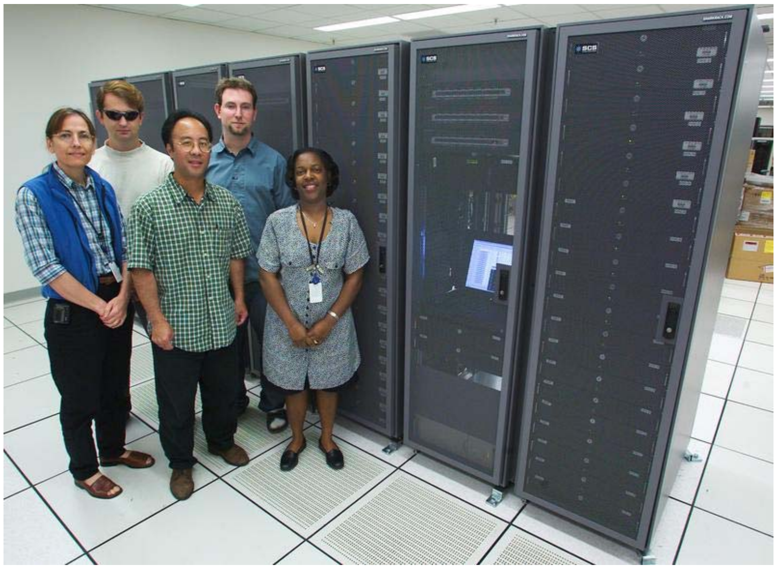
LBNL-57643

Nomination of the
Lawrence Berkeley National Laboratory
Scientific Cluster Support Program
for the

2005 Larry Sautter Award for Innovation
in Information Technology



The Scientific Cluster Support team pictured in front of the 96-processor Linux cluster belonging to the Arup Chakraborty Research Group

PROJECT LEADERS
Tammy Welcome, SCS Program Director
Gary Jung, SCS Project Manager

TEAM MEMBERS
Greg Kurtzer, Technical Lead
Jackie Scoggins
Susan James

SUBMITTED BY:
Gary Jung, UNIX Systems Group Leader
(510) 486-4894 GMJung@lbl.gov
Tammy Welcome, IT Division Deputy
Rachael Post, IT Communications
Lawrence Berkeley National Laboratory
http://scs.lbl.gov

TABLE OF CONTENTS

SIGNIFICANCE OF PROJECT

High performance computing plays an increasingly important role in advancing scientific research today and is a critical tool for keeping our scientists competitive with their peers at other institutions.

At Berkeley Lab, we have developed a successful support service to promote and facilitate the use of Linux clusters in scientific research, allowing researchers to spend more time working on their science and less time worrying about their computers. The service, called the Scientific Cluster Support program, provides a successful methodology for supporting Linux clusters that can be readily adapted for use at other institutions that want to cost-effectively meet the computational needs of their researchers.

PROJECT DESCRIPTION

I. Introduction

Linux-based clusters are a growing trend in high performance scientific computing. Concurrently, there has been a fast-growing interest in the use of Linux clusters for scientific research at Berkeley Lab. For many, a cluster assembled from inexpensive commodity off-the-shelf hardware and open source software promises to be a cost-effective way to obtain a high performance system.

Though many of the concepts sound simple, scientists have found it difficult to navigate the myriad technologies in order to arrive at a cluster configuration that will meet their needs. Similarly, they found it difficult to efficiently manage a multi-node compute cluster. Consequently, early adopters of this technology have had to invest large amounts of effort to realize the full potential of their cluster systems.

The Scientific Cluster Support program (http://scs.lbl.gov/) was developed to address the difficulties of obtaining and running a Linux cluster system. The ultimate goal of this program is to increase the role of scientific computing in Lab research projects, to introduce parallel computing to Berkeley Lab researchers and to develop efficient, cost-effective methods for managing production clusters.

II. A Brief History of Computing at LBNL

Computing has been part of the scientists' research since the 1960s and into the 1980s when central computing was the model. At the time, the Lab used CDC 6600 and 7600 supercomputers. In the 1980s, Berkeley Lab shifted towards interactive timesharing computing on DEC VAX and 8600 series systems. This was the first step towards using smaller, less expensive systems. By the mid 1990s, most of the Lab scientists were computing at their desktops and institutional support for scientific computing had almost disappeared.

In 1996, the National Energy Research Scientific Computing Center (NERSC) was relocated to the Berkeley Lab. Its arrival created greater awareness at LBNL of the need for high performance computing; however, access to NERSC supercomputers required scientists to compete nationally for computing time. As a result, NERSC was only available to a select group of scientists who were selected through the competitive allocation process. As result, there was a huge gap between the users who could get time on the NERSC supercomputers and the scientists that were limited to computing on their desktop system. We defined this discrepancy as the “mid-range computing” gap.

A mid-range computing working group (MRC) was formed in 2000 at Berkeley Lab to determine the need for mid-range computing and to formulate a plan to address this need. Findings from a March 2002 workshop[1] on mid-range computing and subsequent discussions with scientists identified a need not for a shared centralized resource, but rather for affordable centralized Linux cluster support.

The MRC workshop was followed by an application process to determine which research projects were intending to purchase a Linux cluster within the next year and could benefit from a support program. A proposal was drafted and presented to senior Lab management in November 2002. A revised proposal was approved in December 2002 and the SCS project was launched in January 2003.

III. The SCS Program

Ten research projects from seven of the Lab's scientific divisions were originally selected to participate in the four-year, $1.3M Laboratory-funded program. The applicants were selected with the criteria being that

- They were intending to purchase a Linux cluster within the coming year or they already owned a Linux cluster.
- They could describe how their research would benefit from access to a more powerful computational system.
- The selections would span the major areas of science at Berkeley Lab (General Sciences, Life and Environmental Sciences, Physical Sciences).

The goals were straightforward: Develop an efficient, cost-effective cluster support methodology and then collaborate with the selected research projects to procure, build, configure, and maintain their Linux clusters. The research projects would provide the funds for the cluster hardware and the IT Division would provide the expertise for the following:

> **Pre-purchase consulting** - Understand customer applications; determine the appropriate cluster hardware architecture, components, and interconnect; identify the required operating system, compilers, and application software. This is one of the most critical aspects of the process and any mis-steps in this process can result in a cluster that may not perform well or at all.
>
> **Procurement assistance** - Assist with developing a budget; provide consulting on procurement methods; develop specifications for the RFP, including acceptance criteria and required warranty support; and evaluate bids.

[1] Assessment of Mid-Range Computing at LBNL LBID-2443 (October 2002)

Cluster Integration - Install and configure cluster hardware, networking; cluster software, message passing interface software, scheduler, and applications software, computer security. Set up user accounts.

Ongoing systems administration and cyber security - Provide operating system and cluster software maintenance and upgrades; security updates; monitor cluster nodes; hardware troubleshooting; user accounts; scheduler configuration; and assist with compiling and running programs on the cluster.

Computer room space with networking and cooling – Host clusters in a dedicated computer room to ensure access to sufficient electrical, cooling, and networking infrastructure.

IV. Challenges

1) Scheduling. Our plan was to install six clusters in the first year. The average time to build a cluster from the initial user requirements meeting to the time the cluster goes into production is about three months. This required us to carefully develop an overlapping schedule so that the work was spread out evenly to match staffing resources. Complicating the scheduling was the fact that some research groups were dependent on the timing of the availability of equipment funds from their funding agencies.

   Another scheduling variable was customer readiness. Some of the researchers had not used a Linux cluster before; some had used large SMP machines or their desktop systems, but not a parallel Linux cluster. To help them make the transition successfully, we gave them access to our test Linux cluster so that they could prototype their applications and software environment to ensure the codes would work before starting the procurement process for their own system.

   Taken together, these factors conspired to make scheduling complex so that we had to constantly update our timelines to accommodate changes while staying within our budget.

2) Staffing. The SCS program start date came within a month after LBNL funding approval. In order to meet project milestones, we assembled the SCS project team and quickly developed specific technical expertise for supporting Linux clusters. Core expertise is critical to addressing changes in high performance computing technology.

3) Costs. The program has a $1.3M budget for the 4-year period. Staffing is set at two FTEs during the installation phase and 1.6 FTEs for ongoing cluster support in years 3 and 4. This project is funded from Laboratory overhead funds and there is a continual pressure at Berkeley Lab to reduce overhead costs. With this limited budget, it was necessary to be very cost-effective in order to stay within our budget while meeting our milestones.

V. SCS Steering Committee

A steering committee[2] was considered necessary because of the high visibility of the project. Members were chosen for their expertise or interest in the following: technical advice, representation of scientific divisions, and cluster management expertise. This body serves as a governing board to provide oversight of the implementation of the clusters and approve changes in scope, schedule, or funding to the project. The committee also serves as a means of communicating those changes to the stakeholders in the program and will participate in determining the path forward for scientific computing at Berkeley Lab.

TECHNOLOGY UTILIZED

It was important for us to develop and implement hardware and software standards that would facilitate the scaling of our cluster systems administration support efforts. This helped focus the scope of technical expertise that we had to develop and allowed the project team members to concentrate on developing more technical depth. Rather than re-invent the wheel, we leveraged the experience of other experts in the HPC community.

Equally important was the use of open source software, such as Linux. In addition to being freely available, it allowed us to make changes to the software to facilitate the integration of various hardware and software components. Moreover, if our changes improved the software, we were sometimes able to propagate the changes back into the open source code base so that everyone else could benefits from our efforts.

For our standard configuration, we felt it was important to allow users to choose the components that are important to them (e.g. CPU, memory, interconnect). However, we insisted on standardizing the software environment, including the operating system, cluster management methodology, version of MPI (Message Passing Interface), job scheduler, and cyber security.

For the key area of cluster management, we looked for existing tools that would allow us to have a scalable method for supporting Linux clusters, but we were unable to find suitable cluster tools that would meet these needs. Instead, we developed an open source cluster management toolkit called Warewulf that greatly simplifies the installation and management of clusters. Warewulf works by allowing the compute nodes to boot a shared image from the master node so that a systems administrator only needs to support the master node and the shared image for the rest of the system. The significantly reduces the amount of system administration effort required to manage several clusters.

The standard configuration includes the following components:

Hardware – Rack mounted 32- or 64-bit Intel or AMD processor compute nodes
Networking - Gigabit or Myricom Myrinet interconnect
Operating System – Red Hat Linux or Centos Linux (http://www.centos.org/)
Cluster Distribution - LBNL Warewulf Cluster Toolkit (http://warewulf-cluster.org/)

[2] SCS Steering Committee Charter http://scs.lbl.gov/html/scssc.html

MPI Implementation - LAM-MPI (http://www.lam-mpi.org/)
Job Scheduler - Sun Grid Engine 5.3 (open source version)
Monitoring Software - UC Berkeley Ganglia (http://ganglia.sourceforge.net/)
Cyber Security - Host-based security, Cisco PIX firewall and one-time use password tokens

IMPLEMENTATION TIMEFRAME

The SCS program was approved on December 4, 2002, and was launched in January 2003.

Clusters for the following groups were placed in the program in 2003:

- Arup Chakraborty Research Group – January 2003
- Ashok Gadgil and Patricia Brown - March 2003
- Mike G. Hoversten and Ernest L. Majer - May 2003
- William H. Miller - May 2003
- William A. Lester - August 2003
- Michael B. Eisen August 2003
- Steven Brenner, Paul D. Adams, Sung-Hou Kim, Stephen Holbrook - December 2003
- Priscilla Cooper and John Tainer - December 2003

The following clusters were phased into the program in 2004:

- Martin Head-Gordon June 2004
- William A. Lester - July 2004 (upgrade)
- Steven G. Louie, Marvin L. Cohen - November 2004

In 2005, a cluster for the following research group is to be added to the program:

- Gretina Detector Project - June 2005

The SCS Program is currently scheduled to continue through September 30, 2006.

COMMUNITY IMPACT

The SCS program has facilitated a number of outreach activities that include participation in:

- The National Center of Excellence for High Performance Computing Technology (NCEHPCT) http://www.highperformancecomputing.org/. SCS team members assisted with the development of community college level curriculum. The team also presented Warewulf at their second annual conference.

- The LBNL FaST (Faculty and Student Team) Summer Program, which connects a college faculty member and three students from various community colleges to participate in a summer learning program focused on high performance computing technology. SCS has hosted students for two years.

- The National Laboratories Information Technology Summit 2005 (NLIT) http://nlit2005.pnl.gov where SCS presented a talk.

The Warewulf Cluster Toolkit, developed by SCS technical lead Greg Kurtzer, is a significant contribution to the HPC community. Warewulf, featured at the Berkeley Lab Computing Sciences booth at the Supercomputing 2003 conference[3], was released by LBNL under the GNU Public License and is publicly available at http://warewulf-cluster.org. With over 20,000 downloads, there has been a widespread adoption of Warewulf by numerous academic research and HPC communities. Warewulf has been deployed at academic institutions such as: UC Berkeley's Department of Chemistry where 5 clusters using Warewulf are in production; the KAYS0 Supercomputer at the University of Kentucky that was the first supercomputer to break the $100/GFLOP price barrier in August 2003; and the University of Buffalo Center of Excellence for Bioinformatics, where they are in the process of converting their 2000+ node cluster to use Warewulf.

CUSTOMER SATISFACTION

I. Customer Survey:

A Customer Survey that solicited customer satisfaction on the quality of SCS services was conducted in May 2005. Responses to the open-ended questions are included below.

Q1. How has this program helped your science? What has been made possible as a result of this program? Do you have any publications as a result?

> *"Our cluster is at the center of everything we do - it has essentially enabled our entire research program. There are at least a dozen papers from the lab that have made heavy use of the cluster."*
>
> *"CFD simulation of large spaces are quite computer intensive, since even a coarse grid can have several thousand nodes. The number of grid nodes increases with the size of the building. Without the cluster, it will be take several days/weeks before 1 solution can be obtained. Systematic parametric studies are required to draw meaningful conclusions. Without the cluster, it is impossible for us to finish our projects on time. We have 2 conference papers, 1 article submitted to a journal and 1 journal article under preparation."*
>
> *"The availability of a machine dedicated to our science, rather than time-sharing, has allowed better development of new code and timely production of results. Yes, publications were produced as a result of work using our cluster."*
>
> *"So far, the cluster has facilitated long 'production runs of 1-2 weeks; such runs are essentially impossible with computers at [larger user facilities] where strict time limits on jobs are enforced and longer jobs are given low priority. At this stage, we are preparing one publication based on calculations with our cluster. Several more are expected by the end of the year."*

[3] http://www.supercomputingonline.com/print.php?sid=4967

Q2. Has this program changed the way you do computing? (I.e. were you using something other than a cluster before? Has this helped you to move towards parallel computing?)

> *"We were using a small cluster managed by myself before. It was much less reliable, and was really just a series of machines, rather than a cluster. SCS has made it possible for us to parallelize almost everything we do."*
>
> *"We could undertake projects that involve modeling very large space."*
>
> *"Yes, the availability of our cluster has moved us towards parallel computing for different problems."*
>
> *"Yes. So far, the cluster has enabled long, parallel production runs, otherwise made difficult by the standard queues at [larger user facilities]. These runs have increased our productivity. In addition, we expect the cluster to give us greater flexibility to address exciting problems as they arise; a devoted cluster is also expected to be especially useful for code development, testing, and debugging."*

Q3. Give us your suggestions for improving the current SCS program or for addressing mid-range computing at the Lab.

> *"I would suggest software development support, including availability of parallelized subroutines."*
>
> *"The SCS is a great program; thus far, we are very pleased with the hardware they advised us to purchase, the speed with which they installed it, and their technical support. Keep up the good work!"*
>
> *"I have no real suggestions. The program has been fantastic."*
>
> *"SCS team does an excellent job in maintaining the cluster and supporting the users!"*

II. Customer Testimonies:

From the Chemical Sciences Division
November 17, 2004

The services SCS provides are tremendously valuable to our research group. The quality of these services is excellent. We get a good response time and prompt solutions for our needs. This allows us to redirect our efforts from cluster maintenance to scientific endeavors.

We have developed and applied a wavefunction optimization method on our new cluster. This method was used for obtaining wavefunction parameters for the computation of excitation energies of biological molecules, an activity directly related to our involvement in the Innovative and Novel Computational Impact on Theory and Experiment(INCITE) program. The availability and power of our new cluster have been crucial for the successful completion of this activity.

> *Energy and Environment Technologies Division*
> *June 28, 2004*
>
> *As group leader of the team that has used our new cluster for nearly one year now, I am writing to compliment your staff for their excellent and outstanding support of our cluster needs. They are knowledgeable, prompt, friendly and efficient in answering our requests and doing the necessary chores to keep our cluster happy and working away.*
>
> *The cluster has been of great help to our scientific research need, which could not be met in the past years with [larger user facility] accounts, and could not be met with individual computers in our group either. This has just the right scale (and scalability) to solve our computing needs.*

BENCHMARKING DATA

Part of the rationale for the SCS program is that centralized expertise and management of Linux clusters would provide better and more cost-effective support versus scientists doing the support themselves. In order to be as cost effective as possible, we took the following steps.

- Standardized components to minimize technical effort
- Used open source software
- Developed Warewulf to scale cluster support efforts.
- Leveraged relationships with the open source software community to access valuable technical expertise.
- Outsourced various tasks – Cluster rack wiring and seismic bracing are outsourced to firms with lower labor costs.
- Developed lower-cost staff using our relationships with the local community colleges to develop junior positions.
- Used competitive bid procurement to ensure we get the most for our money. The competitive bid process has saved on the average of 10 percent as compared to the original cost estimate for hardware.

The SCS program is staffed at 1.6 FTEs and comprises a team of five people from the UNIX Group who work on this project part-time, in addition to providing UNIX desktop and server systems administration. The team provides support for 10 clusters consisting of 297 nodes. This works out to a staffing level of approximately 185 nodes per person.

As a comparison, an article in the December 2003 issue of ClusterWorld magazine[4] refers to an informal survey taken at Supercomputing 2003 of large cluster sites across the U.S. It shows that in 2003, sites required about one full-time person to maintain a 100-node cluster, and two full-time people to maintain a 256-node cluster. More recent informal surveys indicate that most large sites now assign between 100-200 nodes per person. From these numbers, we can verify that we are staffing at a cost-effective level for cluster

[4] Learning the Hard Way: HPC Cluster Challenges - ClusterWorld Magazine, pg 14, Dec 2003.

support; especially since we have determined it is more effort to support several smaller clusters as compared to fewer larger ones.

SCS has also benchmarked against industry service providers and their comparable services cost significantly more. The program is currently working on understanding this comparison more fully. The conclusion, nevertheless, is that it costs much less to provide this support with internal staff.

SUMMARY

The SCS program is one of the IT Division’s most successful projects in recent years. SCS provides a strong alternative for Berkeley Lab researchers who want to conduct scientific computing on Linux clusters. Moreover, the effectiveness of the SCS program has led numerous researchers, who are outside of the program’s funding scope, to use their own project funds to pay for SCS services. The number of anticipated clusters supported by the SCS team is expected to grow by almost 40 percent in 2005. More growth means more support for scientific discovery - the core of Berkeley Lab’s mission.

SCS
Scientific Cluster Support

Appendix

SCS Program Clusters

Division	Principal Investigator	Project Description	Number of Compute Processors
Chemical Sciences	William Miller	Semi-classical Molecular Reaction Dynamics: Methodological Development and Application to Complex Systems	40 AMD Athlon
Chemical Sciences	Martin Head-Gordon	Parallel electronic structure theory	42 AMD Opteron
Chemical Sciences	William Lester	Quantum Monte Carlo for electronic structure	46 AMD Athlon
Materials Sciences	Arup Chakraborty	Signaling and Mechanical Responses Due to Biomolecular Binding	96 AMD Athlon
Material Sciences	Steve Louie Marvin Cohen	First-principles quantum-mechanical simulations	72 AMD Opteron
Physical Bioscience	Kim/Adams/ Brenner/Holbrook	Structural Genomics of a Minimal Genome Computational Structural & Functional Genomics A Structural Classification of RNA Nudix DNA Repair Enzymes from Deinococcus radiodurans	60 Intel Xeon
Environmental Energy Technologies	Gadgil/Brown	Airflow and Pollutant Transport in Buildings Regional Air Quality Modeling Combustion Modeling	56 AMD Athlon
Earth Sciences	Hoversten/Majer	Geophysical Subsurface Imaging	50 Intel Xeon
Life Sciences	Michael Eisen	Computational Analysis of cis-Regulatory Content of Animal Genomes	40 Intel Xeon
Life Sciences	Cooper/Tainer	Protein Crystallography and SAXS data Analysis for Sibyls/SBDR	20 Intel Xeon
Nuclear Sciences	I-Yang Lee	Gretina Detector - Signal deposition and event reconstruction	16 AMD Opteron

Disclaimer:

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, or The Regents of the University of California.