

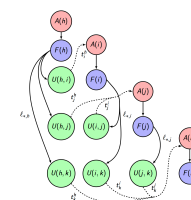
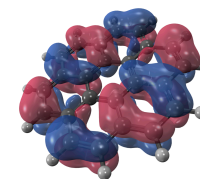
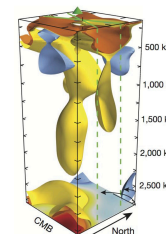
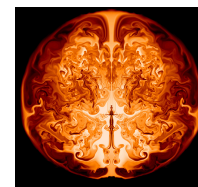


Extreme Scale Software Stack  
Program  
Summary of Impact on Science

DOE Program Manager: Sonia R. Sachs  
DOE Research Director: William Harrod  
May 2016

# DEGAS Project: Impact of advances in data Structures and Runtime Support for Irregular Data-Intensive Applications

- Distributed hash table
  - Applications: HipMer (genomics)
- Irregular data exchange
  - Applications: AMR, HPGMG
- Irregular global matrix update
  - Applications: NWChem, seismic tomography
- Distributed work queue
  - Applications: NWChem, Hartree-Fock
- Dynamic task graph
  - Applications: Sparse symmetric matrix solver



## Speedups

720x

1.2x

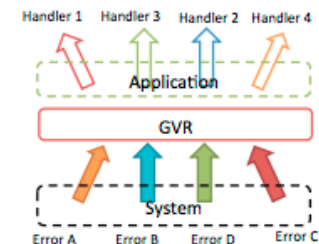
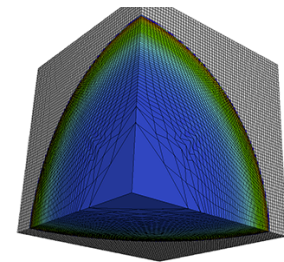
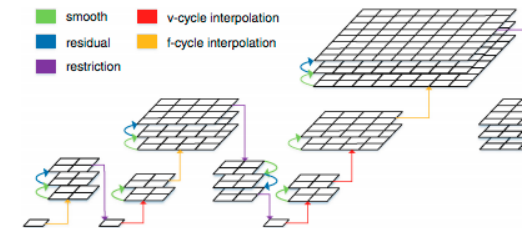
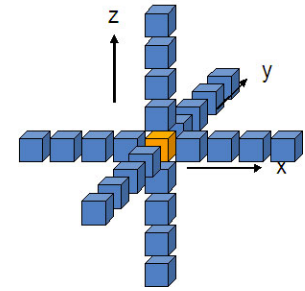
6x

1.2x

2x

# D-TEC Project: Impact of advances on DSL technologies, compilers and runtime systems

- AMR Shift Calculus DSL with ROSE/Polyopt
  - 7.9x for 3D 125 pts stencil
  - 7.3x for 2D 81 pts stencil
- Halide DSL
  - 4.25x for MiniGMG
  - 1.8x on GPU for HPGMG
- Bamboo
  - 1.27x in 32K size for algebraic multigrid
  - 1.29x with 96K grid cells for 3D stencil
- LULESH with X10
  - 1.12x better performance
  - 40% fewer lines of code
- Global-View Resilience (GVR)
  - 85% parallel efficiency on 1K processes with less than 2% code change

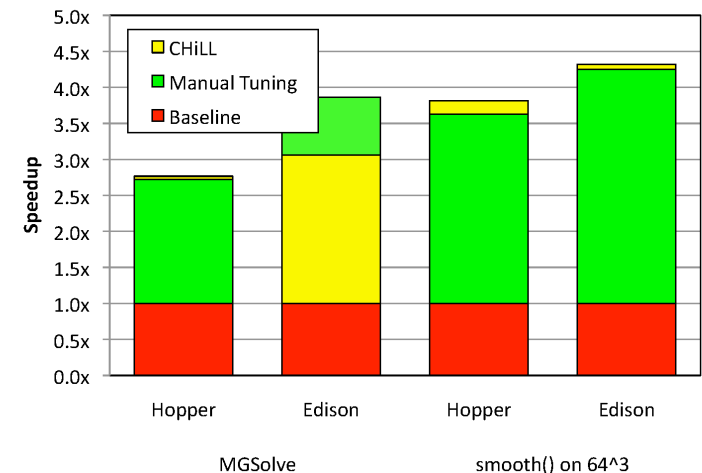


# Traleika Glacier Project: Impact of advances in runtime systems, compiler optimizations, and new programming models

<i>Technology</i>	<i>Rating</i>	<i>Improvements</i>	<i>Comments</i>
CnC	Promising	<b>9x on Genomics Smith-Waterman</b>	CnC specification and reference are open-sourced by Intel; Rice is now driving changes and enhancements; concern over viability for large-scale applications remains
Habanero	Promising		Habanero C support for OCR works well. The lack of a centralized standard or reference implementation for Habanero hampers ability to support OCR with all Habanero features and versions. Newer HC++ has more features, but has the same underlying challenges.
R-Stream	Good	<b>11x on HPGMG Chebyshev</b>	R-Stream is very effective at automatically generating high-performance OCR code from high-level C loop nests, that exploit the full feature set of OCR and allow users to get the full benefit of the EDT model vs. classical execution models.
OCR (Intel, Rice)	Good	<b>67% weak-scaling efficiency on 8000 nodes</b>	Functionally complete with a precise open specification and reference implementation. Advanced features on resiliency, hints, and related technologies are in progress.
OCR (PNNL)	Promising	<b>323x raytracing over 9600 cores</b>	A version derived from the open reference implementation with a focus on performance first. Source code is not available.
OCR (Vienna)	Promising	<b>62x on Seismic codes</b>	Working on distributed version for larger capabilities; a rapid evaluation framework that has been demonstrated as easy to test new ideas; fully standard compliant; open source.
FSim	Good	<b>Simulating 4,096 cores @ 10MHz ea</b>	Very fast, highly parallel, large-scale architecture simulator works and matches the RTL design model at a functional level. Supports bare-metal as well as OCR applications.
Tools	Limited		First-pass tools are working, but are not going to scale as-is to the expected size of future machines without significant additional work. Collaboration with DOE groups working on similar tools is just starting, and will remain in progress. Starting this was delayed until the last year of the XSTG project, due to the evolving OCR design and interfaces.

# Compiler-Directed Autotuning for Geometric Multi-grid

- Problem
  - Geometric multi-grid (GMG), is one of the most popular methods for solving partial differential equations, but is very difficult to optimize on evolving multi-core architectures.
- Solution
  - Leverage **communication-avoiding optimizations** which reduce communication overhead.
  - Apply **CHILL compiler** technology, using a set of novel automated code transformations.
  - Make the approach portable, via **autotuning system** that explores tradeoffs between reduced communication and increased computation, as well as tradeoffs in threading strategies.
  - Demonstrate optimizations on **mini-GMG**, designed to proxy MGSolve in BoxLib/Chombo codes.
- Recent results
  - Improved overall multi-grid solve execution by over 3x on NERSC Edison vs. reference version.  
(*Basu et al., HIPC 2013, WOSC 2013, WOSC 2014*)
  - Improved smooth operator at finest level by over 4x.
  - **CHILL-generated code outperforms manual tuning.**
- IMPACT
  - Achieves comparable and sometimes better performance than hand-tuned code without sacrificing programmer productivity.
  - Demonstrates capability of compiler-directed autotuning, with broad impact on important numerical methods for the DOE Office of Science.



# Compiler Optimization of Higher-Order Stencils in Geometric Multigrid

- Problem
  - Geometric multigrid (GMG), is one of the most popular methods for solving partial differential equations, but it is very difficult to optimize its data movement on evolving multi-core architectures.
  - Higher-order stencils in GMG deliver orders of magnitude improvements in accuracy, but can be compute bound and require excessive register storage to reuse input data.
- Solution
  - Extend compiler technology to optimize compute-intensive stencils using **partial sums**, which **reduce computation and data movement**.
  - Apply **communication-avoiding optimizations** to further reduce data movement.
  - Demonstrate optimizations on **miniGMG**, designed to proxy MGSolve in BoxLib/Chombo codes.

## Recent results

- Improved smooth operator at finest level by over 4x and solver by over 3x. (*Basu et al., IPDPS 2015*)
- Significantly outperforms baseline roofline memory bound via communication-avoiding optimizations.

## IMPACT

- Automated optimization increases programmer productivity and portability, as well as performance.
- 125pt (10<sup>th</sup> order) stencil increases accuracy by 1000X over 7pt (2<sup>nd</sup> order), with only 2X slowdown. Achieving comparable accuracy for 2<sup>nd</sup> order requires 16 Million times more memory.

