Parallel Scaling Characteristics of Selected NERSC User Project Codes

David Skinner, Francesca Verdier, Harsh Anand,
Jonathan Carter, Mark Durst, Richard Gerber,
NERSC User Services Group, Lawrence Berkeley National Lab, Berkeley, CA
March 2005

Abstract:

This report documents parallel scaling characteristics of NERSC user project codes between Fiscal Year 2003 and the first half of Fiscal Year 2004 (Oct 2002-March 2004). The codes analyzed cover 60% of all the CPU hours delivered during that time frame on seaborg, a 6080 CPU IBM SP and the largest parallel computer at NERSC. The scale in terms of concurrency and problem size of the workload is analyzed. Drawing on batch queue logs, performance data and feedback from researchers we detail the motivations, benefits, and challenges of implementing highly parallel scientific codes on current NERSC High Performance Computing systems. An evaluation and outlook of the NERSC workload for Allocation Year 2005 is presented.

Outline:

Introduction:

Motivation:

Scientific computation at all levels of concurrency should benefit from a good match of compute resource to the problem being solved. By surveying the parallelism used by the NERSC workload we hope to make the matching of scientific problem to compute resource as effective as possible. The concurrency at which researchers compute is also of interest to the NERSC center as the overall concurrency of the workload is part of existing metrics for the center's performance.

Methodology:

The overall approach in this work was to identify and examine thrity-nine NERSC projects which in total amount to about 60% of total cycles delivered. The projects were selected so that all major science disciplines of interest to the DOE Office of Science were included. The projects that participated in the Large Scale Reimbursement Program were all included in this study. For the selected projects, we identified the parallel codes run, their level of parallelism and the motivations for running the codes at a given concurrency.

The first step, associating a batch job with a particular code, is essentially ill posed at least in a quantitative sense. Though it is not the norm, a parallel job may contain multiple codes, contain only a test of one component of the full code (e.g., just the solver), or represent development work on a new or changing code. That said, 95% of the production workload may be easily associated with a specific code by looking at the path, script-name, and other details of the batch job stored in log files. In many cases we contacted the user who ran the jobs to determine or check which applications were run.

Knowing which code was run in a set of batch jobs is a valuable start, but from this one can not automatically determine the nature of the calculation. For instance, the problem size in the calculation is often not known from the data we are able to collect. Because of this we can not examine strong scaling performance from any of the data gathered automatically. For this reason we can not precisely determine the parallel efficiency of each code. Instead we focus on empirical trends in how research projects run their codes and document where possible if the problem size is known to be fixed across concurrency.

On seaborg the collection of performance data is streamlined through the poe+ utility which provides a performance report to the user and to the center on a per job basis. These reports contain information about the performance, memory usage, and software dependencies of the programs run in batch. Where possible we consult this body of data in conjunction with Load Leveler batch logs. A companion document[1] describes in more

[1] "Performance Monitoring of Parallel Scientific Applications" LBNL-5503

detail the technical aspects of the extent to which low impact workload performance monitoring and characterization can be done as well as how it can be implemented.

For each project and code we tried to answer the following questions:

- What does the code do?
- How does the project choose how many tasks to run on?
- How much effort did it take to bring the code to its current level of parallelism?
- What would be required to make it scale to higher concurrency?
- What concurrency is the most efficient in terms of science results per time?

In the case studies section of this work we provide where possible summarized responses to these questions. The overview in the next section provides a concise statement of the parallelism in use and the factors which influence the scale of computations on seaborg.

II) Overview

1) Parallel Scaling on Seaborg

The main parallel compute resource at NERSC is the IBM SP seaborg. Before looking at the application level parallelism it is worth laying out the boundary conditions of what is possible on this machine. Seaborg consists of 6080 computational CPUs connected via a colony switch. Parallel computing on seaborg is most often done via the Message Passing Interface (MPI). Individually, each CPU is capable of 1.5Gflops peak performance and has available 100-350MB/s of inter-node bandwidth.

The DOE Office of Science parallel scaling goal for seaborg for the time period of this survey is that half the user workload is run on $1/8^{th}$ or more of the machine. Within the IBM SP's MPI implementation there is an upper bound of 4096 MPI tasks therefore half the user workload should run on 512 processors or more.

The research projects were split into 3 scaling groups based on the concurrency at which each group's cycles are used. We identify Large projects as those using 50% or more of allocated cycles on 512 or more CPU's, Small projects as using 75% or more of allocated cycles on 256 or fewer CPUs, and Medium projects as between the latter two groups.

Scaling Group	% of cycles on N CPUs
Small	75% of cycles for N<=256
Medium	Any between Large and Small
Large	50% cycles for N>=512

Note that “Small”, “Medium” and “Large” apply only to the concurrency patterns of the projects, and not to their computational requirements or allocation size. E.g., a small-scale project may have a large computational requirement.

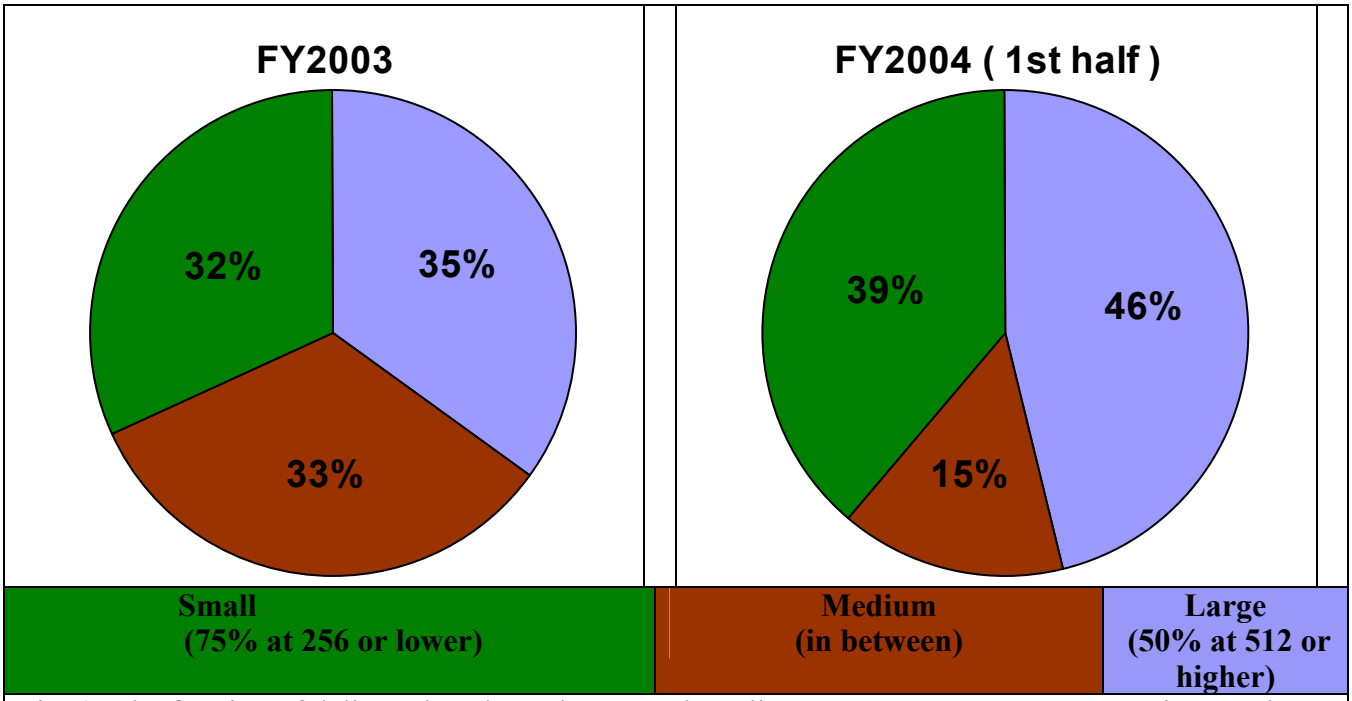


Fig 1) The fraction of delivered cycles going to each scaling category. Current FY04 estimates show a 10%increase in parallel work on 512 processors or more.

Table 1: Scaling summary of NERSC Projects in FY 2003 and the first half or FY 2004

	Number of Projects			% time used		
	Small	Medium	Large	Small	Medium	Large
FY 2003	173	55	34	32.4%	32.4%	35.2%
First ½ FY 2004	183	36	48	39.2%	15.1%	45.8%
Diff FY04 - FY03	+10	-19	+14	+6.8%	-17.4%	+10.6%

The 10.6% increase in large concurrency jobs reflects efforts at the NERSC center through queue policies and HPC consultancy to encourage and enable high concurrency work. In the following sections we break down the aggregate numbers above into the specifics of each research project.

2) Parallel Scaling of NERSC Projects

NERSC projects receive allocations of computer time through their repository or repo. In this study 39 projects were studied. The parallel scaling of these projects, over the entire span of this study is detailed in Table 2. In this table, the column headers “Allocation”, “Prob Size”, “Code Eff”, “Parallel IO” and “Queue” refer to factors which place an upper

bound on the concurrency used by the project, as explained in the text following the table.

Table 2 Parallel Scaling Characteristics of surveyed NERSC Projects

repo	science	Scale	%NERSC	Allocation	Prob Size	Code Eff.	Parallel IO	Queue	% 2048+	% 1024+	% 512+	% 256+	% 1+
mp13	qcd	L	6.1	X		X			2	54	3	13	28
gc3	fusion	L	5.6	X		X			1	28	45	11	15
gc2	accelerator	L	3.6		X			X	9	18	25	12	36
mp19	fusion	L	3.6	X					3	35	51	5	5
m41	fusion	L	3.1		X		X	X	6	42	5	17	30
m77	fusion	L	2.4			X			74	12	7	7	1
mp111	combustion	L	2.2			X			24	14	39	3	19
mp124	astrophysics	L	1.3	X			X		13	41	14	10	22
mp338	fusion	L	1.2		X				33	33	20	1	13
m152	astrophysics	L	1.1			X	X		42	2	52	0	3
mp363	astrophysics	L	1.0	X		X			0	16	35	1	48
m224	fusion	L	1.0		X	X			69	1	18	1	10
m260	chemistry	L	1.0				X		0	56	30	4	9
m328	climate	L	0.8		X			X	0	97	0	2	1
m256	chemistry	L	0.5	X	X				0	38	17	4	41
mp218	combustion	L	0.5	X		X			3	29	61	4	3
Incite2	astrophysics	L	0.4			X			0	10	47	3	41
mp100	materials	L	0.2	X					0	27	53	4	16
m234	geoscience	L	0.1	X		X			0	10	60	13	17
Incite3	enigneering	L	0.1			X			4	3	91	0	1
m106	astrophysics	M	2.4	X		X			0	1	25	4	70
mp94	fusion	M	2.3			X			0	6	26	18	50
mp107	astrophysics	M	1.8				X		8	14	23	26	30
m249	chemistry	M	0.5			X			0	20	17	0	63
m349	accelerator	M	0.4		X				0	19	15	9	58
m372	geoscience	M	0.3	X					0	0	42	54	4
mp27	qcd	S	2.3		X				0	0	0	0	100
mp7	qcd	S	2.0		X				0	4	0	10	86
mp22	astrophysics	S	1.9	X		X			2	2	7	1	89
mp9	climate	S	1.8		X	X			0	0	0	0	100
mp288	Fusion	S	1.5	X	X	X			0	0	1	0	99
mp217	fusion	S	1.4	X			X		1	6	12	19	63
gc4	materials	S	1.1	X		X			2	2	6	0	89
mp351	chemistry	S	1.0		X				0	2	0	0	98
mp208	chemistry	S	0.9		X				0	10	0	8	82
m217	combustion	S	0.8			X			0	0	0	47	53
mp221	materials	S	0.7		X				0	2	12	0	86
mp250	materials	S	0.3			X			0	8	0	62	30
Incite1	chemistry	S	0.0			X			0	0	0	0	100

Though each project has its own scientific goals, computational requirements and algorithms, certain factors emerge in many cases which place an upper bound on the concurrency used by many groups.

- **Allocation risk:** Highly parallel jobs present greater risk to small allocations, e.g. a 2048 way run lasting one hour represents 6% of the average NERSC allocation. Since jobs often need to be refined and converged through many runs, groups often choose their concurrency and in some cases problem size to minimize the risk of depleting their allocation before achieving a calculation of scientific value.

- **Fixed problem size**: The extent in terms of data size of scientific interest may preclude parallelism beyond a certain level. For a calculation with a fixed number of floating point operations the efficiency will drop beyond a certain currency as the calculation becomes spread too thinly across the processors.

- **Efficiency of parallel code**: A portion of the code may serialize, the algorithm may not scale, or some hardware resource may become bogged down. In these cases the parallel efficiency drops off as a result of how parallelism is implemented, as a result of computer architecture limitations, or both. For instance load balance, the scaling of MPI collective operations, and switch contention, fall into this category.

- **File I/O**: Time spent moving data to and from disk constitutes a limitation to scaling. While this could be included in the above parallel efficiency category, we choose to separate it as it often represents a distinct set of challenges from message passing.

- **Time to solution:** Some research needs require pursuing the quickest route to an answer. In this case time spent waiting in a queue counts against the efficiency of the computation. The concurrency may be chosen (revised up or down) in order minimize wait time based on the existing system load and queue structure. This is constraint is often not present for groups with large predictable workloads whose throughput is not impacted by queue latency.

It's important to note that while there is little that the NERSC center staff can do to influence the problem sizes of scientific interest and size of allocations, the latter factors in the above list are being addressed through efforts of NERSC center staff and NERSC Users' Group (NUG). These efforts are summarized in the outlook section and detailed in section IV.

In addition to constraints to scaling we are able to identify enabling technologies and methodologies that tend to increase the lower bound on concurrency.

- **Scalapack:** For groups whose computational problems are expressable in terms of large linear algebraic problems, this library often provides drop-in scalable parallelism (e.g., m77, mp107, gc3.)

- **Treecodes / Load Balance:** Graph theoretic bases and unstructured and or adapative grids provide a scalable approach to treating large problems. Graph partitioning can provide memory efficient and adaptable representations of data and implicit load balancing thereby removing two constraints to parallel scaling. For some calculations expressing otherwise dense problems in a graph theoretic framework can lead to improved time to solution (e.g., m349. incite2, mp363).

- **Code development effort:** The success of certain parallel codes is directly attributable to significant human resources devoted to code improvement. In the case of FLASH, ZEUS, and GAMESS the specific goal of efficient parallelism on a large scale has been achieved through the efforts of people dedicated to code development. (e.g. incite2, m256, mp363, m152)

3) Outlook for FY04-05

For the first half of FY04 45% of the production workload was run using 512 or more tasks. One of the goals of this study is to estimate the outlook for the remainder of FY04 and into FY05. Factors that will influence the final FY04 percentage derive from both the science needs of NERSC customers and from the actions and policies of the NERSC center itself.

- The number of projects with small allocations
- Shifts in the problem sizes NERSC projects choose to examine.
- The parallel scaling reached by INCITE projects.
- Resolution of outstanding MPI and I/O performance issues

In order to work toward achieving higher levels of concurrency in its workload NERSC has

- Shaped the batch queue structure to remove obstacles to scheduling large jobs (March 2003)
- Reduced the charging rate for large jobs (Jan 2004) as a way of promoting higher levels of parallelism. This can be seen as an extension of the FY03 Large Scale Reimbursement program
- Worked with IBM and seaborg users on improving MPI and parallel I/O issues. Many issues involving MPI performance have been addressed over the lifetime of the machine. Remaining issues include variability in runtime, MPI scaling, and GPFS performance.

The above activities are ongoing and will shift as needed to meet the needs and goals of the NERSC center.

Looking ahead to FY05 it is useful to keep in mind the following practicalities:

- Small allocations are at odds with highly parallel work. Accomplishing large scale calculations is more difficult for a large number of small projects than it is for a smaller number of large projects.
- Incremental allocations prevent research groups from productively budgeting their time for large calculations. NERSC and its users would benefit from an allocations process which allocates most all of the time at the start of the year.
- The limit to 4096 MPI tasks maybe extended upward in FY05 versions of IBM's MPI implementation (PE4.1.x). It remains to be seen how well the MPI library performs in the regime beyond 4096 tasks. If in a subsequent release the MPI library formally allows but does not practically support more than 4096 tasks we would recommend continuing with the 512 task definition for high concurrency work.

4) Summary

In summary, meeting the current DOE Office of Science metric will require adjustments to the allocations process and continued work on the part of NERSC center staff to accommodate a high concurrency workload. It may be worth considering metrics, other than concurrency, which could quantify the value of the center to the DOE science community.

III) NERSC Center Scaling Activities

1) NERSC Services to Science Projects

NERSC provides its users with access to a highly available parallel computing platform as well as access to experts in parallel computing, networking, and scientific computing.

NERSC tracks questions asked by users through a trouble ticket system. The questions received during the time period of this survey are reported below.

647	Software Questions
681	Programming Issues
2028	Accounts and Allocations
956	How to Run Parallel Jobs
226	Failures and Outages
36	Networking and Security

Not all interaction between NERSC customers and center staff is documented through the trouble ticket system. Some interactions take place through face to face meetings or extended collaborations. As such the above data is a lower bound in terms of the number of interactions.

The annual user satisfaction survey provides measurements of user satisfaction with NERSC resources and staff services. Particular work by NERSC staff to enhance system wide scalability is detailed in the following reports:

- http://www.nersc.gov/news/survey
- http://www.nersc.gov/nusers/nersc/reports/technical/seaborg_scaling

2) Large Scale Reimbursement Program (LSRP)

The FY03 Large Scale Reimbursement Program was a notable success for several projects.

- mp363 : A new highly scalable hybrid parallel model was implemented. "Without the LSRP the development runs for the code improvement likely could not have been done"
- mp111 : Scaling studies isolated code sections written for vector computers that present bottlenecks to parallelism.
- mp218 : Improved code through LSRP runs. "Over the course of the year we used LSRP for code development and improvement. Based on this year's results the code is compute and not communication bound"
- mp217 : Used LSRP runs to accommodate problem sizes requiring large memory and improved parallel I/O portion of code.
- m41 : LSRP runs were used to test code convergence on new classes of problems. "These calculations will provide the first theoretical description of four-body Coulomb systems which, when results are ready, will be of great interest to the atomic physics community in comparing recent electron scattering and photoionization experiments."
- mp94 : Testing and integration of SuperLU into nimrod.
- mp27 : LSRP runs used to determine optimal level of parallelism
- mp13 : LSRP runs used to "would not have done some of this scaling work if the cycles for development and testing had come from main allocation"
- mp106 : LSRP for parametric studies required to asses the numerical resolution and accuracy of highly parallel code

For others projects the program provided relief from smaller than usual and incremental allocations, which drive projects to conserve their allocation by running smaller scale jobs.

IV) Case Studies by Project Area:

1) Accelerator Physics

In total NERSC had 10 accelerator projects in production. They received 5 percent of the allocation. The codes used by these groups span a wide range from particle propagation to sparse direct solves.

PI Kwok Ko, NERSC Repository m349

Kwok Ko's project, "Advanced Modeling for High Energy Accelerators", works to maximize the capability of parallel electromagnetic and beam dynamics codes to solve challenging problems limiting the performance of current generation of accelerators which includes PEP-II, the Tevatron, RHIC, SNS and LCLS.

m349 had the second largest FY04 accelerator physics allocation and is characterized by medium-scale processor utilization.

Code Tau3P and Ptrack: Tau3P calculates a time-dependent solution of Maxwell's equations in cavities and open electromagnetic structures with or without transiting rigid beams. Tau3P consists of two coupled PDE's with a leap-frog scheme in time and a Discrete Surface Integral method in space, to advance the electric and magnetic fields in nonuniform, nonorthogonal, unstructured grids. It is written in C++.

About 60% of mp349's compute time was used running Tau3P-based codes. Tau3P's scaling is problem-dependent; it has been run using up to 1,024 processors. A code named "ptrack" uses Tau3P to calculate the EM field and adds particle-tracking capabilities. Its usage is included in the 50% quoted above. For the ptrack runs, Tau3P does not scale well for the mesh of interest and runs are limited to 64 and 128 tasks.

Code S3P and Omega3P: Omega3P computes eigenmodes (i.e. solutions to Maxwell's equations in the frequency domain) in 3D perfectly conducting RF cavities in complicated geometries. S3P is a frequency-domain solver for finding the scattering matrix of traveling wave structures.

These codes accounted for about 16% of m349's usage. Both codes have attained a concurrency of up to 1,024 using WSMP (Watson Sparse Matrix Package). Scaling and performance is good given the memory limitations. A 256-way S3P job running on 64 nodes (4 tasks/node) uses 618 GB of memory and achieves 360 Mflops/s/task. Omega3P and another code, BBSIM, are using a significant amount of time at concurrencies of 512 and 640.

Code (nameless): A weak-strong beam-beam code used to study beam diffusion and how particles get lost over many turns in an accelerator.

This code used about 17% of m349's time. It is embarrassingly parallel, needing only to evaluate a complex error function repeatedly. Most of the usage occurred at a concurrency of 1,024.

m349 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
First ½ FY 2004	0%	20.8%	22.0%	10.8%	46.4%
NOTE: FY2004 thru 6/16	0%	10.9%	40.9%	9.4%	38.9%

m349 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2004	1,165,00	590,000	51%	535,745

Conclusions: In last year or two, Lee and collaborators have made significant speedup (about 100X) by changing the algorithms in Omega3P/S3P. The codes could achieve a higher % of peak by choosing iterative linear solvers instead of sparse direct solvers but the overall computing time would be one or two orders of magnitude longer. For example, they had to spend 28 minutes per eigenmode in simulating a complicated accelerating structure previously. With new algorithms, they only need 5 second per mode. **Here it is obvious that scientific throughput has improved although hardware performance has decreased.**

PI Robert Ryne, NERSC Repository gc2

Robert Ryne's project, "Advanced Computing for 21st Century Accelerator Science and Technology", is establishing a comprehensive terascale simulation environment for use by the U.S. particle accelerator community. This simulation environment will enable accelerator physicists and engineers across the country to work together-using a common set of scalable, portable, interoperable software-to solve the most challenging problems in accelerator design, analysis, and optimization

gc2, a SciDAC project, has the largest accelerator physics allocation and has codes with multiple scaling characteristics.

The codes used in gc2 fall into three broad categories: (1) particle "PIC" codes used to self-consistently model charged particle interactions, (2) electromagnetic (EM) codes that solve Maxwell's equations in complex geometries, and (3) electromagnetic "PIC" codes used to explore effects in proposed wakefield plasma accelerators. There is significant

overlap between work being done in gc2 and repositories m349 (EM codes, see elsewhere in this study) and Warren Mori's repository mp113 (plasma accelerators).

Code BeamBeam3D, IMPACT and IMPACT-T:
IMPACT is used for RF linac beam dynamics modeling. It is a parallel Particle-In-Cell (PIC) code that uses map-based techniques or Lorentz-force solvers to describe external beamline elements, and that utilize a variety of field solvers (Green function-based for open and periodic systems; eigenfunction expansion for beams in conducting pipes) to model space-charge effects. The two effects are combined using split-operator methods. IMPACT-T modifies IMPACT to use time as the independent variable instead of the distance down the accelerator.

BeamBeam solves the Vlasov/Poisson equations describing the interaction of colliding beams. It is a parallel PIC code oriented toward modeling colliding beams. It uses a modified version of IMPACT's open boundary condition Poisson solver to obtain the electric fields acting on the beams without gridding the entire problem region. This is important since the beams may be widely spaced, but we may still be interested in studying long-range beam-beam effects. Furthermore, instead of using a domain decomposition approach, BeamBeam3D uses a particle decomposition approach.

For BeamBeam a couple of months were put to scale the code to the current level concurrency. To scale to a higher concurrency, we need to a) switch to another numerical algorithm which is based on neighboring communication instead of global communication. b) high speed internet for global communication like matrix transpose c) optimize parallel implementation

The communication time in BeamBeam comes predominantly from MPI_Allreduce.

IMPACT and BeamBeam runs accounted for 35% of the gc2 usage over the period. Production runs were performed using 1,024 processors. A concurrency of 64 tasks was also used because of uncertainty over how long it takes larger jobs to start.

The way to choose how many tasks to run is based on the scalability of the code with the given problem size and the return time since the submission to the queue. The small 32 CPU jobs are based on a quick return of the job. The 64 CPU jobs are based on the balance between the scalability and the return time. In principle, the job can also scale to 128 or 256 processors. In the interest of turn around time the users often choose 64 processors. The way they choose how many tasks to run is based on the scalability of the code with the given problem size and the return time since the submission to the queue.

IMPACT-T could run smaller concurrencies, but would need longer queue run times for less than 512 cpus. The users also like the turnaround in the 512 cpu queue.
.
Code Omega3P and S3P: Computes eigenmodes (i.e. solution to Maxwell's equations in the frequency domain) in 3D perfectly conducting RF cavities (PDE in space). See description under m239.

Omega3P and S3P usage made up 21% of the project's usage. See discussion under m349.

Code Tau3P and Ptrack: time-dependent solution of Maxwell's equations in cavities and open electromagnetic structures with or without transiting rigid beams (PDE in space & time). See description under m349.

Tau3P accounted for 24% of gc2's usage over the period under consideration.
See discussion under m349.

Code OSIRIS: OSIRIS is a fully explicit, relativistic particle-in-cell electromagnetic code. Particles are used to model the fields and are "pushed" forward to calculate the EM field.

Scaling: Osirus runs contributed to 21% of gc2's usage and runs with 85% parallel efficiency on 2,048 tasks. Most communication is nearest neighbor and done in large messages. The problem sizes currently of interest work well to 1,024 tasks. Development and convergence runs sometimes are done 512 way prior to running high concurrency production runs. The group foresees large problem sizes in the future and thinks the code is ready to accommodate them.

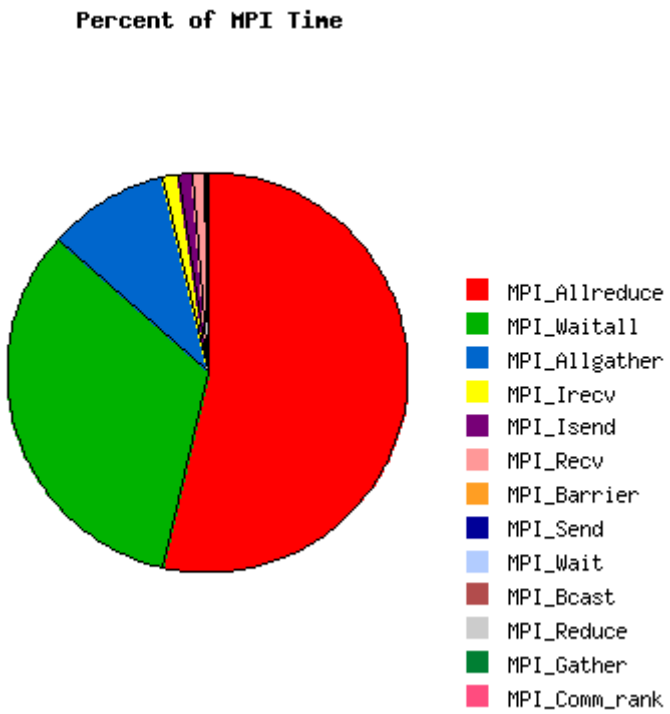
See m349 discussions regarding the EM codes. NERSC has had extensive consultations with this group. We uncovered a problem with the way IBM instrinsics were spawning threads. This lead to a number of reports, see LBNL-repcor_num (ragerber) for references.

gc2 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	8.9%	20.7%	22.2%	14.0%	34.2%
First ½ FY 2004	2.1%	14.3%	39.0%	3.0%	41.6%
Diff FY04-FY03	-76%	-31%	+76%	-79%	+22%

gc2 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	1,000,000	1,586,000	159%	1,802,280	320,000
FY 2004	2,330,000	1,395,000	60%	1,203,228	-



Distribution of time spent in MPI for some of the gc2 workload. Not all runs are included in this analysis. Most runs contributing to this analysis were BeamBeam and parallel FFT tests. The message sizes are typically small one order of 1KB.

Conclusions:

Allocating computer time in small increments makes this group less productive as it prevents good planning about the scale and number of calculations that can be accommodated within the allocation.

2) Astrophysics:

In FY 2004 NERSC had 20 astrophysics projects in production. They received 17percent of the allocation.

PI Julian Borrill, NERSC Repository mp107

Julian Borrill's project, "Cosmic Microwave Background Data Analysis", is concerned with the analysis of data from observations of the temperature and polarization of the Cosmic Microwave Background radiation. This analysis involves:
(i) making maps of the CMB sky from the time ordered observation data,
(ii) calculating the angular power spectrum of the anisotropies in the maps,
(iii) constraining the fundamental parameters of cosmology from the power spectrum.

mp107 had the fifth largest FY04 astrophysics allocation and is characterized by large-scale processor utilization.

All the codes in the repository scale extremely well. The size and maturity of the experimental data influences the final concurrency as early data is first mapped and analyzed on a courser grained level. Variations in workload are expected from year to year. This project used the Large Job Reimbursement program to develop and test a new algorithm called 'gang-parallel' MADCAP enabling more efficient data decomposition to be used.

This project has developed codes that scale extremely well. In the future even larger datasets will require some algorithm restructuring, hopefully allowing them to scale to more than 5000 processors. The filesystem demands of this work will also increase with larger datasets.

Code MADCAP: maximum likelihood power spectrum estimation by Newton-Raphson iteration, with multiple dense linear systems solved exactly by Cholesky decompostition, matrix inversion and multiplication..

This code accounts for roughly 40% of the time used by this repository. MADCAP scales very well, the largest run using 3584 processors along with many production runs with 1024 and 2048 processors.

Code MADmap: maximum likelihood mapmaking using FFT-based circulant matrix multiplication within a preconditioned conjugate gradient solver.

Accounting for about 25% of the time used, this code scales very well – up to 2048 processors have been utilized so far.

Code Planck: A suite of codes for noise analysis of Planck experiment data.

Approximately 10% of the time used goes to the Planck suite of codes. These codes can scale very well, up to 2048 processors, depending on the size of the dataset to be analyzed.

mp107 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	10.5%	7.6%	21.7%	28.2%	32.1%
First ½ FY 2004	1.3%	30.5%	24.6%	19.8%	23.9%
Diff FY04-FY03	-9.2%	+22.9%	+2.9%	-8.4%	-8.2%

mp107 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	800,000	719,800	90%	808,842	82,200
FY 2004	1,165,000	930,000	80%	1,070,925	-

PI Anthony Mezzacappa, NERSC Repository mp124

Anthony Mezzacappa 's project, "Shedding New Light on Exploding Stars: TeraScale Simulations of Neutrino-Driven Supernovae and Their Nucleosynthesis", solves the nuclear many-body problem in order. to understand the ground-state and excitation properties of nuclei.

mp124, a SciDAC project, had the third largest FY 2004 astrophysics allocation and is characterized by large-scale processor utilization.

Code V2D: solves the integro-partial differential equations of either Newtonian or relativistic radiation-hydrodynamics as simulations of the convective epoch of post-bounce supernovae.

Scaling: Accounts for 23% of the group's cycles and are mostly delivered to 1024 and 2048 way runs.

Code FLASH:

Scaling: This well known multigrid code for nuclear astrophysics accounts for 13% of cycles. It is most often run on 512-1760 tasks.

Code ZEUS-MP:

Zeus is a time explicit MHD code.

Scaling: 45% of the group cycles are delivered for 512-4096 way jobs, with the largest number of cycles being delivered for 1024 way jobs. The I/O in this code bogs down for more than 1024 tasks. The current I/O strategy uses a single file per task which creates performance problems at higher concurrency. In order to use their allocation cautiously ZEUS is also often run on 96 tasks. The group would like to attempt much larger problems given better I/O performance and allocation.

mp124 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	26.4%	2.4%	17.4%	20.3%	33.6%
First ½ FY 2004	6.0%	59.7%	12.8%	5.7%	15.9%
Diff FY04-FY03	-20.4%	+57.3%	-4.6%	-14.6%	-17.7%

mp124 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2003	608,000	364,000	60%	309,917
FY 2004	3,700,000	1,225,000	33%	1,114,892

PI Peter Nugent, NERSC Repository mp22

Peter Nugent's project, "Spectrum Synthesis of Supernovae", measures the fundamental parameters of cosmology that shape our current understanding of particle physics through the observation of very distant Type Ia Supernovae. The Supernova Cosmology Project has discovered approximately 120 spectroscopically confirmed high-redshift supernovae over the past four years.

mp22, an FY 2003 Big Splash project, had the fourth largest astrophysics allocation in FY 2004 and is characterized by small-scale processor utilization.

Code PHOENIX: models astrophysical plasmas in 1-dimension under a variety of conditions, including differential expansion at relativistic velocities found in supernovae. The code solves the fully relativistic radiative transport equation for a variety of spatial boundary conditions in both spherical and plane-parallel geometries for both continuum and line radiation, simultaneously and self-consistently using an operator splitting technique.

Scaling: 88% of delivered cycles go to this code which is typically run 64 - 256 way. This code is highly synchronizing, spending most of its communication time in MPI

collectives. The memory access patterns in PHOENIX are disordered (not of regular small stride) . For this reasons memory bandwidth is a barrier to performance.

Code SYNPOL: is a 3-dimensional radiative transfer code used to study the spectropolarimetry of supernovae. It is based on a Monte Carlo treatment of line formation via the Sobolev approximation and includes electron scattering.

Scaling: 11% of cycles go to this code which is most often run 512 way. Some runs at 1024 and 2048 tasks have been done. SYNPOL basically runs at 10% efficiency due entirely from moving around a large chunk of memory as the photons scatter randomly throughout the atmosphere. It scales very nicely, but due to the memory size of the problem is stuck at this low efficiency.

The group mentions that a factor of 10 increase in allocations would allow them the flexibility to try large problems and concurrencies.

mp22 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	3.2%	0.6%	15.6%	80.6%
First ½ FY 2004	0%	4.5%	0%	0.9%	94.6%
Diff FY04-FY03	-	+1.3%	-0.6%	-14.7%	+14.0%

mp22 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2003	720,000	646,000	90%	799,176
FY 2004	1,160,000	980,000	84%	840,217

PI Tomasz Plewa, NERSC Repository incite2

Tomasz Plewa ’s project, “Thermonuclear Supernovae: Stellar Explosions in Three Dimensions”, studies the long-standing problem of thermonuclear flashes on the surfaces and interiors of compact stars. These are extraordinarily complex phenomena to simulate. The principal challenge is the enormous range of dynamically relevant scales as well as

the breadth of physical phenomena involved, the latter including accretion, shear flow and Rayleigh-Taylor instabilities on the stellar surfaces, ignition of nuclear burning under conditions leading to convection, deflagration and/or detonation waves, stellar envelope expansion, and the possible creation of a common envelope binary star system.

This repo is an INCITE project; it had the largest FY 2004 astrophysics allocation and is characterized by large-scale processor utilization.

Code FLASH: The FLASH code is a multi-purpose community code that can solve a very of large variety of problems in physics and astrophysics. This particular simulation treats explosive evolution of white dwarf stars**.** The principal part of the FLASH code consists of a suite of high-resolution (PPM, TVD) solvers for simulations of partial differential equations of classic and relativistic hydrodynamics and magnetohydrodynamics. These solvers are augmented by a number of local equations of state modules and ODE-based nuclear chemistry solvers. The code supplies native multigrid and multipole solvers for computations of PDE of elliptic kinds. These solvers are used to simulate low-speed flows, self-gravitating flows and force-free equilibrium magnetic field configurations.

Scaling: Flash is the only code being run under this project. The code implements continuous remeshing of an adaptive model grid. The parallel scaling is dominated by the performance of the remeshing which is a global operation. Much of the implmentation is aimed at decreasing the amount of floating point work require to solve the problem, e.g., using an adapative grid. As a result the code scales to very large problem sizes and is efficient at producing useful results even though the floating point performance is lower than a code with more floating point work.

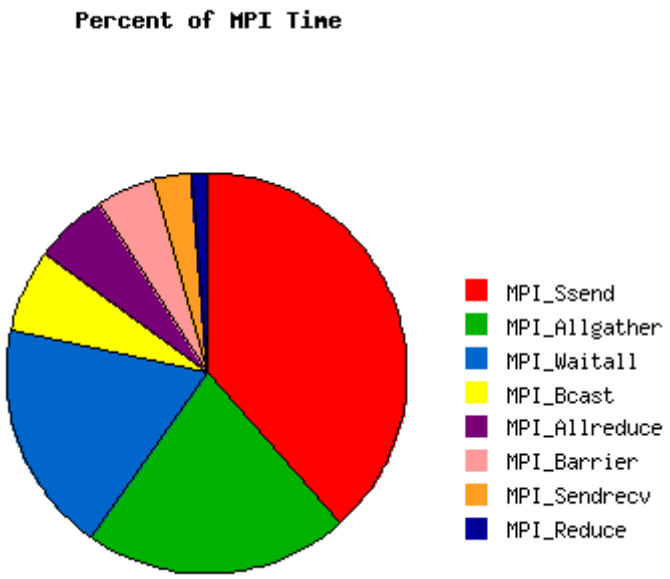
Implementing the code on seaborg was done at 16-256 tasks, it is now regularly running on 1024 tasks. Insofar as the remeshing performance can be improved the code could be scaled up further. The amount of code work required to do this is not clear, though quite a lot of effort in coding has already been done.

incite2 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
First ½ FY 2004	0%	23.8%	39.4%	2.6%	34.2%

incite2 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2004	5,400,000	2,655,000	49%	3,234,044



Distribution of time spent in MPI for some of the incite2 workload. Not all runs are included in this analysis. At 128 way the code sometimes spend 16% of total wallclokc time in MPI routines.

Conclusions: Significant effort has been applied to make this code scale successful. The incite2 project is satisfied with the scale at which the production runs are being done. Significant coding effort has lead to a code that scales to 1024 tasks. While the group has identified sections of the code that could be improved in terms of parallel scaling, they are not currently planning to devote resources to more code work.

PI Joel Primack, NERSC Repository mp363

Joel Primack's project, "Galaxy Formation Simulations", provides the basis for theoretical modeling of galaxy formation and evolution for comparison with diverse observations of nearby and distant galaxies.

mp363 had the sixth largest FY 2004 astrophysics allocation and is characterized by large-scale processor utilization.

Code ART: integrates trajectories of the collisionless particles by employing standard particle-mesh techniques to compute particle accelerations and advance their coordinates and velocities in time. The force evaluation is done on an adaptive grid.

Scaling: 43% 1024 way some 64 way

Code GADGET: force calculations between star, halo, and gas particles are carried out via the Barnes/Hut tree method with a spline softening and the gas hydrodynamics algorithm is a conservative entropy formalism of the standard Smoothed Particle Hydrodynamics (SPH).

Scaling: 52% 128 –512 way

Availability of cycles in the repo influences choices about the scale of runs. With a small amount of time available the group chooses to run many smaller runs rather than risk the whole allocation on a small number of runs.

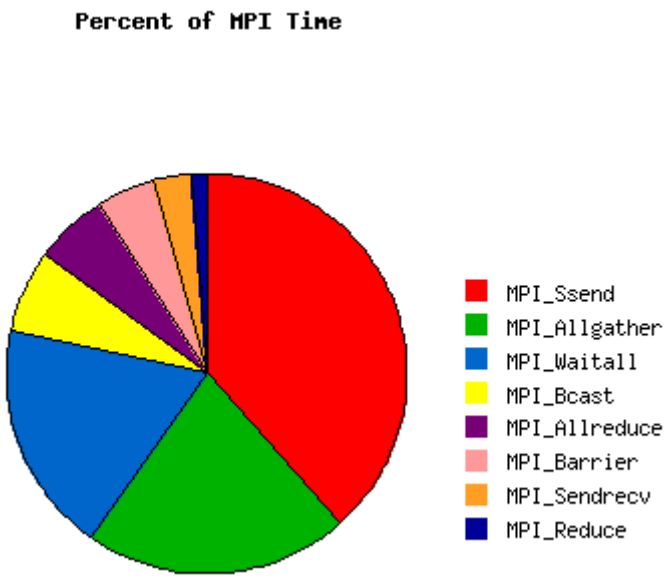
The large scale reimbursement program allowed to group to implement and optimize a hybrid MPI/OMP parallel scheme that scales their cosmological AMR code up to 1024 CPUs. They found that threading across four cpus per MPI task provided the best parallel efficiency. Without the reimbursement program the development runs for this code improvement likely could not have been done. Achieving good parallel efficiency beyond 1024 would require further code development.

mp363 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	0.0%	35.9%	2.3%	61.8%
First ½ FY 2004	0%	40.8%	33.9%	0%	25.3%
Diff FY04-FY03	-	+40.8%	-2.0%	-2.3%	-36.5%

mp363 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	196,000	401,000	205%	419,056	-
FY 2004	700,000	490,000	70%	402,377	100,000



Distribution of time spent in MPI for some of the mp3 workload. Not all runs are included in this analysis.

Conclusions:

The large scale reimbursement program helped this project scale their code up using a Barnes-Hutt tree code approach,

PI Edward Seidel, NERSC Repository m152

Edward Seidel 's project, "Black Hole Merger Simulations", studies pre-coalescence orbits of binary black hole pairs and extracts accurate gravitational wave form signals from the inspiralling binaries and collapsing rotating neutron stars.

m152, an FY 2003 Big Splash project, is characterized by large-scale processor utilization. Big Splash was a precursor to the DOE INCITE program that provided large allocations for specific projects.

Code Cactus: the Cactus framework provides all of the basic services needed for large massively parallel evolutions such as, a parallelization/global-array abstractions, uniform elliptic solver interface, parallel I/O, integrated distributed visualization/analysis/steering capability. The Cactus code directly evolves Einstein's equations as a system of coupled PDEs using hybrid ADM/Hyperbolic methods. The code also uses an elliptic solver for some aspects of the calculation.

Scaling: All the usage in this repo goes to cactus. The code shows good scalability preserving 82% parallel efficiency on 1024 tasks relative to the 16 task performance.

The code had parallel I/O scaling problems which have been largely addressed, through consultations with NERSC staff, by decreasing the number of writer tasks at high concurrency. The large dense grid in the calculation is one driver of the parallel scale. Running on a larger number of nodes provides more memory allowing for larger grids. The group has made heavy use of the 32 GB nodes on seaborg for this reason.

m152 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	46.9%	2.7%	47.1%	0.4%	2.9%
First ½ FY 2004	9.7%	0.1%	90.1%	0%	0.1%
Diff FY04-FY03	-37.2%	-2.6%	+43.0%	-0.4%	-2.8%

m152 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	400,000	442,000	111%	644,352	205,200
FY 2004	1,750,000	225,000	13%	178,964	-

PI Stan Woosley, NERSC Repository m106

The UCSC/UA/LANL/LLNL Supernova Science Center has assembled a collaboration in computational astrophysics devoted both to solving the ``supernova problem'' and to providing sufficient validation checks and community participation that the results will be accepted. They investigate both core-collapse and thermonuclear supernovae and provide detailed nuclear and spectroscopic diagnostics of all their models.

m106, a SciDAC project, had the second largest FY 2004 astrophysics allocation and is characterized by small-scale processor utilization in FY 2004 and medium-scale processor utilization in FY 2003.

Code DYNAMO: solves the nonlinear inelastic MHD equations: equations for conservation of mass and magnetic flux, a heat (entropy) equation, an equation of state, and the magnetic induction equation.

Scaling: 5% of the project cycles went to DYNAMO, which is typically run using 32 processors.

Code geo56u: The code solves the three-dimensional inelastic hydrodynamics equations for the three components of the fluid flow velocity and for three thermodynamic quantities as perturbations to a time-independent reference state.

Scaling: 23% of the project cycles went to geo56u, which is typically run using 128 processors. This code has not been tested whether to see if can be scaled up an order of magnitude to 1024 processors. Due to the large amount of communication necessary with this spectral code (global communication each timestep) it is not likely that it would scale well to 1024 processors. Typically, they run with no more than 8 nodes (128 processsors), because that is sufficient for their stage of development.

Code FLASH: an adaptive mesh compressible hydrodynamics code that is used for studies of mixing in Type II supernova.

Scaling: 5% of the project cycles went to FLASH, which is typically run using 128 processors. For similar systems this code has scaled efficiently to 6420 processors oon the ASCI Red system, for which it was awarded the Gordon Bell prize in 2000.

Code SNe: a modified version of the LBL/CCSE combustion code that solves the low-Mach number formulation of the Euler equations for combustion problems. It is being applied to flame instabilities in Type Ia supernovae in both 2 and 3-d. The low-Mach number formulation allows us to take much larger timesteps than a compressible code, and is the only reason that we are able to study these problems (Landau-Darrieus instability and reactive Rayleigh-Taylor).

Scaling: 50% of the projct cycles went to SNe, which is typically run using 512 processors, as well as a small amount of 1024 processor usage. The current version of this code can not run on much more than 128 processors because there is not enough memory per processor. That is why they are developing a newer version (currently running on their Beowulf cluster) that has finer parallelization and so requires less memory per processor. The group anticipates no problems running on 1024 or more processors on Seaborg with this version. Code performance is dominated by elliptic solves, which require global communication frequently across all the processors. Nevertheless, the number of timesteps required by this code is ~1/100000th of the number needed by a fully compressible code (which is at the moment, their only alternative for 3-d simulations).

The scaling of the primary code (SNe) is currently driven by allocation constraints. The problem size and scale of runs is adjusted to get converged results within the allocated time. In recent work the problem size has increased (512x512x1024 grid at finest resolution) and the concurrency has increased to 512 processors.

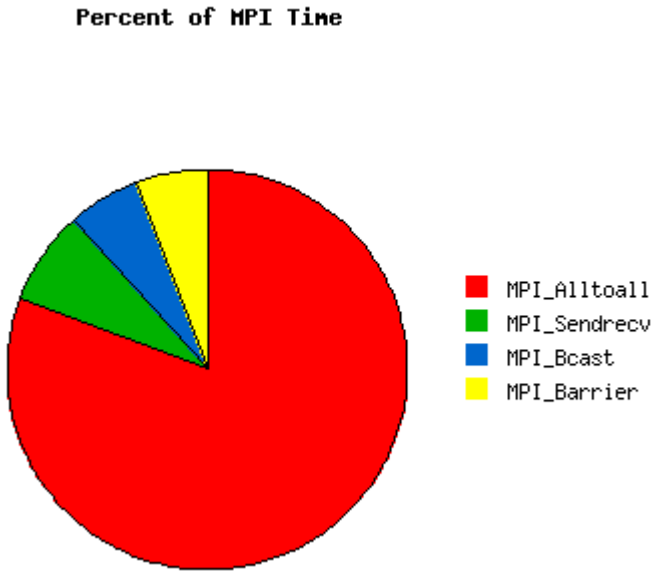
m106 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs

FY 2003	0%	1.7%	42.2%	4.1%	52.0%
First ½ FY 2004	0%	0%	0%	2.6%	97.4%
Diff FY04-FY03	-	-1.7%	-42.2%	-1.5%	+45.4%

m106 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	1,800,000	959,200	53%	963,652	15,800
FY 2004	1,750,000	1,500,000	86%	1,716,534	-



Distribution of time spent in MPI for some of the mp106 workload. Not all runs are included in this analysis. The dominance of MPI_Alltoall is due to the fact that at every timestep the code transposes the data between spectral and physical space.

3) Chemistry:

In FY 2004 NERSC had 31 chemistry projects in production. Chemistry received 8 percent of the allocation.

PI Perla Balbuena, NERSC Repository m249

Perla Balbuena's project, "Computational chemistry search of efficient catalysts", predicts physical and chemical properties of electrodes consisting of bimetallic and trimetallic nanoparticles embedded in a polymer system and dispersed on a carbon support. They determine activity and selectivity of oxygen electroreduction as well as stability and sintering of nanocatalysts embedded in a polymer membrane deposited on carbon surfaces.

m249 was characterized by small-scale processor utilization in FY 2004 and by large-scale processor utilization in FY 2003. The reduction in concurrency was because the DL_POLY code was not found to scale well.

Code DL_POLY: This is a particle mesh ewald molecular dynamics code from CCLRC.

Scaling: DL_POLY accounts for 95% of m249's usage. This code runs on a wide range of concurrencies, for this project runs spanned 16-1600 tasks with roughly the same number of hours being delivered at low and high concurrencies.

Gaussian: The well known ab initio quantum chemistry code.

Scaling: 0.5% of hours delivered went to Gaussian runs. These typically ran on 16-64 tasks.

m249 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	35.4%	17.1%	0%	47.4%
First ½ FY 2004	0%	0%	17.7%	0.0%	82.3%
Diff FY04-FY03	-	-35.4%	+0.6%	-	+34.9%

The driving factor in the decrease in concurrency between FY03 and FY04 is application efficiency. For this repo's calculations the parallel efficiency of DL_POLY decreases 60% between 1024 and 1600 MPI tasks. DL_POLY uses a significant amount of replicated data and has well documented scaling bottlenecks. For 30K atom systems it is not unusual for DL_POLY performance to show a decrease in efficency in the range of 512 tasks.

m249 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	20,000	85,400	427%	180,206	42,800
FY 2004	460,000	149,000	32%	156,890	-

This group took advantage of the FY 2003 Large Jobs Reimbursement Program to try higher scale work. For the current systems of interest and with current software it is more efficient to run on a small number of processors.

Conclusions:

In order to improve the parallel scaling of m249's work, it is likely that either significant changes must be made to DL_POLY to improve it's parallel efficiency or a suitable replacement code with superior scaling properties should be found. The authors of DL_POLY have expressed an interest in decreasing the amount of replicated data in the code, but it is unclear on what time frame this might happen. The Balbuena group has begun investigating the highly scalable CPMD code for its quantum MD workload.

PI Maciej Gutowski, NERSC Repository m260

Maciej Gutowski's project, "Molecular Energetics of Clustered Damage Sites in DNA", uses computational chemistry models to obtain molecular insight and characterization of the structure, energetics, and spectroscopy of singly and multiply damaged (clustered) DNA sites.

m260 had the ninth largest FY 2004 chemistry allocation and is characterized by large-scale processor utilization.

Code NWChem: involves many different mathematical models including nonlinear ODEs and PDEs. The underlying model is the solution of the Schroedinger, Dirac or Hamilton equation. This takes the form of perturbative approaches and many body problems for correlated algorithms. NWChem provides finite basis set (Gaussian and plane wave) solutions to Schrodinger equation for bound electronic states. The Schrodinger equation is solved in either density functional theory scheme or in an ab initio formulation.

Scaling: NWChem can scale to thousands of tasks for sufficiently large molecular systems. For the calculations of interest to this group NWChem does a large amount of disk I/O from each task.

m260 Project Scaling:

Scaling	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	72.0%	14.4%	6.0%	7.6%
First ½ FY 2004	0%	16.0%	69.5%	0.1%	14.3%
Diff FY04-FY03	-	-56.0%	+55.1%	-5.9%	-+6.7%

In FY2003 an investigation of NWChem scaling was conducted. Most of the runs with more than 1024 tasks represent these studies.

m260 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	12,000	249,200	2,076%	475,876	60,800
FY 2004	580,000	185,500	32%	184,716	-

Conclusions:
As a result of the Large Job Reimbursement program it was found that scratchfile I/O demands present a significant impediment to the parallel scaling of NWChem on seaborg. NWChem requires per task scratch files that work well with local disk but do not perform as well on GPFS. Seaborg achieves about one tenth the performance of a PNNL Intel cluster with local disk. As a result seaborg did not prove useful for this particular workload.

PI Robert Harrison, NERSC Repository m256

Robert Harrison's project, "Chemical Discovery through Advanced Computing", develops highly correlated methods for the prediction of accurate energetics and subsequently dynamics for complex chemical processes including multiple potential surfaces.

m256, a SciDAC project, is characterized by large-scale processor utilization.

Code GAMESS: The well known quantum chemistry code.

Scaling: About 75% of cycles delivered since FY03 went to this quantum chemistry code used mostly for 1024 task jobs. The systems of interest are small hydrocarbons.

Code GAUSSIAN: The well known ab intio electronic structure code. This code is used primarily to determine accurate energies for potential surfaces.

Scaling: 15% of the project cycles went to Gaussian runs. For most systems Gaussian 98 does not scale well beyond 32 tasks. For m256 all Gaussian usage took place on 64 or fewer tasks with 16 tasks being the predominant parallel usage.

Code QMAGIC: A fermionic quantum monte carlo code.

Scaling: QMAGIC accounts for 3% of utilization mainly in the range of 128-256 tasks. See mp208 for more information.

m256 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	0%	25.7%	0.9%	73.4%
First ½ FY 2004	0.1%	64.6%	10.7%	6.8%	17.9%
Diff FY04-FY03	+0.1%	+64.6%	-15.0%	+5.9%	-55.5%

The driving factor in the changes in concurrency above is a shift from running Gaussian at low concurrency to running GAMESS at higher concurrency. A great deal of coding effort and software engineering was done on seaborg by m256 to make GAMESS run well at high concurrency. The parallelization scheme was adapted to better suite SMP clusters. This development work was presented at SuperComputing 2004 by Ryan Olson.

m256 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	200,000	129,000	65%	126,063	-
FY 2004	580,000	143,500	25%	180,227	6,500

Conclusions:

The group scaled up their main code making use of the Large Jobs Reimbursement program. In this case a directed effort aimed at software change was required rather than a quick fix or MPI optimization. They have requested more time to continue this work.

PI William Lester, NERSC Repository incite1

William Lester's project, "Quantum Monte Carlo study of photoprotection via carotenoids in photosynthetic centers", investigates the photoprotecting role of carotenoids in light harvesting complexes of bacteria and plants.

incite1, an INCITE project, had the largest FY 2004 chemistry allocation and is characterized by medium-scale processor utilization.

Code Zori: uses embedding methods for the approximate treatment of solvent and fragments, mixed VMC-DMC methods for treating systems of large size, LSQMC (Linear-Scaling QMC) capabilities, and effective core potentials.

Scaling: This code represents all the usage for this project.

incite1 Project Scaling:

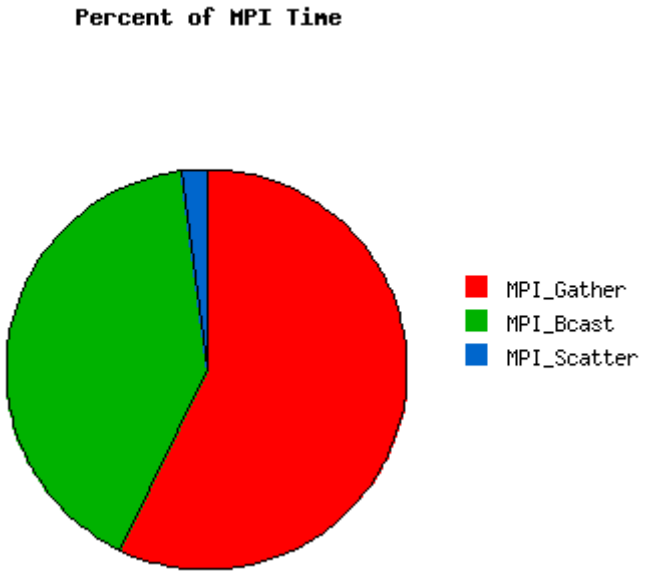
	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
First ½ FY 2004	0%	0%	0%	49.9%	50.1%

This code is new to the SP being ported at the beginning of FY04. It has undergone major revision and the development cycles used thus far have been at low concurrency to conserve the allocation for production runs. NERSC consultants provided assistance in the porting, supplying libraries and compiler help. More significantly the issue of load balance at high concurrency has been addressed with the authors through code instrumentation and testing. Load balance is not a significant issue on the workstation platforms where zori has previously run. For hundreds or thousands of tasks, load balance become important to the code's performance.

The level of NERSC staff time put into code improvements for this project is unusually relative to most other projects. Making the code scale well was given high priority under the INCITE program and as demonstrated above the project made a sharp transition from small concurrency runs to a regular workload of 512 way jobs. The primary impediment to scaling when the code was first ported to the SP was dynamical load imbalance.

incite1 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2004	1,000,000	1,000,000	100%	1,387,169



Distribution of time spent in MPI for some of the incite1 workload. Not all runs are included in this analysis.

Conclusions:

This code was new to the IBM SP and after a few months of effort its scalability was vastly increased. As of this writing the main bottleneck to scaling to higher concurrencies (e.g. 6000 task) is file I/O.

PI William Lester, NERSC Repository mp208

William Lester's project, "Quantum Monte Carlo for Electronic Structure of Molecules", is directed at a range of research problems where the high accuracy of the quantum Monte Carlo method is important for the resolution of:

- Energetics of the A and B States of Cyclopentadienyl Radical
- Mechanism of Organosilane Formation
- Energetics of Combustion Molecules and Radicals
- Peptide proton affinities
- Applications of the Fermion Monte Carlo (FMC) Method to Large Molecular Systems

mp208 had the fourth largest FY 2004 chemistry allocation and is characterized by medium-scale processor utilization.

Code QMagiC: solves the full many-body Schrodinger equation (PDE) in imaginary time using a Monte Carlo method to obtain extremely accurate electronic state energies for molecular systems. The code makes use of a number of optimization methods via the

GNU Scientific Library: Levenberg-Marquardt, Fletcher-Reeves, Polak-Ribiere, Broyden-Fletcher-Goldfarb-Shanno, and steepest descents.

Scaling: Near linear scaling due to minimal communication between nearly independent tasks.

mp208 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	0%	0%	0%	100%
First ½ FY 2004	0%	21.1%	0%	16.4%	62.5%
Diff FY04-FY03	-	+21.1%	-	+16.4%	-37.5%

QMAGIC has been running more at 256 and 1024 tasks due to changes in the systems under study. As the extent of these systems expands (or contracts) the scale of the workload will shift as well.

mp208 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	400,000	326,200	82%	331,466	-
FY 2004	1,160,000	240,000	21%	350,561	60,000

This project took advantage of the FY 2004 Reimbursement Program to scale their codes.

Conclusions:

The concurrency at which QMAGIC runs is principally determined by the size of the system under study. It does not easily become communication bound as the total communication volume increases only linearly with concurrency.

PI Manos Mavrikakis, NERSC Repository mp351

Manos Mavrikakis' project, "First-Principles Catalyst Design for Environmentally Benign Energy Production", is developing a first-principles approach to the design of novel catalytic materials. A fully parallelized and efficient planewave total energy Density Functional Theory (DFT)-based code is routinely used for modeling elementary reaction steps on transition metal surfaces in order to design new catalysts based on first-principles calculations. In particular, the thermochemistry of reactions, atomic and

molecular diffusion barriers, and activation energy barriers for chemical reactions are calculated with remarkable accuracy.

mp351 had the third largest FY 2004 chemistry allocation and is characterized by small-scale processor utilization.

Code DACAPO: uses a fully parallelized and efficient planewave total energy Density Functional Theory (DFT)-based code, of the Car-Parrinello type, for modeling elementary reaction steps on transition metal surfaces and alloys. The total energy of a given configuration of atoms is minimized by using a variety of alternative algorithms, depending on the potential energy surface characterizing the physics of the system treated. Conjugate gradient and BFGS methods are used for the energy minimization. Special eigenvalue solvers are implemented for taking advantage of parallel architectures.

Scaling: For systems the size studied in this project (20-30) atoms DACAPO becomes communication bound for more than 64 tasks.

mp351 Project Scaling:

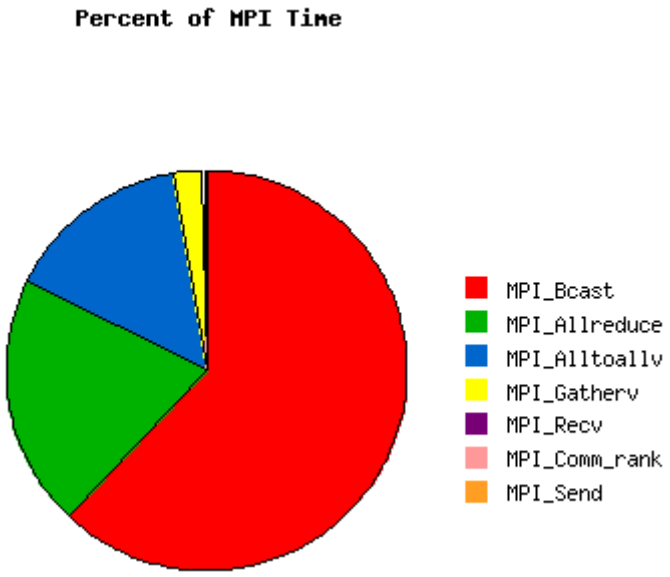
	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	5.4%	0%	0%	94.6%
First ½ FY 2004	0%	0%	0%	0%	100%
Diff FY04-FY03	-	-5.4%	-	-	+5.4%

A very important aspect of this code, which enhances its scalability, is that it incorporates 2 dimensions of parallelism: k-points and planewaves. k-point parallelization yields better performances than planewave parallelization. It is natural that the efficiency decreases with planewave parallelism, for 2 reasons: Relatively small matrices means that communications time will become important; and the subspace eigenvalue problem is an unavoidable, algorithmic bottleneck. The actual speedups will, of course, depend on the problem at hand. Speedups have been close to linear up to about 100 CPUs for typical production runs. The largest jobs the project has run (1024 CPUs) achieved 31% of the theoretical peak MFLOP/s performance, which is very good. However, they do not anticipate using many of these very big jobs, primarily because of the nature of the problems they are interested in solving.

Although they successfully tested large jobs, they do not believe these jobs could serve their scientific goals well. They could easily see using 128 nodes of seaborg for their activation energy barriers determination jobs, but using more nodes than that would not be efficient or necessary. In fact, the project Principal Investigator sees a significant threat to excellent quality supercomputing research by expanding in the direction of using more and more nodes per job.

mp351 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	300,000	167,600	56%	240,268	13,000
FY 2004	700,000	351,000	50%	529,344	-



Distribution of time spent in MPI for some of the mp351 workload. Not all runs are included in this analysis.

Conclusions:

For this research reasonable scaling to 1024 tasks can achieve 30% efficiency. However the same research can be better achieved in the range of 100 tasks where near linear speedup is possible.

4) Climate

In FY 2004 NERSC had 24 climate projects in production. They received 9 percent of the allocation.

PI David Randall, NERSC Repository m328

David Randall's project, "Super Parameterization for ARM", is testing a completely new approach to cloud parameterization for climate models that they believe will yield much

more reliable simulations of future climate change. They are conducting a 15-year AMIP 2 simulation with the super-parameterization.

m328 had the second largest FY 2004 climate research allocation and is characterized by large-scale processor utilization.

Code Super-CAM: uses a combination of spectral and finite-difference methods to solve a system of coupled nonlinear PDEs. This code is based on the NCAR Community Climate Model, used by many group. The standard code is not as computationally expensive as modified version used by this group. This version is a few hundred times as expensive because of a more detailed way of parameterization of clouds.

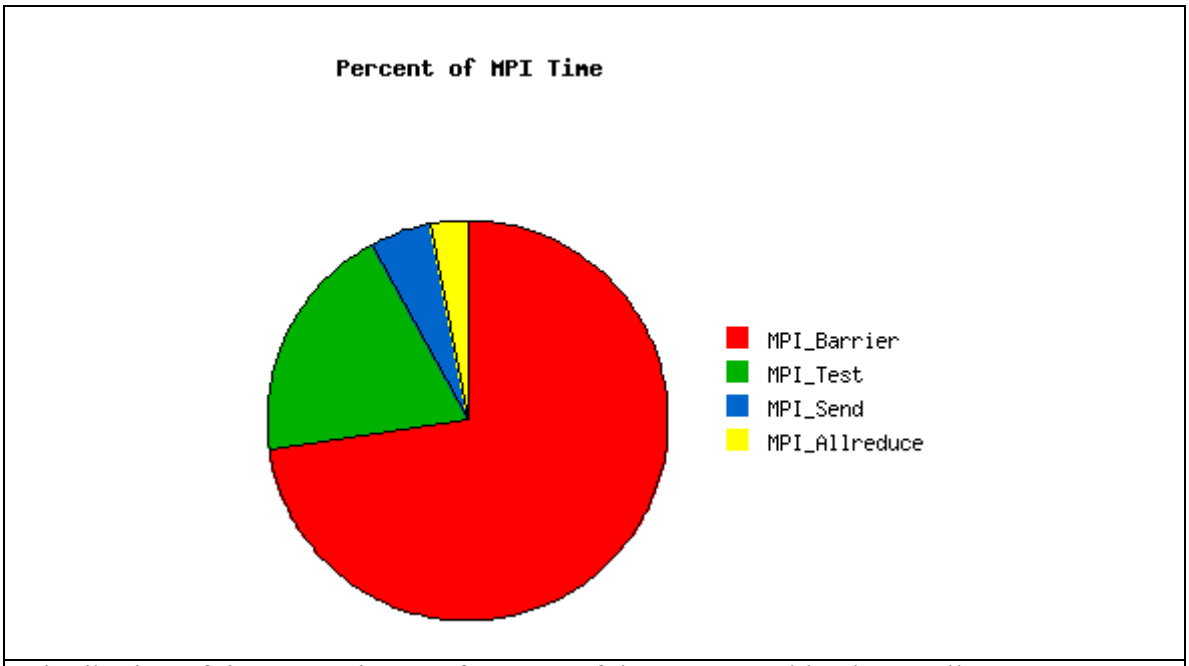
Scaling: This code accounts for nearly all usage by the group and is run 1024 way. The communication in this code is characterized by a small number of large messages. Larger problem sizes would scale better.

m328 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
First ½ FY 2004	0%	97.1%	0.4%	1.9%	0.5%

m328 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2004	1,750,000	680,000	39%	710,551



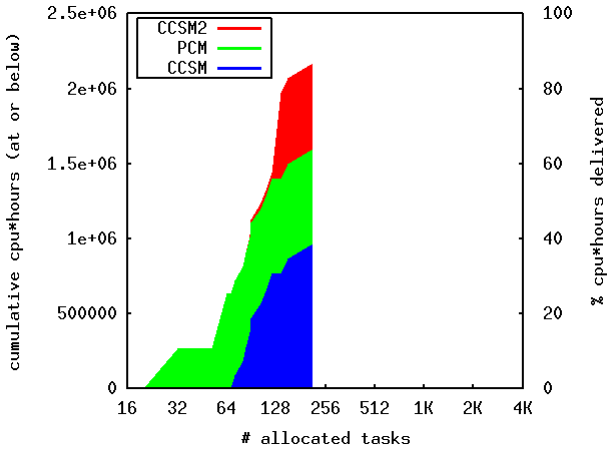
Distribution of time spent in MPI for some of the m328 workload. Not all runs are included in this analysis.

Conclusions:

Parallel efficiency is limited by problem size. This and turnaround time drive this group's choices in concurrency.

PI Warren Washington, NERSC Repository mp9

Warren Washington's project, "Climate Change Simulation with CCSM2: Moderate and High Resolution Studies", carries out a substantial portion of the climate change scenarios for the US contribution to the Intergovernmental Panel of Climate Change (IPCC) fourth assessment report. These scenarios will be run with the Community Climate System Models (CCSM) at a T85 resolution (explained below).



mp9 had the largest FY 2004 climate research allocation and was characterized by small-scale processor utilization until the second half of FY2004. The usage trends depicted to the left changed substantially in FY2004 due to improvements in the organization of their simulations which allowed scheduling greater amounts of work at a time.

Code CCSM(2): The code simulates historical or future climates of the earth by solving a set of time-dependent nonlinear partial differential equations through numerical integration. Four separate components model the oceans, sea ice, the atmosphere, and the land masses. A fifth component accomplishes coupling between the four scientific components.The components are:

- Ocean Model: Based upon the Parallel Ocean Program (POP) code, but with anisotropic horizontal viscosity included and an improved numerical implementation of the Gent-McWilliams eddy parameterization. River runoff is included directly into the oceans from major river basins. Horizontal resolution averages less than 1 degree, with 40 vertical levels.
- Sea Ice Model: The thermodynamic part of the model uses the physics from the Bitz (University of Washington) ice model. It allows for five or more ice thickness categories and elaborate surface treatment of snow and ice melt physics. An elastic-viscous-plastic dynamic code is utilized to simulate ice motion, using the Hunke and Dukowicz (LANL) approach to the solution of the ice dynamics. The ice model horizontal grid is identical to the ocean model.
- Atmospheric Model: The atmospheric component is the massively parallel version of the NCAR Community Atmospheric Model (CAM). This model includes solar and infrared radiation, boundary physics, and precipitation physics. A T42 grid (approximately 2.5 degree; 128 longitudes and 64 latitudes, with 26 vertical levels) is being used for calibration simulations, and a T85 grid (approximately 1.25 degree; 256 longitudes and 128 latitudes) for the bulk of simulations for the IPCC assessment.
- Land Model: The land model contains completely new biogeophysical and hydrological parameterizations, and demonstrates improved simulation of snow cover and seasonality of runoff (which transports water to the oceans at 0.5 degree resolution).
- Coupler: The coupler's design allows the component models to execute concurrently as separate executables with the information exchange achieved by message passing (MPI). The four scientific modules communicate only with the coupler. Since the component grids are different, there is an interpolation scheme for passing information between the atmosphere component grid and the ocean/sea ice grid that has been developed by Jones (LANL).

mp9 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	0%	0%	0%	100%
First ½ FY 2004	0%	0%	0.1%	0.1%	99.8%
Diff FY04-FY03	-	-	+0.1%	+0.1%	-0.2%

CCSM has two levels of parallelism. The five individual models run concurrently as separate executables in MPMD mode. Each model is parallelized using MPI, OpenMP, or

a hybrid combination. The four scientific models communicate through the coupler, receiving data during one phase of the coupling and sending data during a later phase. The coupling frequency is one per simulated hour for the atmosphere, land, and ice components, and once per day for the ocean component. There is a scientific requirement that the land and ice components compute fluxes before the atmosphere. As a result, there are two distinct integration phases that need to be load balanced.

Because of the relative coarseness of the resolutions currently used by CCSM, the individual components typically scale efficiently up to only 32 or 64 processors. As a result, CCSM is normally run on a fairly low number of processors

Scaling: CCSM represents about 65% of the repo's usage running typically with 64 or 128 tasks.

Code PCM: The parallel climate model is an older climate code. It is a multi-module, coupled code, like CCSM. It uses some of the same modules as CCSM, but some of the modules were less mature and/or were not their own versions. PCM was intended for production use while CCSM was still in development. Now that CCSM is in production PCM represents a diminuishing fraction of the workload.

Scaling: About 36% of usage goes to PCM typically running 32 tasks.. Not all the components that make up PCM are parallelized. The presence of serial bottlenecks prevents the code from running in a highly parallel mode.

Total CPUs	608	352	208	128
Simulated years per day	2.57	2.48	1.57	1.11
Simulated years per day per CPU	.0043	.0072	.0076	.0087

While throughput in terms of simulated years per wallclock day does increase as more processors are added, the efficiency in terms of simulated years per wallclock day per CPU steadily decreases from the smallest configurations, with an extreme dropoff at 608 CPUs. Experimentation continues, but the production IPCC runs will likely be made with a smaller configuration.

NERSC has helped this project by providing preferred queue access, enabling the wait times for these runs to be reduced tenfold.

mp9 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2003	1,134,000	686,000	60%	682,091
FY	3,150,000	2,380,000	76%	2,776,576

2004				

Conclusions:
The least parallel component of the MPMD program group limits the parallelism available to the entire application. At high concurrency MPMD Climate codes have additional synchronization complexity due to each component program module having its own work and communication schedule.

5) Combustion:

For FY 2004 NERSC had 4 combustion projects in production. They received 4.6 percent of the allocation. The three projects discussed here have different characteristics. One of the projects has developed code that scales well to 1024 or more processors, one of the projects is working to eliminate a code module that reduces scaling at 256 processors, and one project is using 256 processors for a particular class of production runs, and focusing code development on a different class of calculations.

PI John Bell, NERSC Repository mp111
John Bell's project, "Applied Partial Differential Equations", develops high-resolution adaptive methods for solving partial differential equations. The current focus applications are magnetic fusion, accelerator design and turbulent reacting flow. Of these applications, computational reacting flow is the most mature and will utilize the bulk of the computational resources. The goal is to develop a high-fidelity numerical simulation capability for turbulent combustion processes such as those arising in furnaces and engines. The key issue in modeling turbulent reacting flow is the interplay between kinetics and the small-scale turbulent eddies in the flow.

mp111, a SciDAC project, had the largest FY 2004 combustion allocation and is characterized by large-scale processor utilization.

Code AMR Combustion: is a system of nonlinear time-dependent partial differential equations describing combustion at low Mach number. The overall computational approach is based on a hierarchical refinement strategy, which clusters the finite difference mesh in regions of interest.

Scaling: This application is the main production code for this repository, accounting for over 95% of the time used. This code scales very well to 256 processors, then efficiency begins to decrease due to a step that is essentially serial. In the long term this piece will have to be rewritten since it prevents some of the largest simulations from being run. While the efficiency begins to drop off after 256 processors, it is still worthwhile running at larger concurrencies to reduce throughput time for a job.

mp111 Project Scaling:

	2,048+	1,024-2,032	512-1,008	256-496	1-240 CPUs

	CPUs	CPUs	CPUs	CPUs	
FY 2003	33.2%	18.3%	27.0%	1.2%	20.4%
First ½ FY 2004	0.1%	1.4%	74.2%	8.7%	15.5%
Diff FY04-FY03	-33.1%	-16.9%	+47.2%	+7.5%	-4.9%

Application runs more efficiently at lower concurrencies, so production runs have settled back to a lower processor count. The Large Jobs Reimbursement program was used to carry out scaling studies that isolated an inefficient code module that has been written for vector computers some time ago. It effectively serializes the code when in this section.

mp111 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	1,200,000	1,012,000	84%	1,074,340	200,000
FY 2004	2,000,000	1,303,500	65%	1,044,685	-

Conclusions: The main problems for this project are with the application design rather than the current hardware. The authors are working to remedy this**.**

PI Ahmed Ghoniem, NERSC Repository mp218

Ahmed Ghoniem's project, "Lagrangian Simulation and Optimization of Turbulent Mixing and Combustion", develop accurate strategies for simulation, control, and optimization of turbulent mixing and combustion--in particular, mixing and combustion at high Reynolds number in complex 3D vortical flows. These flows are typical of the mixing processes employed in almost all combustion-based energy-conversion systems, and thus have wide practical relevance.

mp218 is characterized by large-scale processor utilization.

Code jicf_tree: The governing equations for the transverse jet are the Navier-Stokes equations (PDEs) in three dimensions for slightly viscous, incompressible flow, expressed in vorticity transport form. Solution of the governing equations therefore reduces to integrating ODEs for advection of the vortex elements and for vorticity stretching. An explicit second-order scheme with error estimation and timestep control is used for integration.

Scaling: This application is the main production code for this repository. The application scales well to 1024 processors, typically achieving 65% efficiency with 10^6 elements at this concurrency. As the code is now scalable to 1024 processors and beyond, large-node jobs are now practical and indeed have become the preferred configuration for production runs. Large node jobs enable runs with larger numbers of vortex elements; for example, several 1024-cpu runs peaked with 1.8 million vortex elements describing the flow field. Production runs now focus on the high-resolution, parametric study of actuation mechanisms in the transverse jet flow field running at slightly lower concurrencies in order to preserve limited allocation.

mp218 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	3.7%	72.0%	13.9%	1.1%	9.3%
First ½ FY 2004	3.3%	14.1%	76.8%	4.5%	1.3%
Diff FY04-FY03	-0.4%	-57.9%	+62.9%	+3.4%	-8.0%

Load balance is crucial issue in parallel implementation of treecodes. A dynamic load balancing scheme has been implemented for more efficient parallelization of the treecode After several months of development and debugging on small numbers of processors, the first large-node jobs with the parallel treecode were submitted in spring 2003. Large-node runs were necessary to test the performance of the algorithm on realistic problem sizes. These runs helped compare variants of the dynamic load balancing procedure, to further develop heuristics, and to benchmark performance and scalability. Over the course of the year, this project has taken advantage of the NERSC large scale job reimbursement program to obtain sufficient computational time for testing and iterative development. Based on this years results the code is computation-bound, not communication-bound, for the data and node sizes under consideration. Memory is a severe limitation for this code, especially at higher concurrencies, 1GB per task is not enough considering MPI takes a larger and larger fraction of this. Collective communications are noticeably slower at large numbers of processors, although this is not a major problem.

mp218 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	40,000	51,600	129%	76,692	25,200
FY 2004	465,000	440,000	95%	451,979	-

Conclusions: This project has produced an application that is well engineered, and scales to high processor count. The authors have no doubt that it could scale to considerably higher concurrencies given an allocation of time sufficient to undertake these runs.

PI Arnaud Trouve, NERSC Repository m217

Arnaud Trouve's project, "High-Fidelity Direct Numerical Simulations of Turbulent Combustion - Compression Ignition under HCCI Conditions and NOx Formation in Turbulent Jet Flames", uses direct numerical simulation (DNS) to provide high-fidelity computer-based observations of the micro-physics found in turbulent reacting flows. DNS is also a unique tool for the development and validation of reduced model descriptions used in macro-scale simulations of engineering-level systems. Their code S3D is a compressible Navier-Stokes solver coupled with an integrator for detailed chemistry, and is based on high-order finite differencing, high-order explicit time integration, and conventional structured meshing. The objective is to both re-design S3D for effective use on terascale high-performance computing platforms, and to enhance the code with new numerical and physical modeling capabilities.

m217, a SciDAC project, has the second largest FY 2004 combustion allocation and is characterized by medium-scale processor utilization.

Code S3D: This direct simulation numerical solver (high-order finite difference discretization, high-order Runge-Kutta explicit time integration, characteristic-based boundary conditions treatment, conventional structured uniform/non-uniform computational mesh) treats a system of partial differential equations based on a first-principles description of a compressible, chemically reacting flow (conservation of total mass, chemical species mass, momentum and energy).

Scaling: This is the main code for this project, accounting for the vast majority of the time used. The scaling depends on the dimensionality of the problem studied. For 2D problems of the type studied for the past allocation periods, the program is most efficient at around 256 processors. For planned 3D simulations higher concurrencies should be possible.

m217 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	0%	0.2%	29.1%	70.7%
First ½ FY 2004	0%	0%	0.2%	53.2%	46.6%
Diff FY04-FY03	-	-	+0%	+24.1%	-24.1%

The project is engaged in production runs that are most efficient at 256 processors.

m217 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2003	1,300,000	146,000	11%	144,968
FY 2004	2,900,000	418,500	14%	400,510

6) Materials Science:

In FY 2004 NERSC had 47 materials science projects in production. They received 7.6 percent of the allocation.

PI Wai-Yim Ching, NERSC Repository mp250

Wai-Yim Ching's project, "Theoretical Studies of the Electronic Structures and Properties of Complex Ceramic Crystals and Novel Materials", studies the structure and electronic properties of complex ceramic materials and other novel materials through large scale computations

mp250 is characterized by medium-scale processor utilization.

Code OLCAO: uses the first-principles, density functional theory-based orthogonalized linear combination of atomic orbital method for electronic structure calculations. This is an all-electron method most suitable for complex low-symmetry systems.

Scaling: This application uses about 10% of the time for this project. A new version recently developed, called OLCAO ONDEMAND, scales well to over 512 processors.

Code VASP: Vienna ab-initio simulation package, a plane-wave pseudopotential density functional code. Computes nuclear and electronic structure using density functional theory.

Scaling: VASP accounts for around 90% of the time used by this repository. Depending on the size of the system, VASP can scale up to 1024 processors. The efficiency is only around 40% at this level however.

mp250 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs

FY 2003	0%	12.8%	0%	60.5%	26.8%
first ½ FY 2004	0%	0%	0%	64.1%	35.9%
Diff FY04-FY03	-	-12.8%	-	+3.6%	+9.1%

The VASP code needs very large inter-processor communications bandwidth and low latency (using FFTW).

mp250 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	300,000	574,800	192%	640,433	13,400
FY 2004	525,000	75,500	14%	83,009	-

PI John Cooke, NERSC Repository mp221

John Cooke's project, "Numerical Simulations of Interfaces, Grain Boundaries, and Nanocrystals", investigates the atomic-scale structure and the electronic properties of interfaces, grain boundaries, and nanoparticles using ab-initio density-functional calculations. These projects are developed in conjunction with Z-contrast scanning transmission electron microscopy measurements, that provide atomic-resolution imaging of the structures of interest, and electron energy-loss spectroscopy (EELS) measurements, that provide information on the local electronic structure.

mpp221 is characterized by small-scale processor utilization.

Code VASP: Vienna ab-initio simulation package, a plane-wave pseudopotential density functional code. Computes nuclear and electronic structure using density functional theory.

Scaling: VASP accounts for all the time used by this repository. VASP does not scale well for the systems of interest to the authors.

mp221 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	0.1%	15.6%	0.1%	84.2%
First ½ FY 2004	0%	6.8%	0%	1.4%	91.9%

Diff FY04-FY03	-	+6.7%	-15.6%	+1.3%	+7.7%

mp221 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	248,000	239,600	97%	322,267	-
FY 2004	350,000	121,500	35%	150,268	6,000

Conclusions: VASP is distributed and compiled as a 'black-box', the authors are not able to redesign it to enable more efficient scaling. NERSC consultants have requested permission to provide optimized versions of the code to seaborg users. As yet this has not been finalized. For small systems even these improvements will not lead to better parallel efficiency.

PI Mark Novotny, NERSC Repository mp100

Mark Novotny's project, "Metastability in Materials Science and Scalability of Parallel Discrete Event Simulations", uses the technique of discrete event simulations to investigate magnetization switching in nanoscale magnetic particles and thin films and bulk ferromagnets with domain walls pinned by impurities. Also part of this project is to invetsigate simulations *related* to implementing scalable parallel discrete event simulations (PDES) on massively parallel computers.

mp100 is characterized by large-scale processor utilization.

Code Modeling Asynchronous Automata: These codes (about 20 separate programs) represent various parallelization algorithms that are used in discrete-event simulations of stochastic processes. These algorithms are widely used in designs of communication systems, the traffic on the internet, and event-driven modeling of physical systems in many fields in applied sciences and engineering.

Scaling: These codes represent all the use at NERSC. Each of these codes uses MPI and trivial-parallelization. Regardless of the number of requested SP nodes, all computing processors have balanced loads.

mp100 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	34.1%	36.0%	1.2%	28.8%

First ½ FY 2004	0%	23.3%	62.4%	5.5%	8.8%
Diff FY04-FY03	-	-10.8%	+26.4%	+4.3%	-20.0%

Scaling could go much further, but the kinds of simulations currently being done are best at approximately 800 processors to maximize productivity.

mp100 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	28,000	38,200	136%	38,125	-
FY 2004	134,000	21,000	16%		59,000

Conclusions: Project has no difficulty scaling to very large processor counts.

PI Malcolm Stocks, NERSC Repository gc4

Malcolm Stock's project, "Magnetic Materials: Bridging Basic and Applied Science", is developing a comprehensive capability to model magnetism and magnetic materials with a particular focus on low dimensional (2D, 1D, and 0D) nanosystems (interfaces, nanowires, inclusions).

gc4 had the largest FY 2004 materials science allocation and is characterized by small-scale processor utilization.

Code LSMS: is an order-N first principles method that parallelizes over the number of atoms in the system. Constrained density functional theory is used to describe non-equilibrium magnetic states.

Scaling: This program accounts for about 35% of the total allocation used by this repository. LSMS scales very well, at roughly 80% to well over 1024 processors. The memory requirements of MPI on the SP limit the largest runs that can be done.

CodeVASP: Vienna ab-initio simulation package, a plane-wave pseudopotential density functional code. Computes nuclear and electronic structure using density functional theory.

Scaling: About 40% of the total allocation is spent on VASP. VASP scales reasonably to around 128 processors for systems of interest.

Code pcluster: A plane-wave pseudopotential density functional code. Computes nuclear and electronic structure using density functional theory.

Scaling: About 12% of the allocation. This code does not appear to scale beyond 16 processors.

gc4 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	3.5%	1.1%	0%	0.1%	95.3%
First ½ FY 2004	0%	3.5%	17.1%	1.0%	78.5%
Diff FY04-FY03	-3.5%	+2.4%	17.1%	+0.9%	-16.8%

LSMS could be run at higher concurrency, but this year the project is conserving allocation and is running smaller systems that are more efficient at slightly smaller processor counts. Users of VASP have moved to running at slightly smaller concurrencies to extend allocations. The use of pcluster has dropped from last year to this, moving time out of the smallest column. The user of plcuster is now using VASP at a higher concurrency.

gc4 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2003	760,000	493,600	65%	461,323
FY 2004	1,750,000	430,000	25%	397,717

Conclusions: LSMS scalability is limited by the memory usage of the IBM MPI library. VASP is a third-party product of limited scalability for this project's systems. Pcluster does not scale beyond a single node.

7) Engineering Physics:

In FY 2004 NERSC had 5 engineering physics projects in production. They received 3.2 percent of the allocation.

PI Pui-kuen Yeung, NERSC Repository incite3

Puii-kuen Yeung's project, "Fluid turbulence and mixing at high Reynolds number", conducts direct numerical simulations of unsteady, three-dimensional turbulent flow. They focus especially on the behavior and scaling properties of turbulent velocity fluctuations and the mixing of passive contaminants in the flow.

incite3, an INCITE project, had the largest FY 2004 engineering physics allocation and is characterized by large-scale processor utilization.

Code DNS: solves a system of PDE's. They are transformed to ODE's in time by a Fourier decomposition in space.

Scaling: This code accounts for nearly all the usage by this project. The code's parallel performance is dominated by MPI_alltoall (message size is 100s of KB). The scalar performance is dominated by an ESSL complex scalar 1D FFT.

incite3 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
First ½ FY 2004	2.7%	2.0%	94.2%	0.0%	1.0%

This study covers up to the first half of FY04. At that point incite3 had not ramped up there runs. They are currently running a mix of 512 way and 2048 way jobs. Switch contention induced runtime variability at higher concurrency have been observed similar to those observed in su3_rmd/MILC codes.

incite3 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2004	2,100,000	1,245,000	59%	2,049,604

8) Fusion Energy:

In FY 2004 NERSC had 40 fusion energy projects in production. They received 29 percent of the allocation.

PI Don Batchelor, NERSC Repository m77

Don Batchelor's project, "Electromagnetic Wave-Plasma Interactions in Multi-dimensional Systems", creates simulation codes of wave-plasma interaction which take full advantage of terascale computers.

m77, a SciDAC project, had the fifth largest FY 2004 fusion energy allocation and is characterized by large-scale processor utilization.

Code AORSA: uses a fully spectral method to solve for the wave electric fields in a multi-dimensional plasma, including the full integral representation of the plasma current.

Scaling: 60% of project cycles went to 2D calculations and 40% to 3D calculations. While a large fraction (40%) of the 3D work is done using 1024 tasks, most work is done using 2048 tasks. The scaling of the code is strongly influenced by a ScaLapack dense solve.

m77 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	64.0%	17.0%	9.5%	9.2%	0.4%
First ½ FY 2004	98.7%	0%	0%	0.2%	1.2%
Diff FY04-FY03	+34.7%	-17.0%	-9.5%	-9.0%	+0.8%

This project pursues higher concurrency largely as a means of attaining more memory for large scale structured grids. The code does minimal I/O to disk thereby avoiding one high concurrency pitfall. The throughput of this project benefited from changes to queuing policies that favor larger scale jobs.

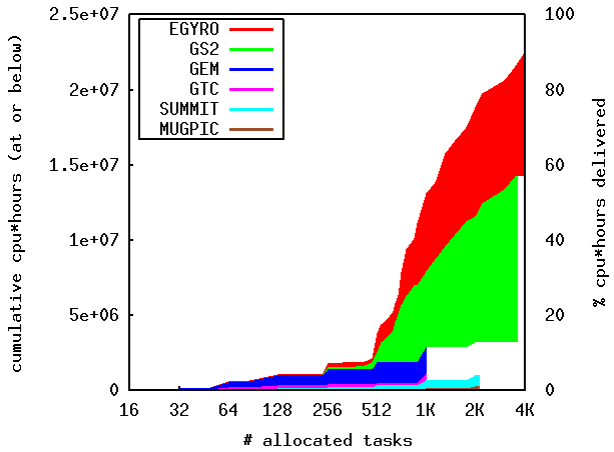
m77 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	1,200,000	644,000	54%	1,154,458	320,000
FY 2004	1,165,000	940,000	81%	769,466	100,000

PI Bruce Cohen, NERSC Repository gc3

Bruce Cohen's project, "Plasma Microturbulence", develops a predictive ability in modeling turbulent transport due to drift-type instabilities in the core of tokamak fusion experiments, through the use of three-dimensional kinetic and fluid simulations and the derivation of reduced models. The project conducts 3D simulation (both flux-tube and global) of electromagnetic and electrostatic drift-wave turbulence and concomitant transport of energy and plasma (including neoclassical transport effects) in toroidal fusion experiments (for stellartors as well as tokamaks).

gc3, a SciDAC project, had the largest FY 2004 fusion energy allocation and is characterized by large-scale processor utilization.



They use three main classes of simulation algorithms to study core tokamak microturbulence: gyrokinetic particle-in-cell (GK PIC), 5-D Eulerian gyrokinetic (EGK), and to much lesser extent the gyro-Landau-fluid (GLF). In each case, the simulation domain can be either global or annular (fluxtube).

Code GTC: The GTC code developed at PPPL is a global GK PIC in general geometry capable of interfacing with numerical equilibria and is most suitable for studying small aspect-ratio tokamaks and stellarators. The GTC code is written in Fortran 90 with MPI/OpenMP for MPP platforms. The code uses ESSL math library optimized for IBM-MPP machines and also uses HDF5 for parallel I/O. FFT is used in the code only for diagnostic purposes.

About 9 percent of the project cycles went to GTC.

Code GYRO: The EGK code GYRO uses FORTRAN (95% F90, 5% F77), input managed by Python, VTK visualization in C++. Bourne shell scripts control interactive and batch execution. LAPACK for dense linear algebra, UMFPACK for sparse linear algebra, MPI for parallelization.

About 35% of the project cycles went to GYRO.

Code GS2: The flux tube EGK code (GS2) is a nonlinear generalization of a widely used linear, implicit, electromagnetic gyrokinetic code. All the linear dynamics remain fully implicit. The nonlinear terms are integrated explicitly with a pseudo-spectral Adams-Bashforth algorithm. The code is modular in design, and is written in Fortran 95. Parallelism is implemented with modules that allow either MPI or (where available) a mix of MPI and SHMEM libraries to be used. Standard multiple domain decomposition techniques are used for four dimensions at a time.

About 30% of the project cycles went to GS2.

Code GEM/PG3EQ/SUMMIT:

Scaling : About 20% of the project cycles go to this suite of codes. The PG3EQ code is presently primarily undergoing development. The 2D domain decomposition scales well to 256 processors, which is as many as is needed for good turnaround on even very large flux-tube runs. With the merger of PG3EQ and GEM electron physics in SUMMIT and the extension to global physics, it is anticipated that 1024 way runs will become useful. The 2D domain decomposition scheme permits such runs, but it is not known if communication bandwidth will limit the parallel performance.

gc3 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	1.8%	25.6%	29.3%	15.5%	17.8%
First ½ FY 2004	0.2%	30.5%	52.7%	4.9%	11.7%
Diff FY04-FY03	-1.6%	+4.9%	+23.4%	-10.6%	-6.1%

The shift to larger concurrency shown above is the result of a shift in workload toward the more scalable GS2 and EGYRO codes.

gc3 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	2,000,000	1,774,000	89%	2,238,988	320,000
FY 2004	3,500,000	2,848,000	81%	3,031,529	52,500

This project had scaled prior to FY 2003 and took advantage of the Reimbursement Programs to expand their allocation. This project always needs more time than they request or get.

PI William Dorland, NERSC Repository mp217

William Dorland's project, "University of Maryland Fusion Energy Research", focuses on:

- 3D simulation of particle, ion and electron energy transport in the core and edge region of tokamak plasmas using the two-fluid Braginskii code DBM and the electromagnetic gyrokinetic code GS2;
- 3D simulation of high-beta disruptions, sawtooth crashes for tokamak plasmasusing the toroidal resistive MHD TORMHD with plasma shaping;
- 2D and 3D simulations of novel centrifugal confinement devices using MHD codes; and
- 2D and 3D full particle and two-fluid simulations of magnetic reconnection.

mp217 is characterized by medium-scale processor utilization.

Code DBM: solves the time-dependent Braginskii fluid equations (a series of PDE's) for the scalar potential, magnetic flux, electron and ion temperatures and densities, and the parallel flow velocity in a flux tube coordinate system.

Scaling: 28% of the project cycles go to DBW, which is typically run on 32 processors.

Code GS2: solves the coupled Vlasov-Maxwell equations. The Maxwell equations for the electromagnetic fields are linear pde's. The Vlasov (or with collisions, the Fokker-Planck) equation is a nonlinear pde.

Scaling: 4% of the project cycles go to GS2, which is typically run on 512 processors.

Code TORMHD: solves the time-dependent magnetohydrodynamic equations (a series of PDE's) for the magnetic field, temperature, density, and momentum on a toroidal grid.

Scaling: 8% of the project cycles go to TORMHD, which is typically run on 32 to 64 processors.

Code F3d: Explicitly solves the two-fluid MHD partial differential equations with the addition of the Hall terms and electron inertia in Ohm's law. Variables stepped forward in time include: the magnetic field, the density, the fluid velocity, and the electron and ion pressure.

Scaling: 16% runs 256-512 way

Code P3d: solves Maxwell's equations with a leapfrog trapezoidaltime-stepping scheme. Errors introduced to the fields bydiscretization are corrected by solving Poisson's equation with amultigrid algorithm. The particle positions and velocities are advanced with a Boris algorithm.

Scaling: 16% of the project cycles go to P3d, which is typically run on 512 to 1024 processors.

mp217 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0.7%	5.7%	11.3%	17.5%	64.8%
First ½ FY 2004	1.1%	5.1%	13.0%	21.9%	58.8%
Diff FY04-FY03	+0.4%	+0.6%	+1.7%	+4.4%	-6.0%

This shift is largely due to larger concurrency P3D runs. This group has successfully scaled their codes largely in order to accommodate large memory sizes. Over the last year code devlopment was done to improve the I/O portion of the code at start up and termination. The I/O portions of their main code P3D/F3D still present a scaling bottleneck. NERSC is working on methodologies for improving parallel I/O.

p3d: p3d scales well to 1024 processors, with the exception of our I/O routines. Although workarounds exist to somewhat alleviate the problem, we are currently working with the NERSC consultants to solve the underlying problem and achieve 61% parallel efficiency at 1024 tasks.

f3d: We are still actively pursuing to understand why the code does not scale well to 1024. Preliminary results suggest poor scaling in the inter-node communication based on a large volume of small messages. This code scales worse than p3d because there is less computation (physics) done per step requiring communication.

DBM and **TORMHD** : These codes are not typically run on 64 or more nodes because runs that large would consume too much of our allocated time. For each set of parameters, the codes must be restarted many times to obtain a sufficiently long time sequence of the turbulence.

mp217 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	300,000	574,800	192%	640,433	33,200
FY 2004	950,000	468,500	49%	523,504	11,000

Conclusions:

The group states that science output per time drives the choices about scale and parallelism in their runs. Improvements in parallel I/O or a larger allocation would expand these choices. The group would like to expand the size of problems addressed but needs to resolve code issues.

PI Steve Jardin, NERSC Repository mp288

Steve Jardin's project, "3D Extended MHD simulation of fusion plasmas", consists of a multilevel comprehensive plasma simulation code package, and applications of various levels of the code to increasingly more realistic fusion problems.

mp288, a SciDAC project, is characterized by small-scale processor utilization.

Code M3D: M3D is built around the PetSC package. It models 3-dimensional toroidal geometry using a structured finite difference mesh in the toroidal direction and an unstructured finite element mesh in the poloidal plane. The M3D code is solving the MHD equations, which are similar to the Navier Stokes equations but are higher order because they contain the effects of the magnetic field.

Scaling: This code accounts for most of the group's usage. There is also a small fraction (5%) of related CHOMBO/AMR workload.

mp288 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0.2%	0.5%	1.6%	0.3%	97.5%
First ½ FY 2004	0%	0%	0%	0%	100%
Diff FY04-FY03	-0.2%	-0.5%	-1.6%	-0.3%	+2.5%

The physics of their problems dictate that they need to run for long times, or many time steps. The spatial resolution that they normally use allows their problems to scale well up to about 64 processors. In order for the codes to continue to scale well up to 1024 or more processors, they would need to increase the problem size (i.e., weak scaling). This is needed for convergence studies and difficult cases. However, when they increase the problem size, they also need to use more time steps for a given problem. Their allocation size prevents them from doing very many of these big runs per year.

mp288 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2003	600,000	559,000	93%	492,169
FY 2004	1,400,000	1,021,000	73%	799,880

Conclusions:

Main interesting problem sizes have limited scaling due to surface to volume ratio. Some larger problems are of interest but are expensive to run.

PI Wei-li Lee, NERSC Repository mp19

Wei-li Lee's project, "3D Gyrokinetic Particle Simulation of Turbulence and Transport in Fusion Plasmas", studies first-principles plasma transport using the global toroidal gyrokinetic particle simulation code GTC.

mp19 had the third largest FY 2004 fusion energy allocation and is characterized by large-scale processor utilization.

Code GTC: is a global toroidal code which solves the gyrokinetic Vlasov-Maxwell equation using the particle-in-cell (PIC) method for the dynamic equations and iterative methods for the ellipitic Poisson's equation. The operations are carried out in the configuration (real) space using magnetic coordinates with a field-line following mesh.

Scaling: GTC accounts for 52% of mp19's usage and is typically run on 64-1024 tasks, At lower concurrency (64 tasks) MPI only is used. For higher concurrency runs MPI and OMP becomes more efficient for GTC than MPI alone. If allocation was not an issue mp19 would run GTC mostly 1024 way as this provides the shortest time to solution. The group reports that they must be very careful with 1024 way calculations as they can quickly exhaust their allocation running at this scale.

Code XGC: The X-point included Guiding Center code is a Monte Carlo particle code for a neoclassical calculation of the edge pedestal build up. Hamiltonian dynamics are used to model heat flow from the core and the impact of turbulence transport.

Scaling: XGC accounts for 33% of usage and is run on 512-2048 tasks. It requires minimal communication between tasks and scales nearly linearly.

mp19 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0.0%	30.2%	58.6%	7.8%	3.4%
First ½ FY 2004	8.2%	44.5%	37.0%	1.1%	9.2%
Diff FY04-FY03	+8.2%	+14.3%	-21.6%	-6.7%	+5.8%

Due to an increased allocation the group is able to run GTC at higher concurrency .A larger portion of XGC runs were done in early FY04 also contributed to the change in overall scaling.

mp19 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	1,200,000	1,283,800	107%	1,524,701	226,200
FY 2004	2,800,000	1,490,000	53%	1,838,820	80,000

One of the main authors of GTC reports "Regarding the merits of running on different number of nodes, 1024 tasks is the best number of nodes for GTC at the present time. The GTC code scales up very well from 128 tasks to 512 tasks. There is a drop in efficiency from 64 to 128 tasks because a mixed MPI/OpenMP parallelization is utilized for 128 or more tasks. For 1024 tasks or more, our code scales up well computationally, but efficiency in term of physics drops."

PI Pitch Pindzola, NERSC Repository m41

Mitch Pindzola's project, "Computational Atomic Physics for Fusion Energy", implements collision codes to study a wide range of atomic collision processes present in the edge region of controlled fusion plasmas.

m41, a SciDAC project, has the second largest FY 2004 fusion energy allocation and is characterized by large-scale processor utilization.

Code TDCC: Solves the time-dependent close-coupling for electron and photon ionization from two active electron target atoms.

Scaling: This code accounts for 68% of usage and is run 64-2304 way. Its scaling efficiency depends on problem size. For the systems of interest to this project the code scales well to 1024 tasks.

Code TDSE: Solves time-dependent Schrodinger equation for bare ion collisions with one active electron target atoms.

Scaling: The TDSE code accounts for 1% of usage and is run 256-2048 way. This code partitions a large 3 dimensional lattice over a set of 10 nodes, then partitions the impact parameters over 10 sets of lattices for a total of 100 nodes (1600 processors). The data decomposition must be properly matched with the number of processors. In some cases this rules out certain concurrencies.

Code R-matrix:

Scaling: A variety of different R-matrix codes account for 4% of usage. They are typically run 512-1024 way. Parallel I/O performance inhibits the scaling of these codes. There is currently work under way to address the I/O problems in the PSTGR codes.

Code S3D:

S3D calculates excitation and charge-transfer cross-sections associate with ion-atom collisions by direct integration of 3-dimensional Schrodinger equations.

Scaling: 26% 512-1024 way. The calculations are independent. Queuing time and time to solution drive the number of tasks at which S3D is run.

m41 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	3.4%	25.0%	8.4%	28.7%	34.6%
First ½ FY 2004	7.9%	56.2%	1.4%	7.8%	26.7%
Diff FY04-FY03	+4.5%	+31.2%	-7.0%	-20.9%	-7.9%

NERSC consultants reworked the MPI in one of the R-matrix codes, improving performance 47%. The main changes where to how MPI_Allreduce was used. Work remains to be done improving the parallel I/O of some codes.

m41 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	300,000	644,800	215%	937,262	213,000
FY 2004	1,150,000	1,472,000	128%	1,473,284	100,000

This project took advantage of the FY 2003 Reimbursement Program to scale their codes. This project always needs more time than they request or get.

They benefited from the reimbursement program through the ability to run very large jobs enabling important convergence tests. The codes were examined with respect to many crucial parameters of the problem, for example box size, mesh spacing, and number of continuum channels included in the calculation. Results showed that convergence is quite difficult to achieve and that further large jobs will be needed to get accurate results. These jobs are currently being planned.

These calculations will provide the first theoretical description of the four-body Coulomb system which, when results are ready, be of great interest to the atomic physics community in comparing to recent electron scattering and photoionization experiments.

Conclusions:
This group's codes are a mixture of domain partitioning with message passing and task partitioning with no message passing. They have developed production codes which scale well for the systems they are studying. The group also does development runs at lower concurrency to develop new algorithms and new codes, or at times to maximize throughput.

PI Abhay Ram, NERSC Repository m224

Abhay Ram's project, "Computational studies in plasma physics and fusion energy", simulates ICRF and LHRF heating and current drive models which can be readily applied to on-going rf experiments in tokamaks such as the Alcator C-Mod, JET, and ASDEX Upgrade.

m224 had the fourth largest FY 2004 fusion energy allocation and is characterized by large-scale processor utilization.

Code GS2: is used to simulate drift-type turbulence and transport in realistic tokamak geometries. GS2 is a parallel, initial-value, gyrokinetic, Vlasov, semi-implicit algorithm implemented in a flux-tube geometry.

Scaling: 95% of the project cycles go to GS2, which is run 1024 and 2048 way. This code requires a particular attention to matching problem size and concurrency. The volume of communication and parallel efficiency depend strongly on the parallel layout. The authors are working on scripts which can provide users with a good match between individual problems and processor layout.

m224 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	67.8%	3.3%	13.6%	2.0%	13.3%
First ½ FY 2004	70.4%	0.1%	21.2%	0.3%	8.0%
Diff FY04-FY03	+2.6%	-3.2%	+7.6%	-1.7%	-5.3%

m224 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	422,400	240,000	57%	251,680	-
FY 2004	4,000,000	1,225,000	31%	927,484	100,000

PI Alan Turnbull, NERSC Repository mp94

Alan Turnbull's project, "Simulation of Magnetically Confined Fusion Plasmas", is a comprehensive program of simulations for the equilibrium, stability, and transport of various toroidal magnetic confinement configurations relevant to fusion energy, with an emphasis on the scientific basis for integrated theoretical and numerical modeling and the validation of the models against DIII-D.

mp94 is characterized by medium-scale processor utilization.

Code NIMROD: The NIMROD code is designed to perform nonlinear, initial-value simulations of long-wavelength phenomena in fusion-reactor-relevant plasmas using the extended magnetoydrodynamics model MHD plus two-fluid and kinetic effects to simulate the electromagnetic and fluid behavior. By using an object-based paradigm, NIMROD is able to provide flexibility in physics models - single fluid or two-fluid MHD models with analytic or gyrokinetic closures - and flexibility in the geometry - allowing for studies of any axisymmetric fusion concept.

Scaling: 70% of the project cycles go to NIMROD which is typically run 512 way. NIMROD scales well to 128 processors but scaling deteriorates beyond this for the standard problem sizes of usual interest to this group. For a zero-beta RFP test computation for 10 time steps, using the SuperLU_DIST library, and a 48x48 mesh of bicubics with lphi=7 (0<=n<=42); this is a reasonably realistic test case for timing: For a fixed problem scaling from 30 to 352 processors with the 48x48x7 mesh:

```
 2 blocks, 15 layers,  2 nodes/30  procs,  3.16 GFlops, 25.5 Gb
 4 blocks, 15 layers,  4 nodes/60  procs,  5.36 GFlops, 31.9 Gb
 9 blocks, 15 layers,  9 nodes/135 procs,  8.26 GFlops, 49.3 Gb
16 blocks, 15 layers, 16 nodes/240 procs, 10.7  GFlops, 77.3 Gb
16 blocks, 22 layers, 22 nodes/352 procs, 12.6  GFlops, 93.4 Gb
```

For an increasing problem scaling up to 1548 processors:
For a 72x73x7 mesh:

```
36 blocks, 22 layers, 50 nodes/792 procs, 17.4 GFlops, 397 Gb
```

For a 72x72x8 mesh:

```
36 blocks, 43 layers, 100 nodes/1548 procs, 28.5 GFlops, 792 Gb
```

Efficiency decreases gradually, but there does not appear to be a sharp fall-off.

Although they do not have detailed information on the amount of data communicated during NIMROD simulations, the fact that there is a large number of different communications calls during a time-step and the small amount of memory (the 8-node test uses less than 1/8 of the available memory) relative to computations based on probabilistic models imply that latency is much more important for efficient communication with NIMROD than bandwidth. This is supported by their experience with running NIMROD on PC clusters that use ethernet communication exclusively.

Code GYRO: GYRO is a global nonlinear electromagnetic gyrokinetic code (Global EGK), which solves the nonlinear gyrokinetic-Maxwell equations in shaped tokamak geometry. It is the only nonlinear microturbulence code in existence that is both global and electromagnetic. GYRO distributes 3 indices of the distribution function and keeps 2 on-processor. The distributed indices can be permuted with on-processor indices via generalized transpose operations. The collisionless time advance is a second-order implicit-implicit Runge Kutta algorithm.

Scaling: 22% of the project cycles go to GYRO, mostly run 1024 way with some 2048 way usage. The previous performance degradation in GYRO due to all-to-all communication was eliminated. The code now scales well up to 1024 processors. For electromagnetic simulations with 32 modes, we obtain nearly linear scaling up to 512 processors, with good scaling to 1024. With more modes, scaling improves to larger numbers of processors.

Code BOUT: BOUT is a turbulence code developed at LLNL, which solves the electromagnetic collisional Braginskii equations in the complicated geometry near and across a magnetic separatrix.

Scaling: 8% of project cycles go to BOUT, which is run 128 way. BOUT code scaling was tested on the T3E where the code performance scaled close to linear with the number of processors up to 64 processors. The parallelized version of BOUT works well with a poloidal domain decomposition, with a factor of 69 speedup for 60 processors on the T3E-900. The super-linear speedup on the T3E is likely due to the better utilization of cache memory for the larger number of processors. This case was extended to 120 processors on the T3E and the data was found to be on the same offset linear curve. The code has been ported to Seaborg with more than a factor of 2speedup over the T3E. On Seaborg, the code scales well to 64 processors for small problems and to 128 for larger problems.

mp94 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs

FY 2003	0.0%	10.3%	16.8%	8.1%	64.8%
First ½ FY 2004	0.1%	0.1%	38.3%	32.0%	29.7%
Diff FY04-FY03	+0.1%	-10.2%	+21.5%	+23.9%	-35.1%

mp94 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	1,272,000	752,200	59%	870,286	86,200
FY 2004	2,300,000	975,000	42%	1,026,093	-

This project took advantage of the FY 2003 Reimbursement Program to scale their codes.

Conclusions:

This group has scaled their NIMROD and GYRO workloads up between FY03 and FY04.

PI Harold Weitzner, NERSC Repository mp338

Harold Weitzner's project, "Advanced Toroidal Theory and Computation", uses nonlinear 3D MHD simulations to study the large scale dynamical behavior of magnetic fusion experiments. Present efforts include simulations of pellet driven disruptions in tokamaks, high beta disruptions in spherical tori (e.g. NSTX), and resistive evolution of compact quasi - axisymmetric stellarators (NCSX).

mp338 is characterized by large-scale processor utilization.

Code XGC: particle code to model plasma transport in the edge region of divertor tokamaks. XGC (X-point included Guiding Center) code is a Monte Carlo particle code for a neoclassical calculation of the edge pedestal build-up.

Scaling: 90% of project cycles are used by XGZ, typically run with 2048 tasks and some usage at 1024 tasks. The tasks in this code are nearly independent. A negligible amount of time is spent communicating.

mp338 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0.0%	51.2%	32.5%	4.5%	11.8%

First ½ FY 2004	44.8%	26.6%	15.1%	0%	13.5%
Diff FY04-FY03	+44.8%	-24.6%	-17.4%	-4.5%	+1.7%

mp338 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	80,000	210,400	263%	213,725	-
FY 2004	1,500,000	700,000	47%	708,519	100,000

Conclusions: Problem size and time to solution drive choices in concurrency.

9) Geosciences:

In FY 2004 NERSC had 9 geosciences projects in production. They received 1 percent of the allocation. The two geosciences projects represented in this survey stated that they did not have sufficient allocation to run at the concurrencies that they would have preferred.

PI Michael Hoversten, NERSC Repository m234

Michael Hoversten's project, "Realistic three-dimensional velocity anisotropy modeling", simulates seismic wave propagation through 3D structurally complex and anisotropic (directionally varying seismic velocities) geological models.

m234 has the fourth largest geosciences allocation and is characterized by large-scale processor utilization.

CodeAnisWave3D: uses a finite difference solution to the wave equation which replaces the partial derivatives of velocity and stress in space and time by their second-order finite difference approximations.

Scaling: This is the principal application used by this project. The code scales well to 512 processors for the type of systems currently being studied. The authors note that scaling is limited by software design. Currently there is a 2D decomposition into tasks and that a 3D decomposition would be beneficial for scaling to much higher processor counts.

Collective communications are also seen to be a barrier to scaling, showing poor performance at high concurrencies.

m234 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	15.5%	39.6%	20.2%	24.7%
First ½ FY 2004	0%	0%	97.5%	0%	2.5%
Diff FY04-FY03	-	-15.5%	+57.9%	-20.2%	-22.2%

The authors report that the principal constraint is lack of allocation. Each large simulation project requires 20,000-30,000 hours, and some of the allocation needs to be used for scaling and benchmark studies. Very large jobs are not feasible within the current allocation.

m234 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2003	200,000	73,000	37%	52,766
FY 2004	150,000	49,000	33%	30,551

Conclusions: The project is limited by allocation constraints, or else could scale to somewhat higher processor counts. Scaling to much higher processor counts would probably require a code rewrite to decompose work into more tasks.

PI Gregory Newman, NERSC Repository m372

Gregory Newman's project, "Faster and Better solutions to three-dimensional electromagnetic geophysical inverse problems", develops inverse solutions for three dimensional electromagnetic inverse problems. Their goal is to image the subsurface electrical properties at an unprecedented scale in 3D.

m372 had the largest FY 2004 geosciences allocation and is characterized by medium-scale processor utilization.

Code mt3d: Solves vector Helmholtz equation for electric fields for magnetotelluric problem using a finite difference scheme.

Scaling: This code accounts for approximately one third of the time used. The code scales well to 1000 processors.

Code teminv: Solves the transient electromagnetic reverse problem.

Scaling: This code accounts for approximately two thirds of the time used. This code has demonstrated scalability to 800 processors on seaborg. Currently the computational bottleneck in our optimization work is the time required to solve the forward problem using standard iterative Krylov subspace methods, such as qmr etc.

m372 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
First ½ FY 2004	0%	0%	41.6%	53.9%	4.5%

The allocation was insufficient to run systems at higher concurrencies.

m372 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2004	1,000,000	238,000	24%	231,033

Conclusions:

This group plans to continue software development to improve scalability. Ongoing work with Sandia will focus on improving the multi-level library. Improvements here show potential in reducing the computation time required to solve the 3D inverse problem.

10) Quantum Chromodynamics:

In FY 2004 NERSC had 10 QCD projects in production. They received 11 percent of the allocation. Most QCD codes are characterized by small memory parallel sparse matrix calculations. In lattice QCD the lattice sizes of interest are often of fixed size across a large number of runs. As a result parallel scaling of these codes becomes tightly constrained as the ratio of computation to communication drops below a critical value.

PI Keh-Fei Liu, NERSC Repository mp7

Keh-Fei Liu's project, "Lattice QCD Monte Carlo Calculation of Hadron Structure", implements the overlap fermion numerically to test chiral symmetry and scaling via the calculation of hadron masses, quark masses, and the chiral condensate.

mp7 had the second largest FY 2004 QCD allocation and is characterized by small-scale processor utilization.

Code overlap quark propagator (XLATQCD): The new overlap fermion action involves a matrix sign function. They approximate the square root of the matrix by the optimal rational fraction approach and we invert the matrix with conjugate gradient with multiple mass algorithm. To speed up the convergence, we project out some of the smallest eigenvalues and treat the sign function of these states exactly. The overall inversion of the quark matrix to obtain the quark propagator is also done with conjugate gradient with multiple quark masses.

This code accounts for most of the usage within this project.

mp7 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	3.2%	0.6%	15.6%	80.6%
First ½ FY 2004	0%	4.5%	0%	0.9%	94.6%
Diff FY04-FY03	-	+1.3%	-0.6%	-14.7%	+14.0%

The code sub-divides a lattice into adjacent sub-lattices, with each sub-lattice assigned to a single processor. For a fixed number of lattice sites, as the number of nodes is increased, the size of each sub-lattice decreases. Thus the surface to volume ratio increases, thus increasing the proportion of inter-sub-lattice communication, and thus reducing efficiency. For 16^3 x 28 lattices, we found it most cost effective to run on 128 processors. This is born out by strong-scaling examples Performance on a 16^3 x 24 lattice improves 40% moving from 256 to 512 processors. Beyond 512 tasks to surface to volume ratio increase sufficiently to slow the code down.

On 28^3 x 32 lattices, we will obtain comparable efficiency on 800 processors. Our codes will become more efficient with smaller interconnect latency.

mp7 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)

FY 2003	1,200,000	790,600	66%	963,652	25,600
FY 2004	2,330,000	1,165,000	50%	1,086,268	-

PI Donald Sinclair, NERSC Repository mp27

Donald Sinclair's project, "Lattice QCD at finite temperature and densities and the spectrum of lattice QCD with chiral 4-fermion interactions", conducts simulations of lattice QCD with an irrelevant chiral 4-fermion interaction (XQCD) which renders the Dirac operator non-singular in the chiral limit allowing simulations with massless quarks, or arbitrarily light quarks.

mp27 had the third largest FY 2004 QCD allocation and is characterized by small-scale processor utilization.

Code su3immu: The fermion fields in the problem are represented by gaussian-distributed stochastic complex 3-vectors on the sites of the lattice. The equations for the `time' evolution of the gauge fields, which are SU(3) matrices residing on the links, involve the inverse of the Dirac operator, a sparse matrix over the sites and colors which itself depends on the gauge fields, acting on the fermion fields. The sparsity pattern of the Dirac operator is that of the 4-dimensional discrete Laplacian.

Scaling: This code represents 15% of usage and has largely been moved off of seaborg as a result of changes in queuing policies.favoring larger concurrency jobs. It now runs well on a 2.4Ghz Pentium4 Linux cluster.

Code xqcd: The underlying mathematical/physical model is a Hybrid Molecular-Dynamics formulation of QCD on a Euclidean space-time lattice. It requires the solution of Hamilton's/Lagrange's equations for a system of coupled fields residing on the sites and links of a 4-dimensional hypercubic lattice with periodic/antiperiodic boundary conditions. These are a set of coupled non-linear ordinary differential equations.

Scaling: This code consumes 80% of delivered cycles and is most often run with 192 tasks on 12 nodes and is dominated by a parallel sparse matrix vector operation.

mp27 Project Scaling:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0%	0.3%	0.1%	0.1%	99.6%
First ½ FY 2004	0%	0%	0.6%	0%	100%
Diff FY04-FY03	-	-0.3%	-0.1%	-0.1%	+0.4%

A concurrency of 192 tasks has been identified as optimal for the fixed problem size of interest. The performance drops from 500 Mflops to 100 Mflops as the concurrency is increased from 192 to 512 tasks. Experiments up to 1024 tasks were conducted under the large scale reimbursement program, but currently production work continues at the previously identified concurrency.

The scaling of the code can be modeled as follows. On 512 processors, each task has a 16^2*1*1 piece of the lattice and performs 256*1212 = 310272 floating point operations per conjugate gradient iteration (where the code performs most of its floating point operations). At around 500 Mflops this should take about 600 mu-secs. During this time each processor exchanges 8 messages of length 256*24 = 6144 bytes with processors on other nodes (4 sent, 4 received). For a switch bandwidth of 25 Mbytes/sec/processor this will take about 2000 mu-sec. This would reduce the performance to [600/(600+2000)]*500 = 115 Mflops if communication and computation cannot be overlapped, or [600/2000]*500 = 150 Mflops if they can. The additional performance loss is presumably due to the 2 MPI_Allreduce global sums per iteration. At higher concurrencies the cost of the of this reduction would increase.

mp27 Allocation:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used
FY 2003	1,600,000	762,600	48%	916,087
FY 2004	4,000,000	940,000	24%	933,286

Conclusions:
The computations of interest are for a fixed problem size which is set by the physics of interest and the goal of being nearly cache resident on each CPU. Increasing concurrency requires decreasing message size and this incurs a penalty in terms of communication bandwidth. The bandwidth of small message ~15KB dominates the scaling performance of this work. The ability to effectively overlap communication with computation or larger caches would also benefit this project.

PI Doug Toussaint, NERSC Repository mp13

Doug Toussaint's project, "QCD with three flavors of improved Kogut-Susskind quarks", conducts simulations with three flavors of dynamical quarks, including a zero temperature simulation with a light quark mass of 0.1 times the strange quark mass and the study of the high temperature behavior of QCD at a lattice spacing of 1/8T.

mp13 had the largest FY 2004 QCD allocation and is characterized by large-scale processor utilization.

They use a Monte Carlo simulation to calculate the Feynman path integral for interacting quarks and gluons. This basically amounts to generating sample configurations of the variables with a probability proportional to their weight in the path integral, and then evaluating quantities like energy or susceptibilities by averaging over these sample configurations. The algorithm for generating these samples is conceptually identical to the way the air molecules in a room reach their equilibrium Boltzmann distribution.

Code MILC: They have developed a family of codes, the "MILC codes", for the simulation of QCD with Kogut-Susskind dynamical quarks, Wilson dynamical quarks, and in the quenched approximation. The high level algorithm for the generation of sample configurations is a molecular dynamics evolution of the field variables through a fictitious "simulation time", which brings the system to an equilibrium in which each field configuration appears with the desired probability. Physical observables are then computed by averaging the quantities evaluated in the set of sample field configurations. In computing the "force" in this molecular dynamics evolution a sparse matrix must be inverted. This is done via the conjugate gradient algorithm.

mp13 Scaling Characteristics:

	2,048+ CPUs	1,024-2,032 CPUs	512-1,008 CPUs	256-496 CPUs	1-240 CPUs
FY 2003	0.1%	36.4%	2.6%	22.7%	38.2%
First ½ FY 2004	3.8%	79.8%	4.0%	0%	12.4%
Diff FY04-FY03	+3.7%	+43.4%	+1.4%	-22.7%	-25.8%

There are two distinct types of calculations represented in the above data.
High temperature calculations require smaller sized grids and do not scale up. These runs comprise most of the 1-240 CPU runs. Increasingly the group is doing these calculations off of seaborg (moved to Linux clusters) so that their allocation can be devoted to the low temperature calculations with require large grids and can not be done elsewhere.

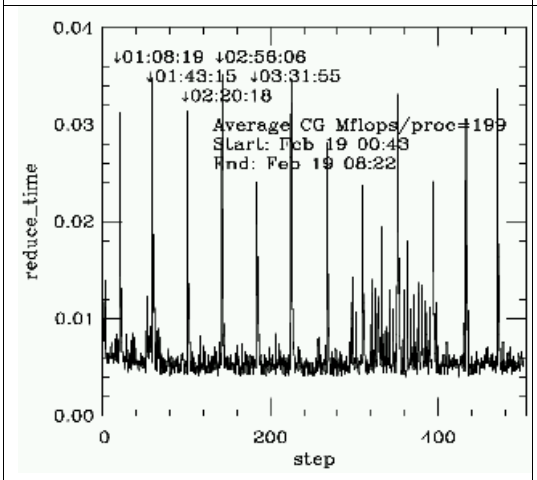
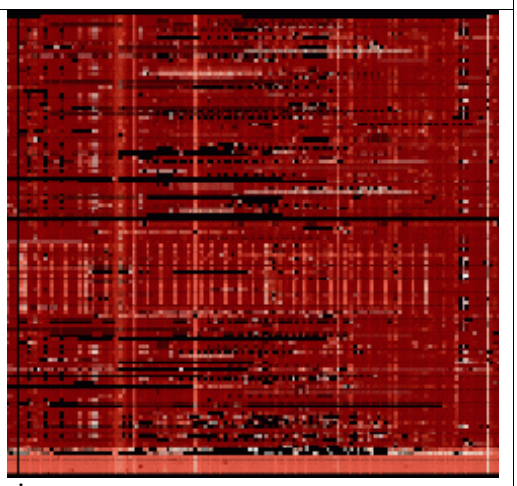
The scaling bottleneck for this code is the MPI_Allreduce .in the conjugate gradient step. Poor scaling performance in the range of 512 tasks was largely resolved after consultancy with USG in May 2003. Using 15 tasks per node (as opposed to 16) allowed good performance into the range of 1K tasks. Leaving a CPU per node out of the compute pool is a technique for mitigating poor MPI_Allreduce performance at high concurrency. For this code, the sacrifice of one CPU per node is more than compensated for in terms of efficiency by improvement in MPI_Allreduce performance. On 1200 tasks using 15/16 cpus cuts the MPI_Allreduce time by a factor of three.

Table 1: Distribution of times spent in communication and number of MPI calls in a 1200 MILC/SU3_RMD run.

MPI Call	% of time	Buffer Size MB	% of calls
MPI_Allreduce	51.7	8.42	1.6
MPI_Wait	36.7	0.0	64.5
MPI_Isend	5.9	2.5	33.7
MPI_Recv	3.4	2.8	~0.0
MPI_Barrier	1.5	0.0	~0.0

Scaling to 2K tasks nominally incurs only additional 10% efficiency loss, but significant problems with performance variability arise. Fluctuations in performance of a factor of 11 have been observed over the course of a run. This code reports performance data per iteration, hundreds of time over the course of a long run. The magnitude of these fluctuations increases with concurrency. The duration and frequency of the fluctuations coincides with GPFS activity coming from outside the job itself. GPFS and MPI use the same communications fabric on the SP. Contention over the switch from I/O workload from other jobs or system activity often has minimal effect on lower concurrency jobs. At higher concurrency and especially for codes dominated by fully synchronizing MPI calls, this contention can become a significant source of performance variability.

The performance variability of this code at high concurrency was studied in detail by NERSC USG staff and Doug Toussaint in late 2003 and early 2004. Starting from performance data given from production runs USG and NSG identified GPFS activity (reads in particular) that consistently coincide (both in time and intensity) with the performance fluctuations. Several runs were examined and the GPFS contention was seen to come from a variety of user and system activity.

User Data : MPI_Allreduce performance	System Data: system wide sar read rate
	 time ----->

Switch contention between MPI and GPFS: MPI_Allreduce performance (as measured from within the code) is degraqded during periods of GPFS read activity from a different parallel code. Several such correlations have been measured.

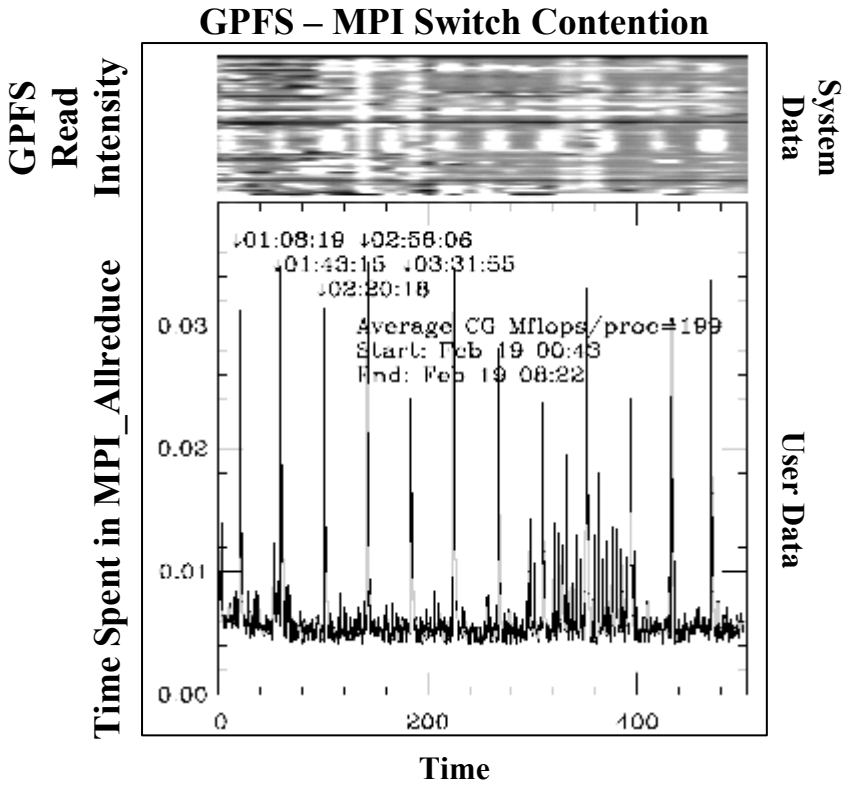


Figure X: The data at the top is GPFS read activity (brighter is more intense). The data at the bottom is performance data as reported from within a 1200 task application code. The time axes for both have been aligned and it is seen that GPFS read activity taking place in another application impacts the performance of mp13's workload at high concurrency. The application responsible for the I/O is a Rmatrix code (m41) running on a separate set of nodes. The impact of switch contention has been measured in other contexts and the impact on performance is generally more pronounced at higher concurrency.

mp13 Allocation and Large Jobs Reimbursement:

	Requested SP Hours	Awarded SP Hours	% Awarded	Real SP Hours Used	Large Jobs Reimbursement (added to award)
FY 2003	1,370,000	1,250,000	91%	2,376330	257,400
FY 2004	3,600,000	2,437,000	68%	3,089,128	100,000

This project had scaled prior to FY 2003 and took advantage of the Reimbursement Programs to better understand the scaling properties of their code. PI Doug Toussaint explained that he would not have done some of this scaling work if the cycles used for development and testing had come out his main allocation.

Conclusions:

Running MILC with 2,000 or more processors becomes difficult not so much because of limitations in the code, but due to architectural limitations imposed by using the same switch fabric for file I/O and MPI. Finding ways to mitigate GPFS-MPI switch contention are challenging but worth pursuing. In the current situation running on 2000 or more tasks does not make best use of the given allocation.

In the case of MILC the reporting of per iteration performance information provides an opportunity to scrutinize in detail the code's parallel efficiency. For most projects and codes performance data is reported as an average across the duration of the batch job and as such the temporal details of contention, if present, would be averaged away.