第１２回先進スーパーコンピューティング環境研究会（ASE 研究会）実施報告

東京大学情報基盤センター　准教授　片桐孝洋

4 月 25 日(水)13 時 30 分～18 時 10 分、東京大学情報基盤センター４階遠隔講義室にて、第 12 回先進スーパーコンピューティング環境研究会（ASE 研究会）が開催されました。

国内の大学・研究機関、および企業からの参加者は 17 名でした。活発な議論がなされました。

今回は、招待講演として Lawrence Berkeley National Laboratory から Osni Marques 博士をお呼びし、アプリケーション開発環境の現在と今後に関する講演を行いました。また、国内の招待講演として、東京大学の相島健助 助教をお呼びし特異値計算における dqds 法の講演、山梨大学の鈴木智博 准教授による近年の計算機における QR 分解の進展に関する講演、および、筑波大学の山本和磨氏による非線形固有値問題の並列アルゴリズムに関する講演を行いました。

Osni Marques 博士の講演は、エクサスケールを目指すアプリケーション開発に関するものです。現在米国では、1PFLOPS で実践されているが、今後の環境として 10PFLOPS への段階、さらにその先には 100PFLOPS への段階で、大きな壁が、プログラミングの観点から予想されています。アプリケーションソフトウェアがエクサスケールへ到達するには、まだまだ難問が山積しています。そこで、数値計算ライブラリを含む、適切なツール群を利用することで、その壁を１つにすることが重要であると考えています。特に Marques 博士は、米国エネルギー省(DOE)により開発された数値計算ライブラリやシミュレーションのためのツール群をまとめ、保守・管理・運営していくプロジェクトである ACTS Collection プロジェクトの代表です。長年、当該分野に貢献されてきました。現在日本でも、エクサスケールに資するソフトウェアの開発を、国際協調して開発していくことが重要性と認識されています。既存開発物の活用と、不足機能については、国際的に協調して共同開発していくことが求められています。スパコンセンターとしては、実際のスパコンにそれらをインストールした上で、保守・管理・普及していく必要があります。ユーザが行う計算科学の活動の支援をしていくことが重要な職務になっています。また、100PFLOPS を超える環境で必要となる数値計算ライブラリの技術的事項に関するワークショップが DOE 主導のもと、すでに行われていることも興味深く拝聴しました。

Marques 博士の当日発表に関し、本原稿の後に発表資料を掲載します。ご興味のある方はご覧ください。当日のプログラムを以下に載せます。

- **Program**
  - 13:30 - 14:15 Invited Speaker Dr. Kensuke Aishima (University of Tokyo, Japan), Dr. Yuji Nakatsukasa (University of Manchester, UK), Dr. Ichitaro Yamazaki (University of Tennessee, USA)

- ✧ Title: dqds with aggressive deflation for singular values
- ✧ Abstract: Matrix singular values play an important role in many applications. Accordingly, numerical methods for computing singular values are of great importance in practice. In order to compute the singular values, the given matrix is first transformed to a bidiagonal matrix with suitable orthogonal transformations, and then a certain iterative method is applied to the bidiagonal matrix. In 1994, Fernando and Parlett discovered the differential quotient difference with shifts (dqds) algorithm for computing singular values of bidiagonal matrices to high relative accuracy. The dqds algorithm is currently implemented in LAPACK as the DLASQ routine. Our objective is to reduce the dqds runtime without loss of high relative accuracy. More specifically, we incorporate into the dqds a technique called aggressive deflation, which has been applied successfully to the Hessenberg QR algorithm. We propose an efficient and stable implementation by taking advantage of the bidiagonal structure. Numerical results are also shown to illustrate that our aggressive deflation strategy often reduces the dqds runtime significantly. In addition, a shift-free version of our algorithm has a potential to be parallelized in a pipelined fashion. Our mixed forward-backward stability analysis proves that with our proposed deflation strategy, all the singular values are computed to high relative accuracy.

- ➢ 14:25 - 15:10 Invited Speaker Associate Professor Tomohiro Suzuki (Yamanashi University, Japan)
  - ✧ Title: On implementations of tile QR factorization algorithm for recent hardware
  - ✧ Abstract: There are many important matrix factorizations in dense linear algebra. Classic implementations suffer from performance limitations due to the use of L2 and L1 BLAS operations. The scalability limitation exists even in a blocking algorithms which are rich in L3 BLAS. Such limitations are called fork-join bottlenecks. In order to take advantage of the architectural features on recent multi-core or many-core systems, tile algorithms for the matrix factorization are

proposed. In this talk, we present our implementations of the tile QR factorization algorithm for the GPU system and the multi-core CPU cluster system. It is implemented with OpenMP and MPI hybrid programing model for the multi-core cluster system i.e. T2K open supercomputer (U.Tokyo). For the GPU system, we also show the implementation for the multi GPU environment. In order to achieve high performance, it is important to tune each sub program (kernel) of the tile algorithm. In addition to that, a proper scheduling with checking dependencies among all kernels has an equivalent importance. Some studies for an optimized scheduling for the tile QR algorithm are reported.

- 15:20 - 16:20 Invited Speaker Dr. Osni Marques (Lawrence Berkeley National Laboratory, USA)
  - Title: Dealing with Application Development -- Now and Henceforth
  - Abstract: The development of simulation codes is often a costly process that results from the combination of the increasing complex problems to be solved and the evolution of computer architectures. Practitioners are expected to develop highly efficient codes, although emerging computer architectures pose formidable challenges in achieving adequate levels of performance. Code developers usually have a range of choices for programming? MPI, OpenMP, PGAS Languages, CUDA, and the emerging OpenACC? but whose benefits / advantages may not be clear. To easy the development process, scientific software libraries are increasingly used in simulation codes: in many cases, this approach has lessened the development effort, contributed to an optimal usage of the available computational resources, and lessened issues related to portability and application lifecycle. However, how will advances in programming and hardware impact libraries? This presentation will discuss some of these issues.
- 16:30 - 17:15 Invited Speaker Mr. Kazuma Yamamoto, Mr. Yasuyuki Maeda, Mr. Yasunori Futamura, Professor Tetsuya Sakurai (Department of Computer Science, University of Tsukuba, Japan)
  - Title: Adaptive parallel algorithm for stochastic estimation of

nonlinear eigenvalue density

- ✧ Abstract: A numerical method that estimates the eigenvalue density of nonlinear eigenvalue problems in the specified region has been proposed. Nonlinear eigenvalue problems arise in science and engineering. Since parameter settings for eigensolver that based on eigenvalues are required, accuracy and parallel efficiency can be improved by using eigenvalue density. In this presentation, we propose an algorithm for efficient execution of the estimation method on parallel computers. Conventional approach requires the solutions of linear systems for each integral point that uniformly distributed on the complex plane. Thus, it causes the load imbalance and requires a large computational cost due to the variation of solution time for linear systems. The proposed master-worker type adaptive algorithm improves the load balance and reduces the computational cost by the placing integral points according to the density of eigenvalue in the specified region. Moreover, we propose a look-ahead algorithm that balances the loads more efficiently by recycling the variables in the linear solver. We evaluate the efficiency of the proposed algorithms by several numerical examples.

- ➢ 17:25 - 18:10 [Regular Presentations] Satoshi Itoh (Information Technology Center, The University of Tokyo, Japan)
  - ✧ Title: Study of plugging-in AT mechanism in OpenFOAM
  - ✧ Abstract: OpenFOAM is an open source CFD software package. It is free software and developers can describe the governing equations simply with its instinctive interface, it is spread widely. OpenFOAM is based on the finite volume method (FVM), so that the main application is CFD. However, it has a problem that it is difficult to achieve high performance on high-end machine such as supercomputers. We are developing ppOpen-AT, which is an infrastructure of auto-tuning (AT) for ppOpen-HPC. ppOpen-HPC is a numerical middleware for post Petascale era. One of its features is auto-tuning mechanism (ppOpen-AT). We chose OpenFOAM as one of testing software. In this study, we optimize OpenFOAM manually for the first step of auto-tuning. We show

numerical results on T2K, and discuss the AT methodology for OpenFOAM.

- 18:10: Closing Remarks Takahiro Katagiri (The University of Tokyo)
- 18:40 - A Banquet near Nedu station

ASE研究会の開催情報はメーリングリストで発信をしております。研究会メーリングリストに参加ご希望の方は、ASE研究会幹事の片桐（katagiri@cc.u-tokyo.ac.jp）までお知らせください。

以上

University of Tokyo
April 25, 2012

Dealing with Application Development – Now and Henceforth –

Osni Marques
Lawrence Berkeley National Laboratory
OAMarques@lbl.gov

(under the auspices of the Japan Society for the Promotion of Science)

Contents

- Applications and the software stack
- The DOE ACTS Collection
- Technology transition
- Impact of hardware evolution on libraries

Office of Science
U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

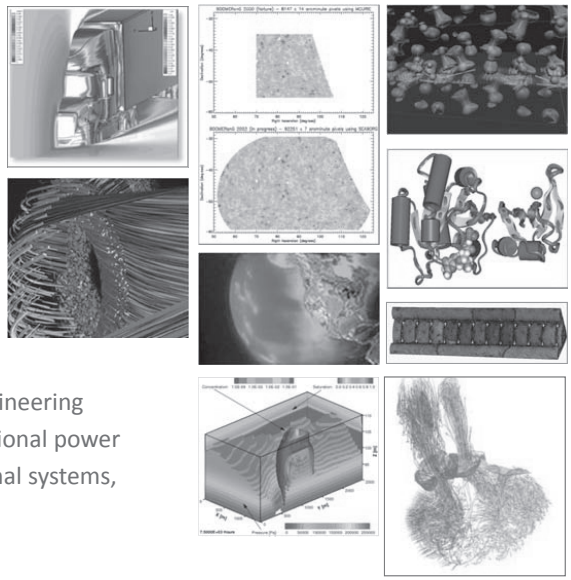
HPC Applications

- Accelerator Science
- Earth Sciences
- Material Sciences
- Biology
- Chemistry
- Astrophysics

⋮

Commonalities

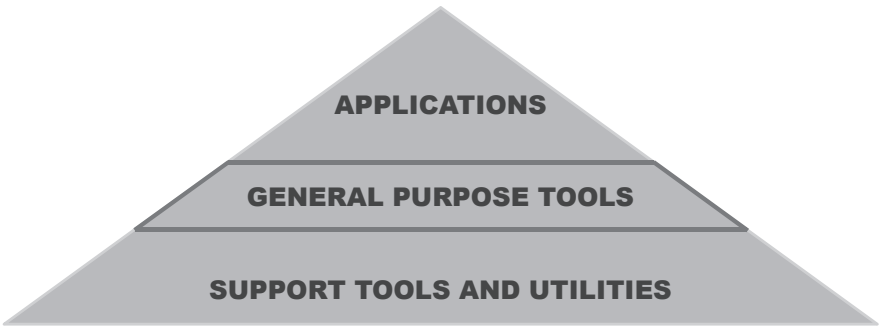
- Advancements in science and engineering
- Increasing demands for computational power
- Reliance on available computational systems, languages, and software tools



Office of Science
U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

Software Stack



HARDWARE

Leading technology paths (swim lanes):

- **Multicore: maintain complex cores, and replicate (x86 and Power7, Blue Waters, NGSC)**
- **Manycore/embedded: use many simpler, low power cores from embedded systems (BlueGene, Dawning)**
- **GPU/Accelerator: use highly specialized processors from gaming/graphics market space (NVidia Tesla, Cell, Intel Knights Corner/MIC)**

Risks in swim lane selection:

- **Select too soon: users cannot follow**
- **Select too late: fall behind performance curve**
- **Select incorrectly: subject users to multiple disruptive technology changes**

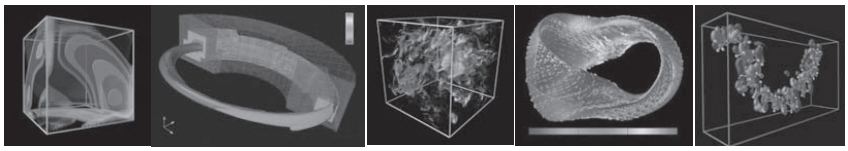
Office of Science
U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

The DOE ACTS Collection

- **Goal:** The Advanced CompuTational Software Collection (ACTS) makes reliable and efficient software tools more widely used, and more effective in solving the nation's engineering and scientific problems

- Long term maintenance
- Independent test and evaluation
- Outreach and dissemination
- High level user support



Office of Science U.S. DEPARTMENT OF ENERGY

5

BERKELEY LAB

The DOE ACTS Collection: Current Functionalities

Category	Tool	Functionalities
Numerical	Trilinos	Linear Solvers (Iterative Methods): e sparse linear systems
	Hypre	• Eigenvalues • Singular Value Decomposition • Linear Least Squares e sparse linear systems (grid-centric
	PETSc	and nonlinear systems of equations)
	SUNDIALS	ased) • Nonlinear optimization • Newton-based • CG • Direct search erential equations, nonlinear algebraic
	ScaLAPACK	
	SLEPc	
	SuperLU	• ODEs rge, sparse, nonsymmetric linear
	TAO	• distributed dense arrays that can be accessed through a shared memory-like style unconstrained
Code Development	Global Arrays	• solution of PDEs on a complex geometry • moving geometry • overlapping grid al Address puters
	Overture	ies
Code Execution	TAU	
Library Development	ATLAS	Tools for the automatic generation of optimized numerical software (dense linear algebra)

Office of Science U.S. DEPARTMENT OF ENERGY

6

BERKELEY LAB

Software Interfaces

```
CALL BLACS_GET( -1, 0, ICTXT )
CALL BLACS_GRIDINIT( ICTXT, 'Row-major', NPROW, NPCOL )
  :
CALL BLACS_GRIDINFO( ICTXT, NPROW, NPCOL, MYROW, MYCOL )
  :
CALL PDGESV( N, NRHS, A, IA, JA, DESCA, IPIV, B, IB, JB, DESCB, INFO )
```

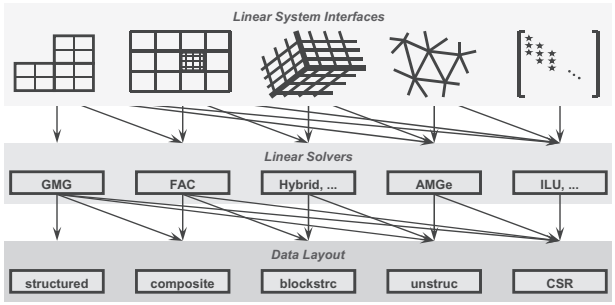
function call
(ScaLAPACK)

- **-ksp_type [cg,gmres,bcgs,tfqmr,...]**
- **-pc_type [lu,ilu,jacobi,sor,asm,...]**

command line
(PETSc)

More advanced:

- **-ksp_max_it <max_iters>**
- **-ksp_gmres_restart <restart>**
- **-pc_asm_overlap <overlap>**
- **-pc_asm_type [basic,restrict,interpolate,none]**



problem domain
(Hypre)

Office of Science U.S. DEPARTMENT OF ENERGY

Addressing Application Performance Issues

- How does performance vary with different compilers?
- Is poor performance correlated with certain OS features?
- Has a recent change caused unanticipated performance?
- How does performance vary with MPI variants?
- Why is one application version faster than another?
- What is the reason for the observed scaling behavior?
- Did two runs exhibit similar performance?
- How are performance data related to application events?
- Which machines will run my code the fastest and why?
- Which benchmarks predict my code performance best?

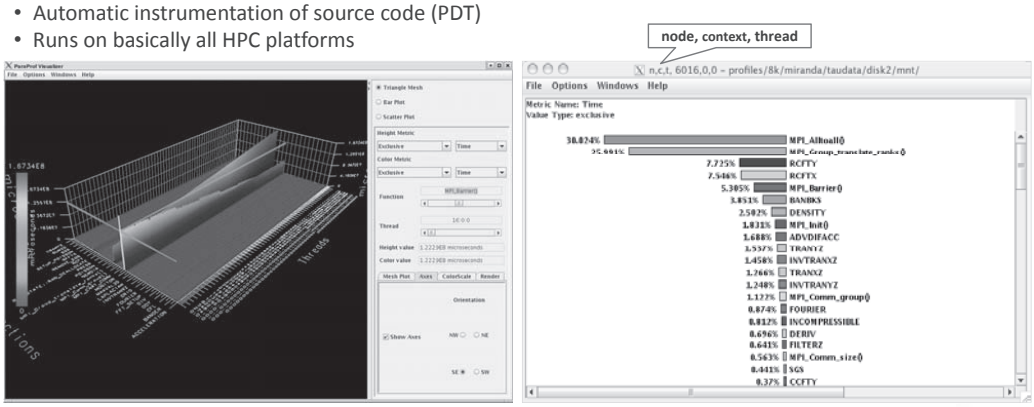
⋮

(*courtesy of Sameer Shende*)

Office of Science U.S. DEPARTMENT OF ENERGY

TAU: Performance Analysis

- **Profiling**: summary statistics of performance metrics (# of times a routine was invoked exclusive or inclusive time or hardware counts, calltrees and callgraphs, memory and message sizes etc)
- **Tracing**: when and where events took place along a global timeline (timestamped log of events, message communication events)
- Automatic instrumentation of source code (PDT)
- Runs on basically all HPC platforms
- 3D profile browser (paraprof)
- To use TAU, one only needs to set a couple of environment variables and substitute the name of the compiler with a TAU shell script
- Ex. Flat profile of Miranda (LLNL; hydrodynamics / Fortran + MPI) on an BG/L;



(See http://tau.uoregon.edu/tau.ppt)

Using TAU (basic level)

- TAU supports several measurement options (profiling, tracing, profiling with hardware counters etc)
- Each measurement configuration of TAU corresponds to a unique stub makefile and library that is generated when you configure it
- To instrument source code using PDT, choose an appropriate TAU stub makefile in <arch>/lib:

  % setenv TAU_MAKEFILE /usr/local/packages/tau/i386_linux/lib/Makefile.tau-mpi-pdt
  % setenv TAU_OPTIONS '-optVerbose ...' **(see "tau_compiler.sh –help")**
- Use tau_f90.sh, tau_cxx.sh or tau_cc.sh as Fortran, C++ or C compilers:

  % tau_f90.sh foo.f90 **(instead of "mpif90 foo.f90")**
- Execute application and analyze performance data:

  % pprof **(for text based profile display)**
  % paraprof **(for GUI)**

Software Tools for Application Development, Portability and Performance

- **min**[*time_to_first_solution*] (prototype)
- **min**[*time_to_solution*] (production)
- **min**[*software-development-cost*]
- **max**[*software_life*]
- **max**[*resource_utilization*]

- Outlive Complexity
  - Increasingly sophisticated models
  - Model coupling
  - Interdisciplinary

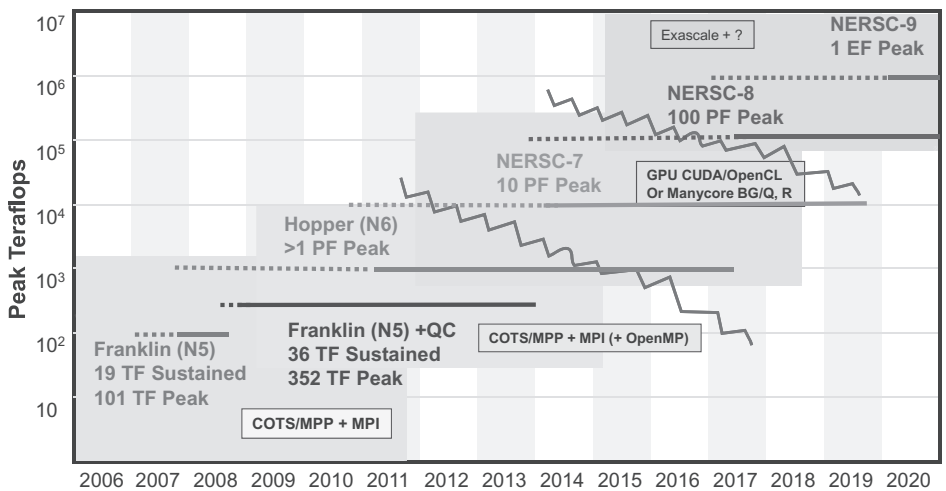
  } *software evolution*
- Sustained Performance
  - Increasingly complex algorithms
  - Increasingly diverse architectures
  - Increasingly demanding applications

  } *long-term deliverables*

Office of Science U.S. DEPARTMENT OF ENERGY

Technology Transition

… and impacts to a facility like NERSC



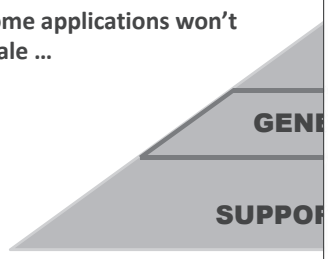
Source: Horst Simon & Kathy Yelick

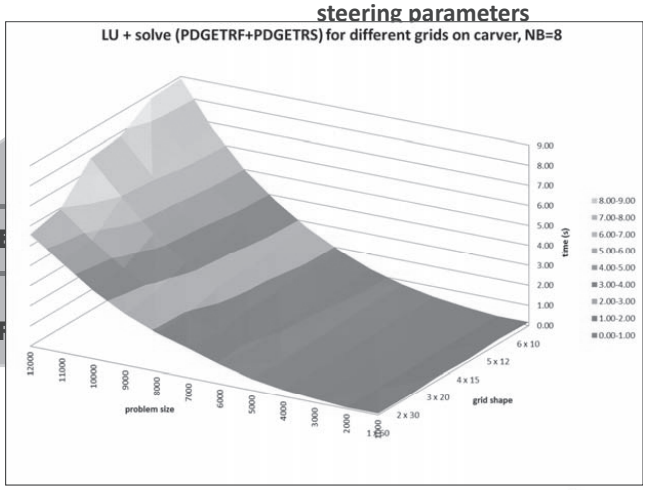
Office of Science U.S. DEPARTMENT OF ENERGY

Parametric Research and Integration

- **Hand-tuning algorithmic parameters can be cumbersome**
- **Auto-tuning produces a single tuned library (max cores per node)**
- **Some applications won't scale …**

- **Auto-tuned algorithmic parameters**
- **Auto-tuned libraries through steering parameters**





Office of Science U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

Hardware and Programming Impacts on Libraries

- *Workshop on Exascale Programming Challenges*, Marina del Rey, CA, July 27-29, 2011
- *Workshop on Extreme-Scale Solvers: Transition to Future Architectures*, Washington, DC, March 8-9, 2012

- Focus on solvers…

Office of Science U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

Current PF Machines

(Michael Feldman, http://www.hpcwire.com/hpcwire/2012-04-10/the_processors_of_petascale.html)

System	Country	Processor	Interconnect	Petaflops
K Computer	Japan	Fujitsu SPARC64	Tofu	11.28
Tianhe-1A	China	Intel x86/NVIDIA GPU	Galaxy	4.70
Nebulae	China	Intel x86/NVIDIA GPU	InfiniBand	2.98
Jaguar	US	AMD x86	Gemini	2.33
TSUBAME 2	Japan	Intel x86/NVIDIA GPU	InfiniBand	2.29
CURIE	France	Intel x86/NVIDIA	InfiniBand	2.00
Helios	Japan	Intel x86	InfiniBand	1.50
Roadrunner	US	AMD x86/PowerXCell	InfiniBand	1.37
Lomonosov	Russia	Intel x86/NVIDIA GPU	InfiniBand	1.37
Cielo	US	AMD x86	Gemini	1.36
Tianhe-1A Hunan	China	Intel x86/NVIDIA GPU	Galaxy	1.34
Pleiades	US	Intel x86	InfiniBand	1.32
Hopper	US	AMD x86	Gemini	1.29
Tera-100	France	Intel x86	InfiniBand	1.25
Kraken	US	AMD x86	SeaStar	1.17
Oakleaf-FX	Japan	Fujitsu SPARC64	Tofu	1.13
Sunway Blue Light	China	ShenWei SW1600	InfiniBand	1.07
HERMIT	Germany	AMD x86	Gemini	1.04
Mole 8.5	China	Intel x86/NVIDIA GPU	InfiniBand	1.01
JUGENE	Germany	PowerPC 450	Custom	1.00

Office of Science U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

Challenges for Next Generation Solvers (100PF – and beyond)

- Extreme levels of concurrency
  - millions of nodes with thousands of lightweight cores
  - hundreds of thousands of nodes with more aggressive cores
- Resilience and non-deterministic behavior
  - hard interrupts (failure of a device)
  - soft errors (change of a data value due to faults in logic latches)
- Reduced memory sizes per core
  - more computation on local data, minimization of synchronization
  - shift the focus from the usual weak scaling to strong scaling

Office of Science U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

Challenges for Next Generation Solvers (100PF – and beyond)

- Data storage and movement
  - on a node, data movement will be much more costly, than other operations
  - data access will be much more sensitive to data layout
- Deep memory hierarchies
  - solvers may need to be hierarchical (e.g. cache-oblivious)
- Portability with performance
  - current programming possibilities are not interoperable
  - abstractions

Office of Science U.S. DEPARTMENT OF ENERGY

17

Next Generation Solvers Features

- Communication/synchronization hiding algorithms
  - e.g. dot products interposed with matrix-vector multiplies
- Communication/synchronization reducing algorithms
  - e.g. s-step Krylov methods
- Mixed-precision-arithmetic algorithms
  - e.g. LU + triangular solves in s.p., iterative refinement in d.p.
  - reduction of memory usage
- Fault-tolerant and resilient algorithms
  - localized checkpoints and asynchronous recovery
  - checksum

Office of Science U.S. DEPARTMENT OF ENERGY

18

Next Generation Solvers Features

- Energy-efficient algorithms
- Stochastic algorithms
  - nondeterminism in data and operations at very large scales
- Algorithms with reproducibility
  - bit-wise identical results from one run to another may be too costly

Office of Science U.S. DEPARTMENT OF ENERGY

19

Transition to New Solvers

- Evolutionary algorithmic research
  - development and optimization on existing (heterogeneous) petascale architectures
- Transition to new application-library interfaces
  - departure from an MPI-only programming model
- Research community interaction (longer term)
  - complexity of issues that need to be addressed
- Revolutionary algorithmic research
  - rethink the solver process (algorithmic and programming approaches)
  - revisit algorithms that may not perform well on current systems
  - co-development of ideas in the computational science community

Office of Science U.S. DEPARTMENT OF ENERGY

20

The DOE ACTS Collection: Current Functionalities

Category	Tool	Functionalities
Numerical	Trilinos	Algorithms for the iterative solution of large sparse linear systems
	Hypre	Algorithms for the iterative solution of large sparse linear systems (grid-centric interfaces)
	PETSc	Tools for the solution of PDEs (sparse linear and nonlinear systems of equations)
	SUNDIALS	Solvers for the solution of systems of ordinary differential equations, nonlinear algebraic equations, and differential-algebraic equations
	ScaLAPACK	High performance dense linear algebra routines
	SLEPc	Eigensolver package built on top of PETSc
	SuperLU	General-purpose library for the direct solution of large, sparse, nonsymmetric linear systems
	TAO	Tools for the solution of optimization problems (nonlinear least squares, unconstrained minimization, bound constrained optimization)
Code Development	Global Arrays	Library that enables a shared-memory view for distributed-memory computers
	Overture	Framework for solving partial differential equations in complex geometries
Code Execution	TAU	Set of tools for analyzing the performance of multi-language programs
Library Development	ATLAS	Tools for the automatic generation of optimized numerical software (dense linear algebra)

http://nkl.cc.u-tokyo.ac.jp/VECPAR2012
http://acts.nersc.gov/events/Workshop2012

Office of Science
U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

Thank you!