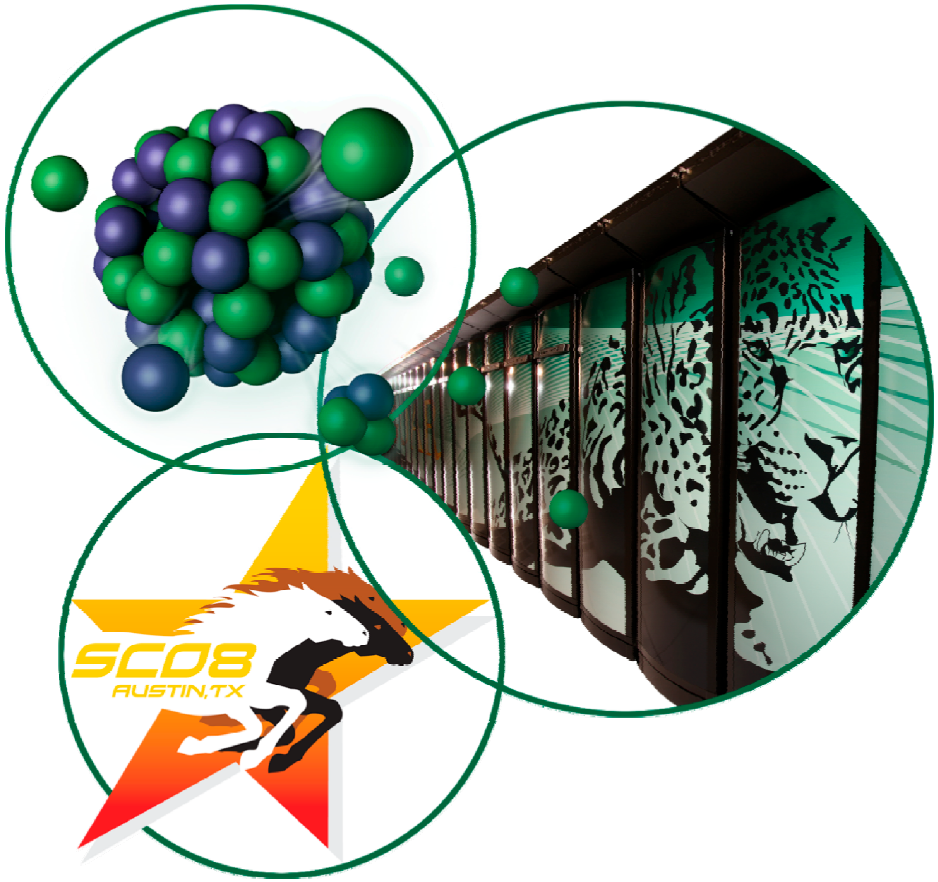
Performance Engineering Research Institute (PERI)



Presented by

Patrick H. Worley

Computational Earth Sciences Group
Computer Science and Mathematics Division

Managed by UT-Battelle
for the Department of Energy

SciDAC
Scientific Discovery through Advanced Computing

Oak Ridge National Laboratory

Performance engineering: enabling petascale science

Petascale computing is about delivering performance to scientists

Maximizing performance is getting harder

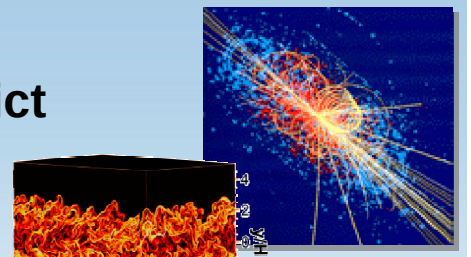
- Systems are more complicated
  - O (100 K) processors
  - Multicore with SIMD extensions
- Scientific software is more complicated
  - Multidisciplinary and multiscale

IBM BlueGene at LLNL

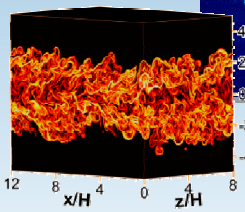
Cray XT4 at ORNL

PERI addresses this challenge in three ways

- Model and predict application performance
- Assist SciDAC scientific code projects with performance analysis and tuning
- Investigate novel strategies for automatic performance tuning



BeamBeam3D accelerator modeling



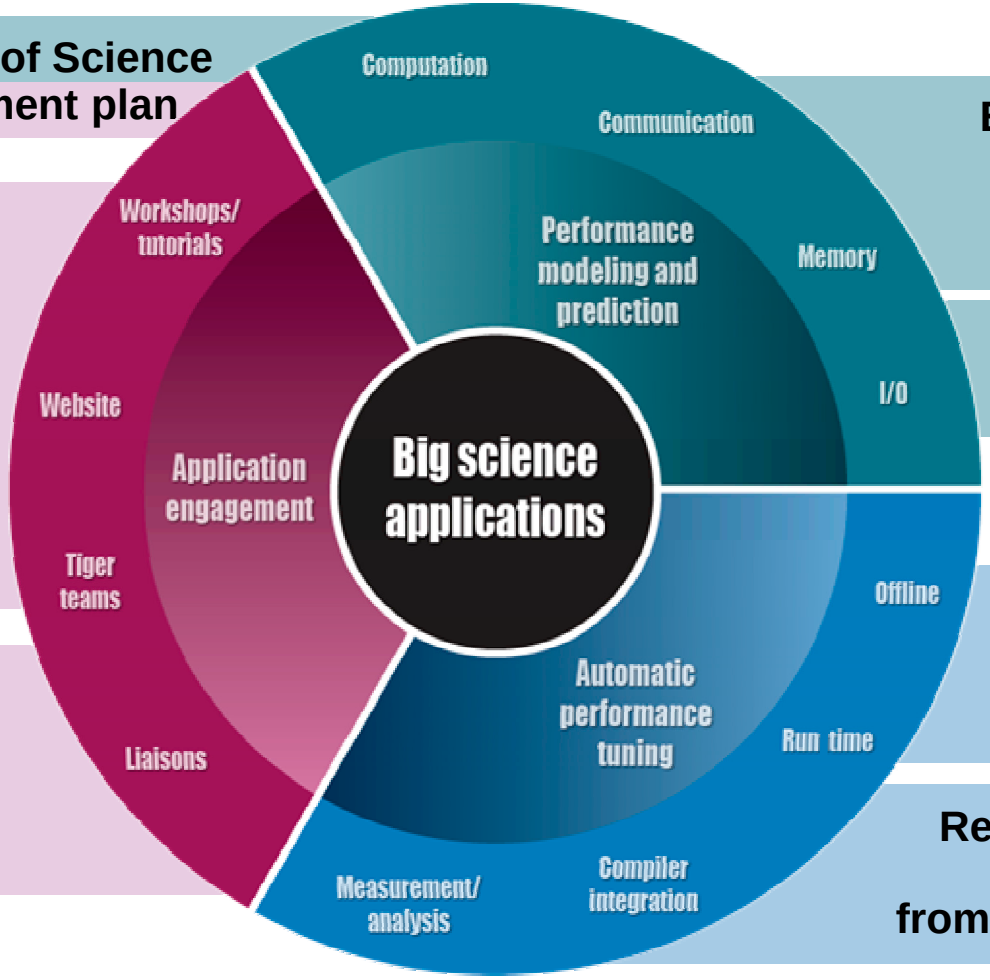
S3D turbulent combustion modeling

SciDAC-2
Performance Engineering Research Institute (PERI)

Informing Office of Science 10-year procurement plan

Providing near-term impact on the performance optimization of SciDAC applications

Informing long-term automated tuning efforts



Evaluating architectures and algorithms in preparation for move to petascale

Providing guidance in automatic tuning

Long-term research goal to improve performance portability

Relieving the performance optimization burden from scientific programmers

Engaging SciDAC software developers

Application survey

- Collect data on SciDAC-2 and INCITE code characteristics and performance requirements
- Use data to determine PERI engagement activities and to direct PERI research

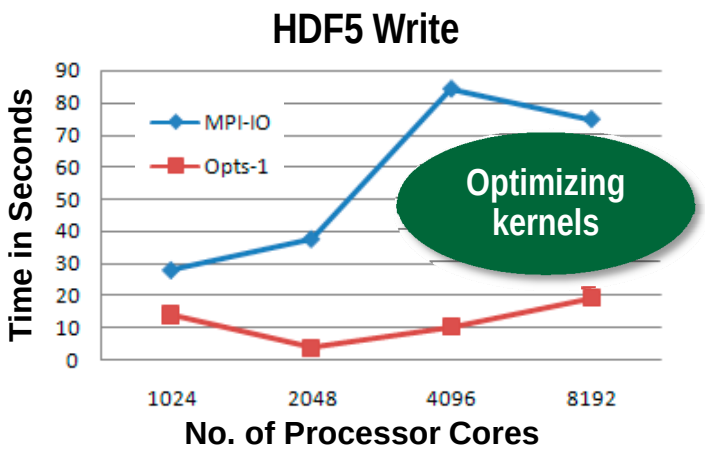
http://icl.cs.utk.edu/peri/

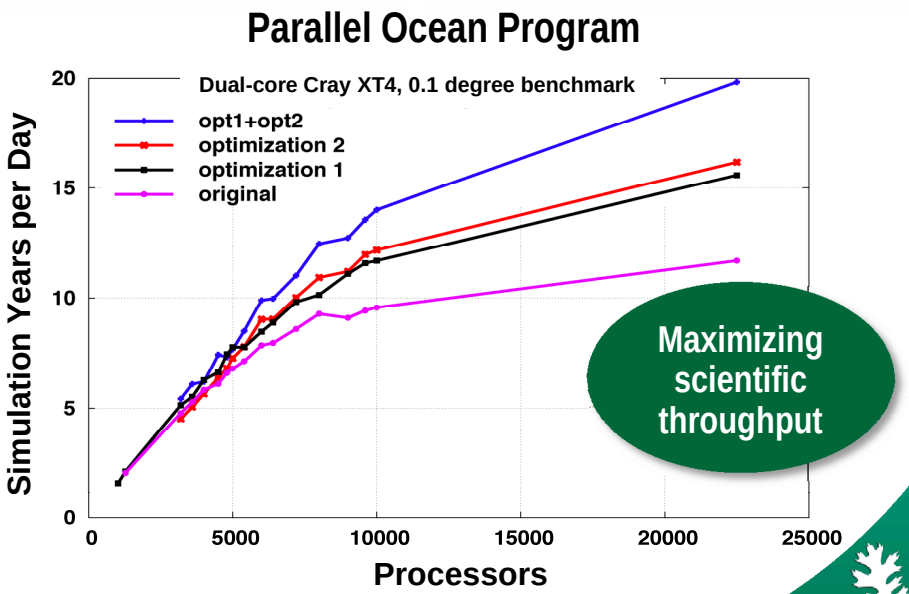
Application liaisons

- Long-term partnerships between PERI researchers and scientific code teams
- Currently working actively with 7 teams and tracking needs of 6 others

Tiger teams

- Focus on DOE's highest priorities: SciDAC-2, INCITE, JOULE
- Currently building models to estimate performance at scale and on new architectures





Performance modeling

Modeling is critical for automation of tuning

- Guidance to the developer
  - New algorithms, systems, etc.
- Need to know where to focus effort
  - Where are the bottlenecks?
- Need to know when we are done
  - How fast should we expect to go?
- Predictions for new systems

Recent progress

- Statistical techniques to extrapolate application scaling performance from smaller runs
- Technology to capture load-balance information at scale
- Capability to model MPI communication patterns within modeling assertions framework
- Open-source network simulator for modeling MPI performance
- Capability to explore performance impact of decomposition mapping strategies

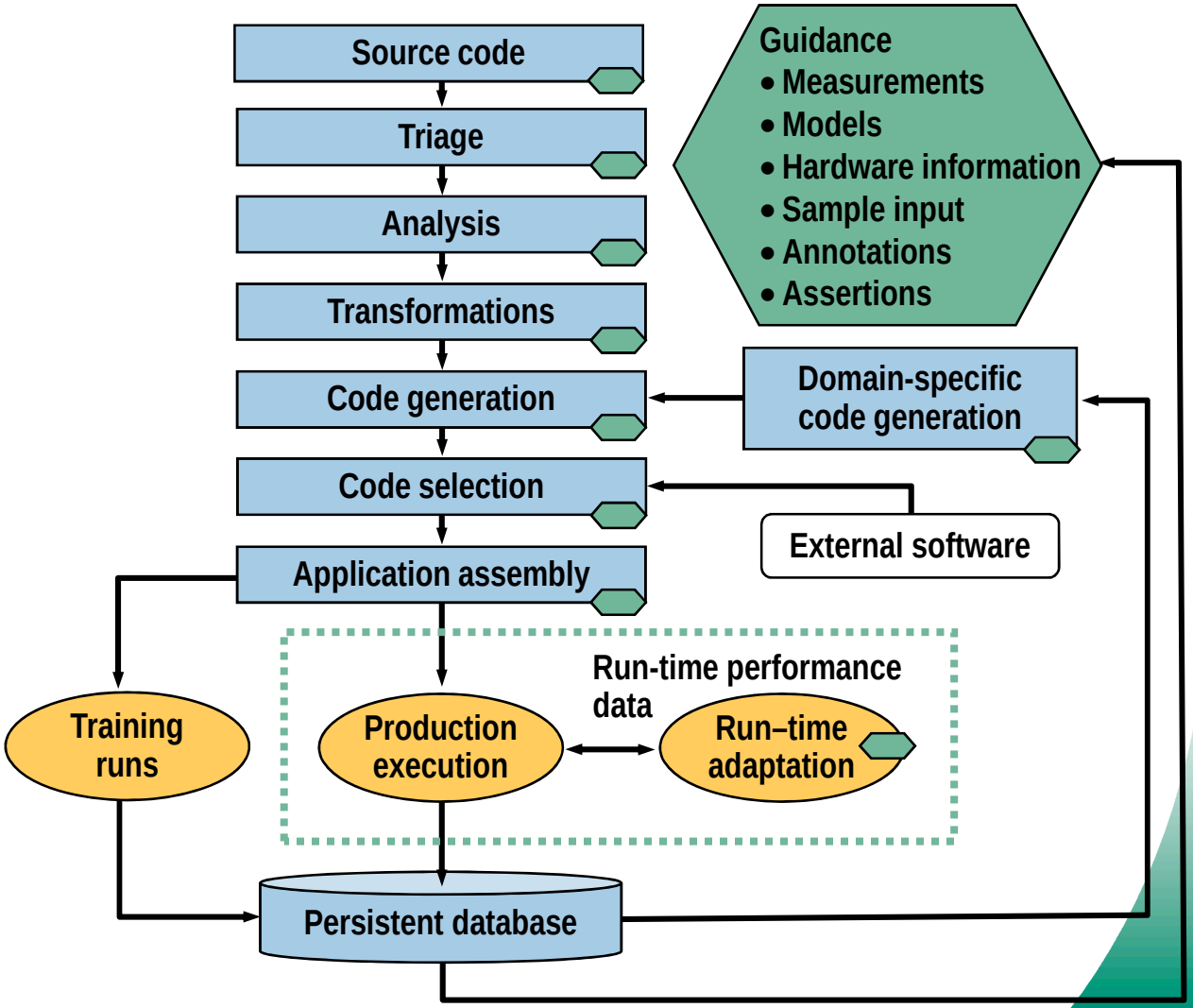
Modeling efforts contribute to procurements and other activities beyond PERI automatic tuning

Automatic performance tuning of scientific code

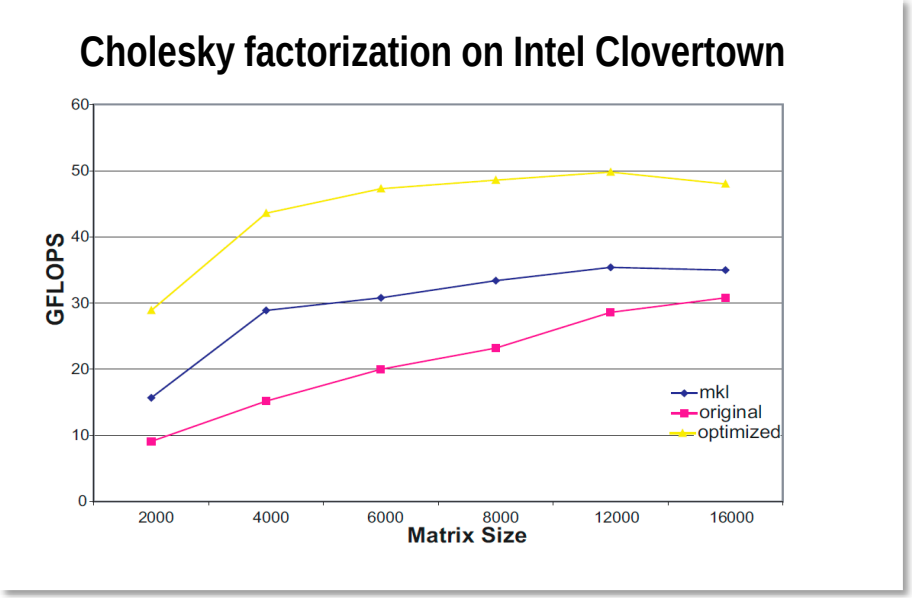
- **Long-term goals for PERI**
  - **Obtain hand-tuned performance from automatically generated code for scientific applications**
    - **General loop nests**
    - **Key application kernels**
  - **Reduce the performance portability challenge facing computational scientists**
    - **Adapt quickly to new architectures**
  - **Integrate compiler-based and empirical search tools into a framework accessible to application developers**
  - **Run-time adaptation of performance-critical parameters**

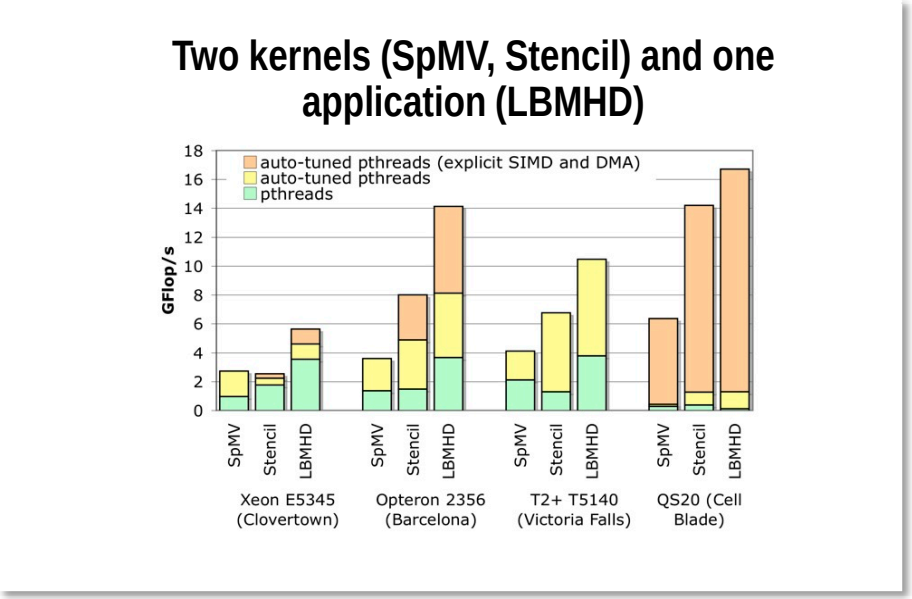
Automatic tuning flowchart

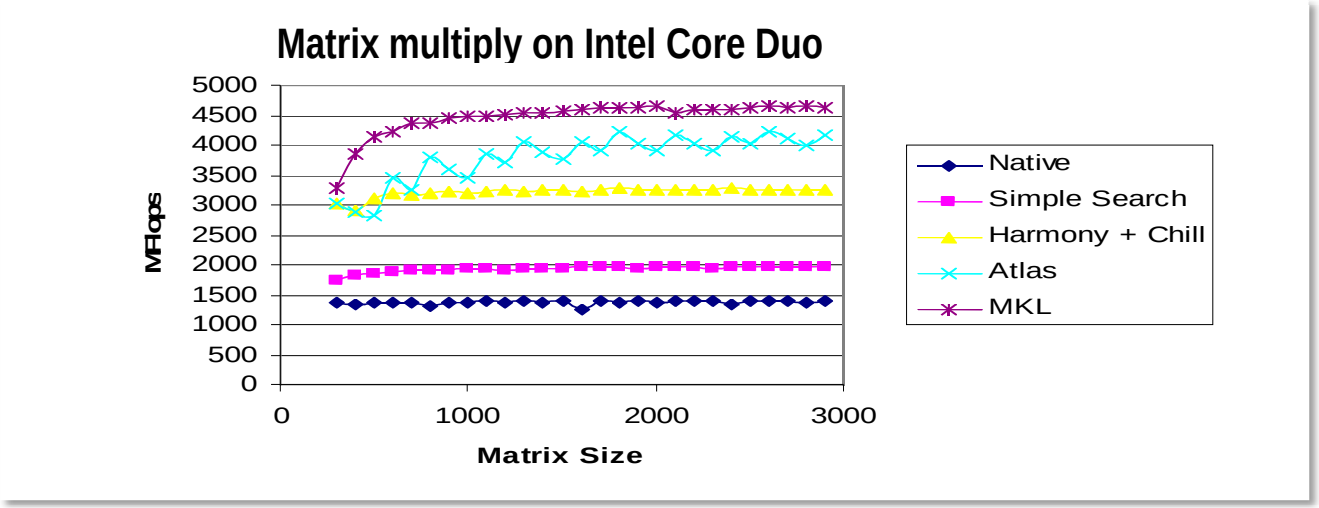
1: Triage	Where to focus effort
2: Semantic analysis	Traditional compiler analysis
3: Transformation	Code restructuring
4: Code generation	Domain-specific code
5: Code selection	Modeling and empirical search
6: Assembly	Choose the best components
7: Training runs	Performance data for feedback
8: Run–time adaptation	Optimize long-running jobs



Automatic tuning examples







The team

Argonne National Laboratory Paul Hovland Boyana Norris	**Lawrence Berkeley National Laboratory** David Bailey Katherine Yelick	**Lawrence Livermore National Laboratory** Bronis de Supinski Daniel Quinlan	**North Carolina State University** G. Mahinthakumar	**Oak Ridge National Laboratory** Philip Roth Jeffrey Vetter Patrick Worley (PI)	**Portland State University** Karen Karavanic	**Rice University** John Mellor-Crummey
University of California–San Diego Allan Snavely	**University of Maryland** Jeffrey Hollingsworth	**University of North Carolina** Rob Fowler	**University of Oregon** Allen Malony Sameer Shende	**University of Southern California** Jacqueline Chame Robert Lucas (PI)	**University of Tennessee** Jack Dongarra Shirley Moore	**University of Utah** Mary Hall

Contact

Patrick H. Worley

Computational Earth Sciences Group
Computer Science and Mathematics Division
(865) 574-3128
worleyph@ornl.gov

Osni Marques

Office of Advanced Scientific Computing Research
DOE Office of Science