

From P to Q: The USP-Rice Blue Gene Collaboration

October 17th, 2016

Paul Whitford (pcw2@rice.edu)

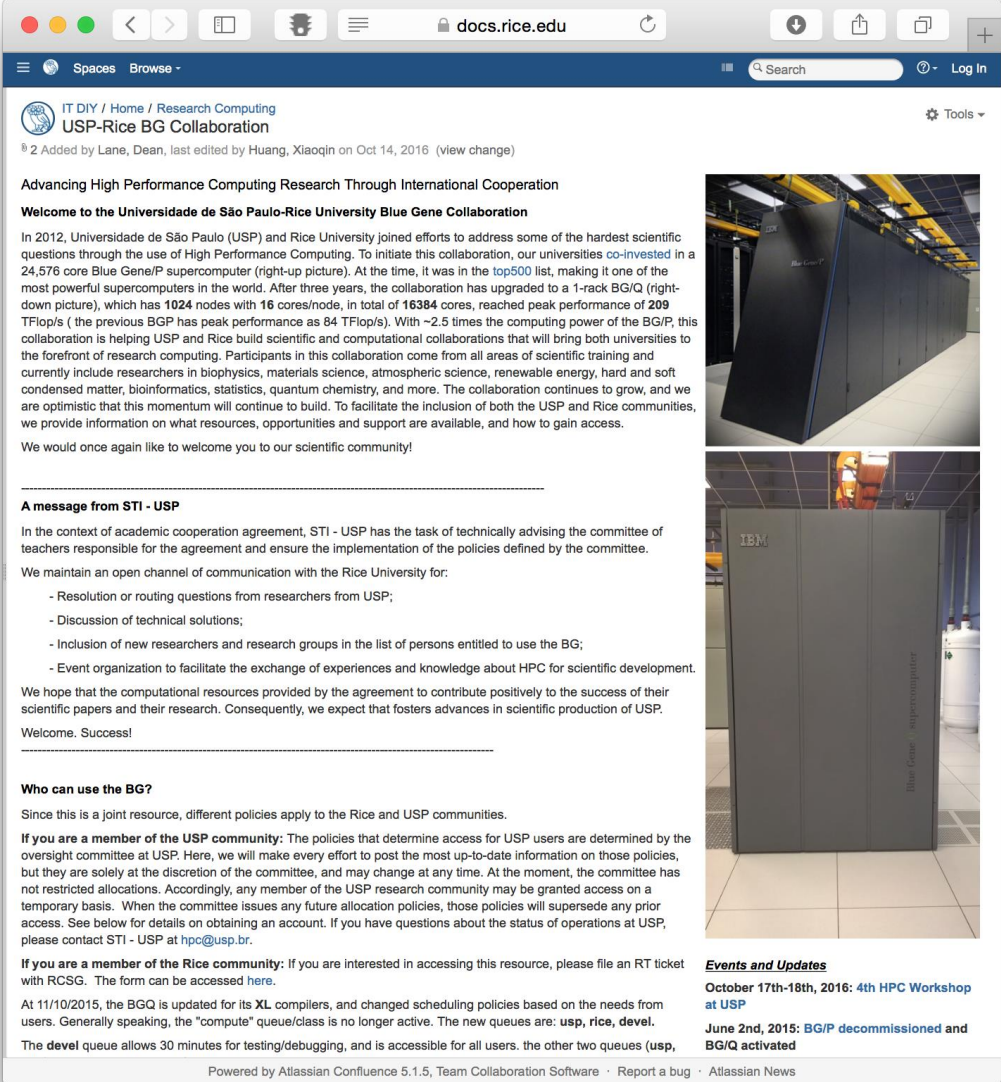
Liaison to USP for HPC
Rice University

Assistant Professor of Physics
Northeastern University

USP-Rice Blue Gene Collaboration Webpage

“One-stop shop”
Information on:

- Getting accounts
 - Logging on
 - Compiling code
 - Running jobs
 - Getting help
 - Updates and activities
 - Transferring data
 - Acknowledgements
-
- All BG/Q information available



The screenshot shows a web browser displaying the 'USP-Rice BG Collaboration' page on the docs.rice.edu Confluence instance. The page header includes navigation links like 'Spaces', 'Browse', and a search bar. The main content area is titled 'USP-Rice BG Collaboration' and includes a welcome message, a message from STI - USP, and information about who can use the BG. On the right side, there are two images of supercomputers: a large blue and black system and a smaller grey system. At the bottom right, there is a section for 'Events and Updates' listing workshops and decommissioning events.

IT DIY / Home / Research Computing
USP-Rice BG Collaboration
2 Added by Lane, Dean, last edited by Huang, Xiaoqin on Oct 14, 2016 (view change)

Advancing High Performance Computing Research Through International Cooperation

Welcome to the Universidade de São Paulo-Rice University Blue Gene Collaboration

In 2012, Universidade de São Paulo (USP) and Rice University joined efforts to address some of the hardest scientific questions through the use of High Performance Computing. To initiate this collaboration, our universities co-invested in a 24,576 core Blue Gene/P supercomputer (right-up picture). At the time, it was in the top500 list, making it one of the most powerful supercomputers in the world. After three years, the collaboration has upgraded to a 1-rack BG/Q (right-down picture), which has 1024 nodes with 16 cores/node, in total of 16384 cores, reached peak performance of 209 TFlop/s (the previous BGP has peak performance as 84 TFlop/s). With ~2.5 times the computing power of the BG/P, this collaboration is helping USP and Rice build scientific and computational collaborations that will bring both universities to the forefront of research computing. Participants in this collaboration come from all areas of scientific training and currently include researchers in biophysics, materials science, atmospheric science, renewable energy, hard and soft condensed matter, bioinformatics, statistics, quantum chemistry, and more. The collaboration continues to grow, and we are optimistic that this momentum will continue to build. To facilitate the inclusion of both the USP and Rice communities, we provide information on what resources, opportunities and support are available, and how to gain access.

We would once again like to welcome you to our scientific community!

A message from STI - USP

In the context of academic cooperation agreement, STI - USP has the task of technically advising the committee of teachers responsible for the agreement and ensure the implementation of the policies defined by the committee.

We maintain an open channel of communication with the Rice University for:

- Resolution or routing questions from researchers from USP;
- Discussion of technical solutions;
- Inclusion of new researchers and research groups in the list of persons entitled to use the BG;
- Event organization to facilitate the exchange of experiences and knowledge about HPC for scientific development.

We hope that the computational resources provided by the agreement to contribute positively to the success of their scientific papers and their research. Consequently, we expect that fosters advances in scientific production of USP.

Welcome. Success!

Who can use the BG?

Since this is a joint resource, different policies apply to the Rice and USP communities.

If you are a member of the USP community: The policies that determine access for USP users are determined by the oversight committee at USP. Here, we will make every effort to post the most up-to-date information on those policies, but they are solely at the discretion of the committee, and may change at any time. At the moment, the committee has not restricted allocations. Accordingly, any member of the USP research community may be granted access on a temporary basis. When the committee issues any future allocation policies, those policies will supersede any prior access. See below for details on obtaining an account. If you have questions about the status of operations at USP, please contact STI - USP at hpc@usp.br.

If you are a member of the Rice community: If you are interested in accessing this resource, please file an RT ticket with RCSG. The form can be accessed [here](#).

At 11/10/2015, the BGQ is updated for its XL compilers, and changed scheduling policies based on the needs from users. Generally speaking, the "compute" queue/class is no longer active. The new queues are: **usp**, **rice**, **devel**. The **devel** queue allows 30 minutes for testing/debugging, and is accessible for all users. the other two queues (**usp**,

Events and Updates

October 17th-18th, 2016: 4th HPC Workshop at USP

June 2nd, 2015: BG/P decommissioned and BG/Q activated

Powered by Atlassian Confluence 5.1.5, Team Collaboration Software · Report a bug · Atlassian News

usp.rice.edu

Discussion Points

- Blue Gene/Q specifications
- Running and optimizing calculations
- Gaining access
- Available support

Where We Started (June 2012)

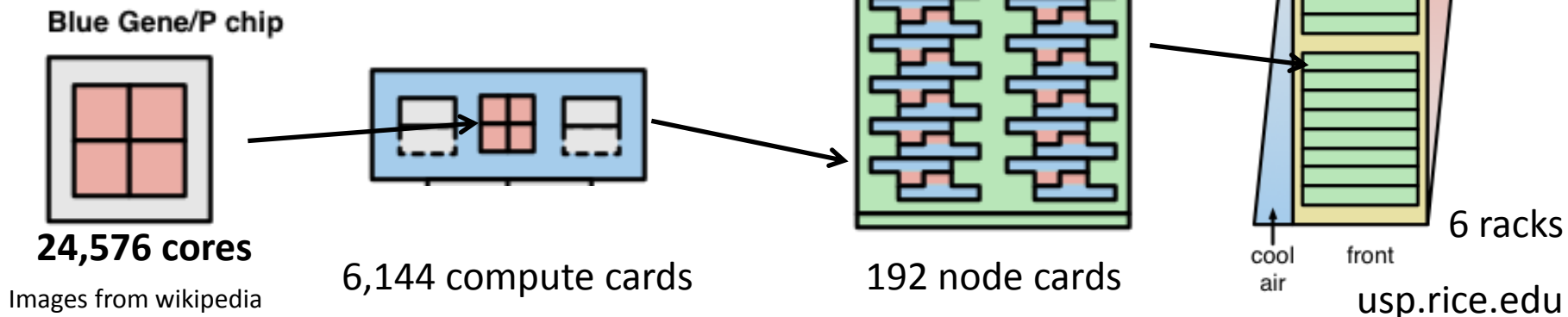


IBM Blue Gene/P

- 377 on the Top500 list (as of June 2012)
- 84 teraflops of performance
- 24,576 cores

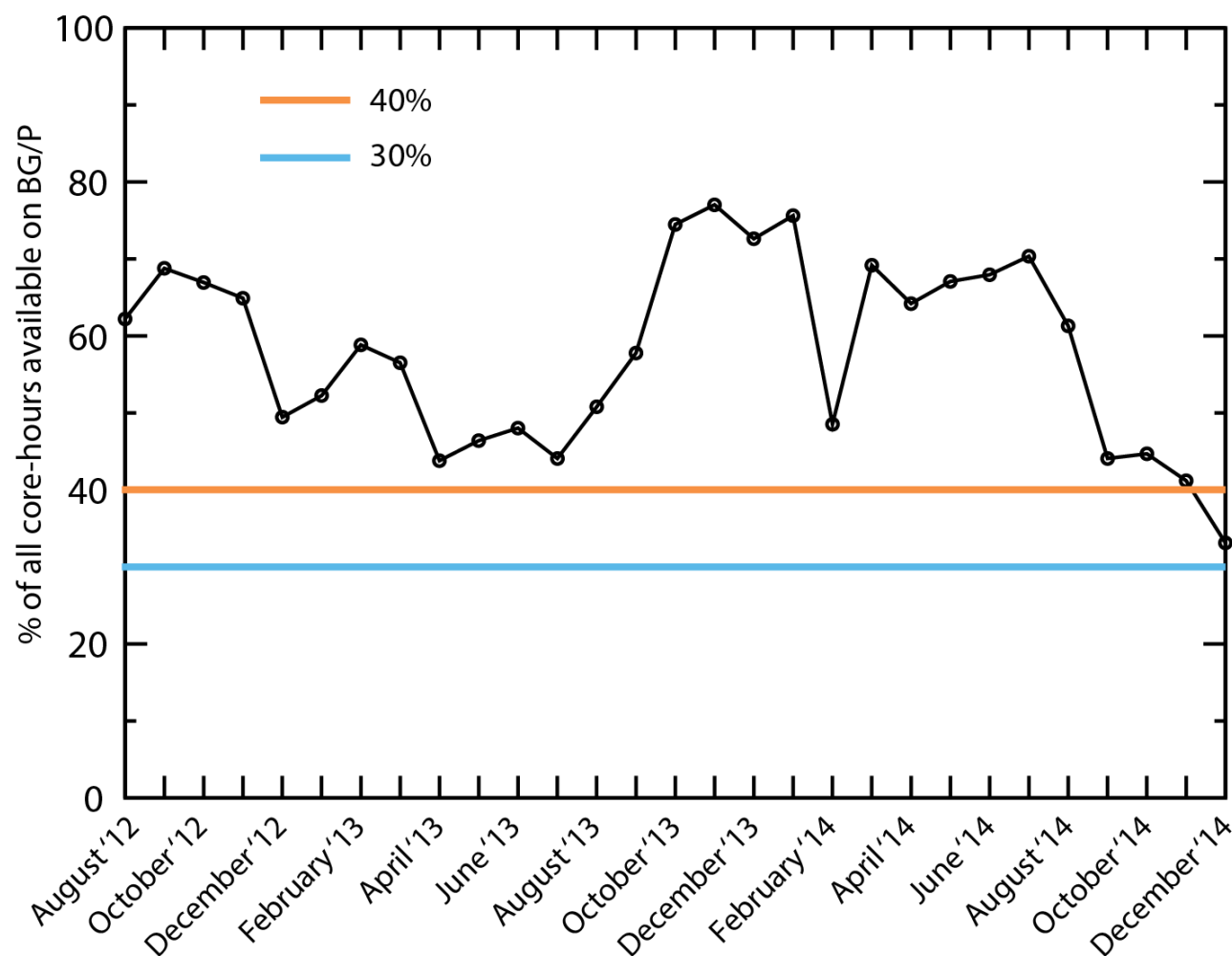
USP-Rice Blue Gene/P (retired)

- Low-power processors, but massively parallel
 - PowerPC 450 processors: 850 MHz
 - High bandwidth and low latency connections
- 30% dedicated for USP



Utilization of BG/P by USP

>100 users from USP, USP São Carlos, USP Ribeirão Preto and FMUSP, across many departments



Many excellent publications have been produced!

The Upgrade (June 2015): BG/Q

- 1-rack Blue Gene/Q
- Qualified for Top500 ranking (~293)
- 40% dedicated to USP usage
- Already installed software
 - Abinit, Amber14, BLACS, BLAS, CP2K, Espresso, FFTW, FHI-aims, FLEX, GAMESS, Gromacs, GSL-GCC, HDF5, Jasper, LAMMPS, LAPACK, MPIblast, NAMD, NETcdf, Numpy, Repast, ScalaPack, Siesta, Vasp, WGS, WPS, WRF
 - Many additional libraries available
 - Additional software is built, upon request

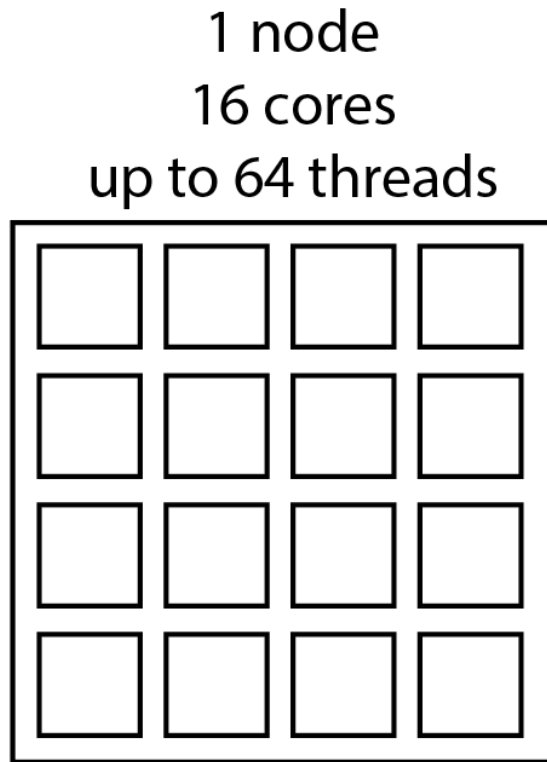


Differences Between BG/P and BG/Q

Attribute	BG/P	BG/Q
Maximum Performance	84 TFLOPS	209 TFLOPS
Power Consumption	371 MFLOPS/W	2.1 GFLOPS/W
Total number of cores	24576	16384
Number of nodes	6144	1024
Cores per node	4	16 + 1 spare + 1 for O/S
Memory per node	4 GB	16 GB
Possible threads per core	1	4
CPU clock-speed	850 MHz	1.6 GHz
Minimum nodes/job	128 (512 physical cores)	32 (512 physical cores)
Scratch disk space	245TB	120TB

4 times the number of cores and memory per node. Multi-level parallelization can lead to large performance increases with BG/Q, and larger-memory applications.

Technical Consideration 1: Distribution of Calculations



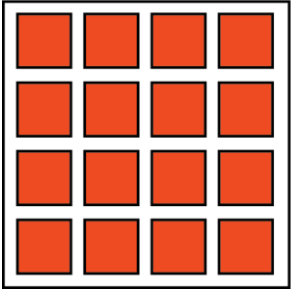
32 node minimum, per job
 $32 \times 16 = 512$ cores
up to 2048 threads

Each job receives integer multiples of 512 cores (32 nodes), regardless of what is requested/utilized.

MPI, or thread-based (openMP) parallelization may be used, as well as multi-level parallelization

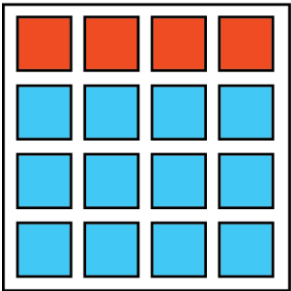
Examples for Utilizing Multi-level Parallelization

1 node
16 mpi ranks per node
1 thread per rank



`/bgsys/drivers/ppcfloor/bin/runjob --np 512 --ranks-per-node=16 -exe ExampleSoftware`

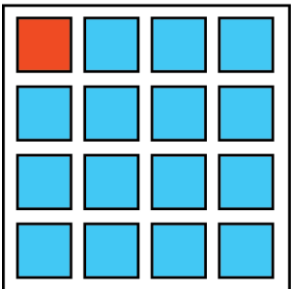
1 node
4 mpi ranks per node
4 threads per rank



`OMP_NUM_THREADS=4` (sets the threads per rank)

`/bgsys/drivers/ppcfloor/bin/runjob -np 128 --ranks-per-node=4 -exe ExampleSoftware`

1 node
1 mpi rank per node
16 threads per rank



`OMP_NUM_THREADS=16`

`/bgsys/drivers/ppcfloor/bin/runjob -np 32 --ranks-per-node=1 -exe ExampleSoftware`

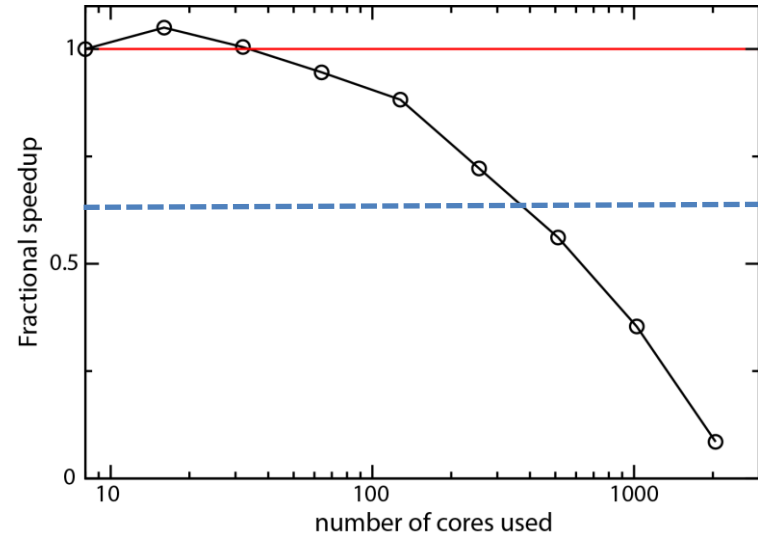
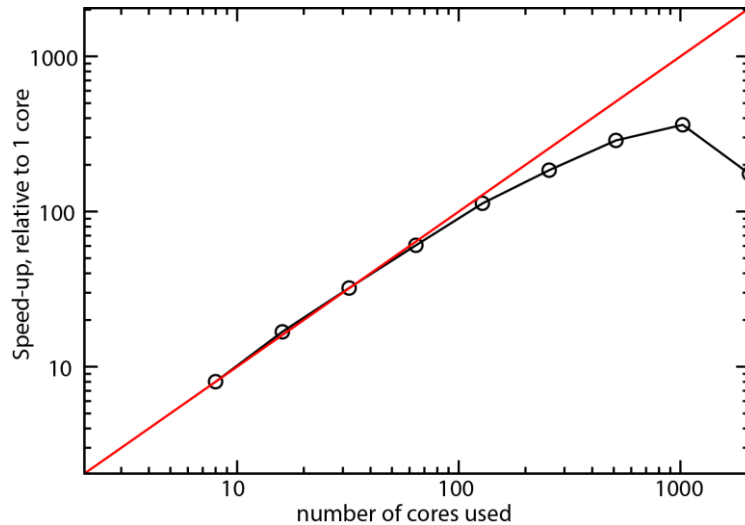
Examples utilize all 512 cores (32 nodes) allocated to the job

Technical Consideration 2: Scalability

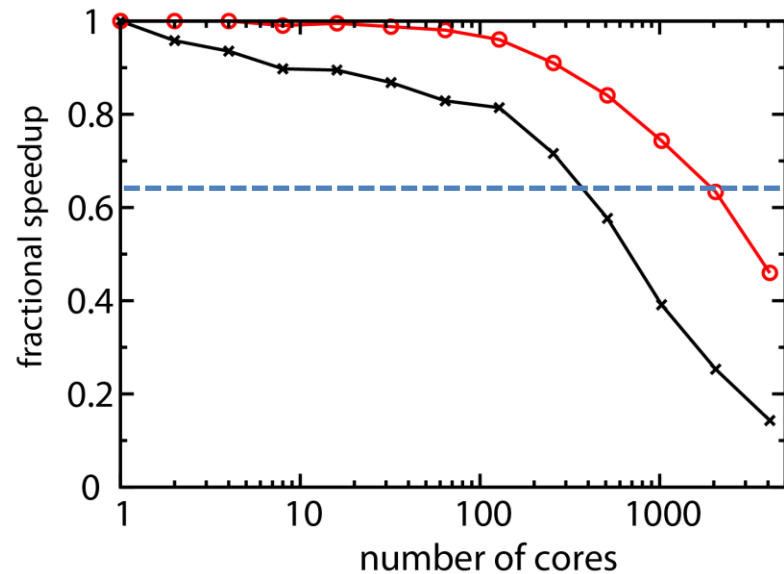
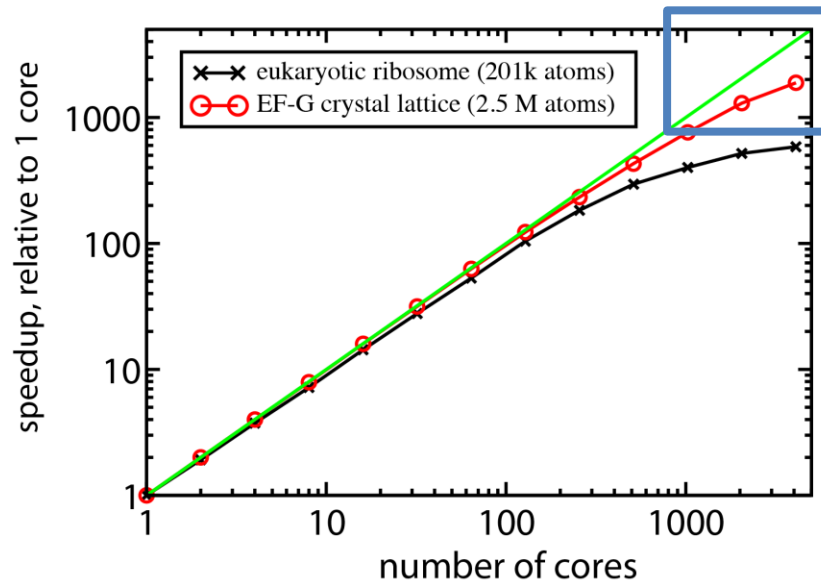
- You should always start with scalability tests.
 - Repeat a fixed-size calculation for a range of number of cores
 - **Frequently reveals computational bottlenecks**
- User feedback is incredibly valuable!
- Please share your performance stats with us
 - Can help users when applying for time on other resources (e.g. XSEDE, INCITE, PRACE)
 - **Will facilitate more efficient calculations**
 - Will help us optimize builds

Examples of Scaling Data

100k atom explicit-solvent simulation using NAMD on BG/P (Performed by Filipe Lima)



200k-2.5M atom simulations using an implicit-solvent model in Gromacs on Stampede



Many Considerations Impact Scalability

- Is your specific calculation well suited for parallelization?
 - For example, molecular dynamics simulations for large systems tend to scale to higher numbers of cores (weak vs. strong scaling)
- Have you enabled all appropriate flags at runtime?
 - Software packages often guess... you need to test various settings
 - For example, in MD simulations, you must tell the code how to distribute atoms across nodes
- Is MPI, openMP, hybrid MPI+openMP enabled in the code?
 - MPI launched many processes, each with its own memory and instructions
 - openMP launches threads that share memory
 - hybrid: each MPI process has its own memory, which is then shared with its own openMP threads
- Are you trying the most appropriate combination of calculation, parallelization and flags *for the machine you are using*?
- Does your calculation take long enough to complete that scalability is noticeable?
 - Sometimes, initializing parallelization may take minutes, or longer.
- Is the performance reproducible?
- How well is the code written?

With so many factors impacting performance, it is essential that you perform scaling tests before doing production calculations

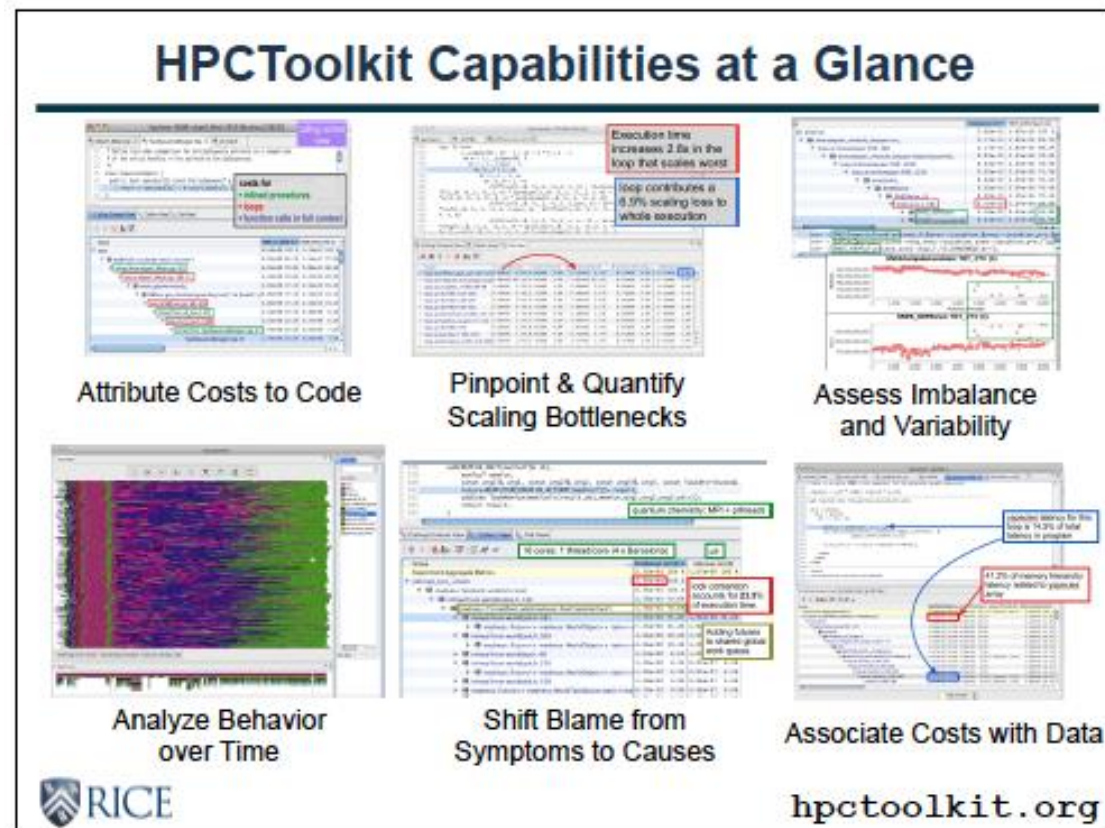
For when the code needs improvement...

High-performance tuning with HPC toolkit

HPC Toolkit developed at Rice (Mellor-Crummey Group)

HPC Toolkit functionality

- Accurate performance measurement
- Effective performance analysis
 - Pinpointing scalability bottlenecks
 - Scalability bottlenecks on large-scale parallel systems
- Scaling on multicore processors
- Assessing process variability
- Understanding temporal behavior



Experts are available to help you profile your code, even if you think it works well!

Technical Consideration 3: Cross Compilation

- Frontend
 - Linux
- Backend
 - 64-bit PowerPC A2 processor core
 - Minimal OS, lacking many system calls
- Code is compiled on frontend and executed on the backend.
- Can be very difficult for users new to BG
 - *There is support staff to help with compiling*



Xiaoqin Huang

Getting Help, Organizing Collaborative Efforts

Center for Research Computing (CRC)



World-class support and service for research computing



A thriving community of HPC scientists sharing research and building collaborations with a significant presence in national cyberinfrastructure issues.



Gaining Access to BG/Q

- USP users
 - First, email request to USP-STI (hpc@usp.br), requesting approval for an account
 - Subsequent account handling is managed by Paul Whitford (pcw2@rice.edu)
 - Technical support available
 - Full details available at usp.rice.edu



Any Questions?

Feel free to contact all of us. We are here to help!

STI hpc@usp.br

Paul Whitford pcw2@rice.edu

Xiaoqin Huang xiaoqin.huang@rice.edu