3444, 2008.
[14] G. McCartney, T. Hacker, and B. Yang. Empowering faculty: A campus cyberinfrastructure strategy for research communities. 2014.
[15] J. Nolfi. Purdue University Formula SAE Team. https://engineering.purdue.edu/fsae/wordpress/, 2015. [Online; accessed 02-September-2016].
[16] F. Pérez and B. E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007.
[17] L. Pouchard, B. Zhang, P. Smith, A. Gasc, and B. Pijanowski. Data storage and sharing for the long tail of science. In *New York Scientific Data Summit (NYSDS)*, 2016.
[18] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA, 2015.
[19] C. Scalo, S. K. Lele, and L. Hesselink. Linear and nonlinear modelling of a theoretical travelling-wave thermoacoustic heat engine. *Journal of Fluid Mechanics*, 766:368–404, 2015.
[20] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The hadoop distributed file system. In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, pages 1–10. IEEE, 2010.
[21] C. Trapnell, L. Pachter, and S. L. Salzberg. Tophat: discovering splice junctions with rna-seq. *Bioinformatics*, 25(9):1105–1111, 2009.
[22] S. Van Der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.