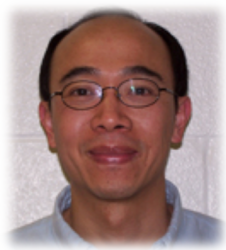
Co-Design of Hardware/Software for Predicting MAV Aerodynamics





E. de Sturler, **W. Feng**, C. Roy, A. Sandu, D. Tafti

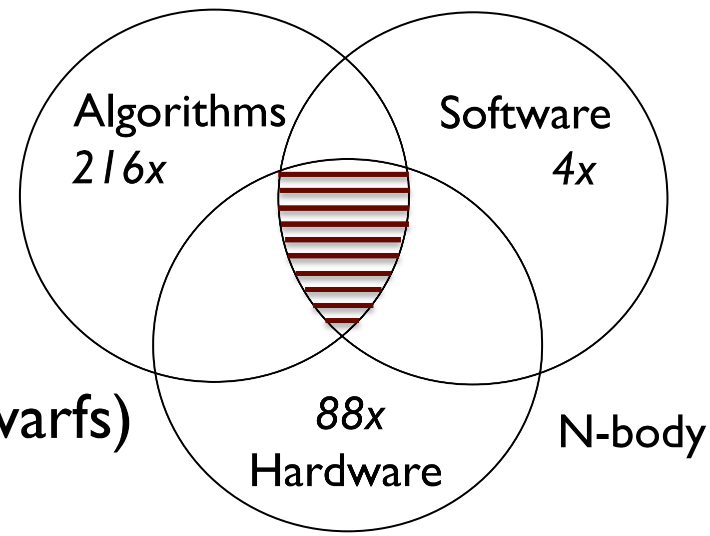




J. Edwards, H. Luo, F. Mueller

Fariba Fahroo, Computational Mathematics

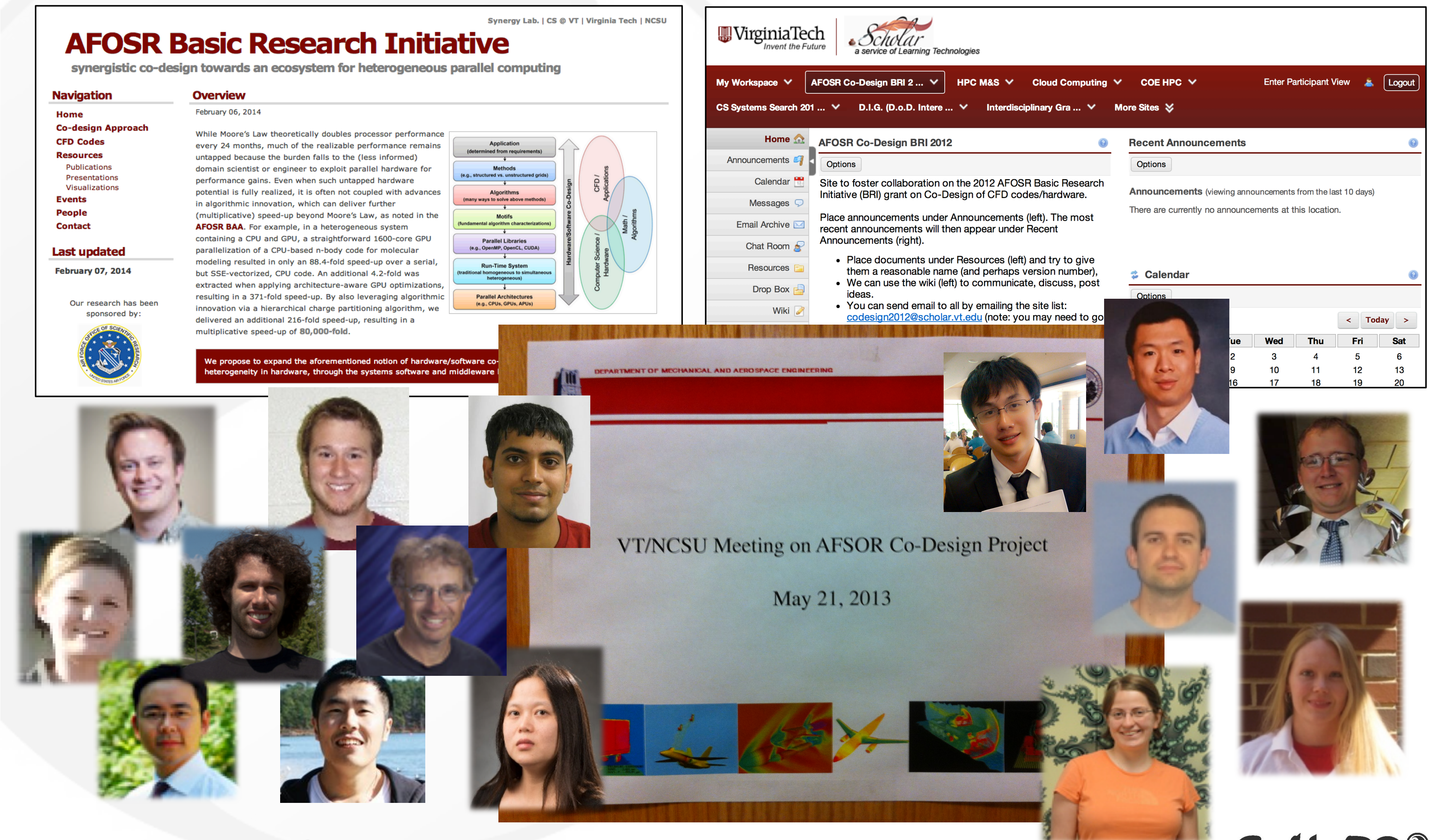
2014 AFOSR BRI Workshop
W. Feng, wfeng@vt.edu, 540.315.1545

Vision



- Synergistic co-design process for the *structured/unstructured grid motifs* (or dwarfs) in computational fluid dynamics (CFD) to support aerodynamic predictions for micro-air vehicles (MAVs).
  - Malleable **algorithms**
    … that can be mapped and optimized in **software**
    … onto the right type of processing core in **hardware**
    … at the right time
  - Co-design feedback to vendors to assist in guiding future hardware design

Synergistic Co-Design: Enabling and Empowering



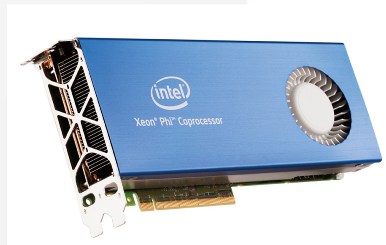
Team @ VT & NCS

- Virginia Tech (13)
  - Eric de Sturler — Numerical Methods (solvers & preconditioners)
  - Wu Feng — Parallel Computing (performance, power, portability)
  - Chris Roy — CFD (structured grid and ALE mesh movement)
  - Adrian Sandu — Numerical Methods (time stepping & discretization)
  - Danesh Tafti — CFD (pressure-based multiblock structured)
  - Research Scientist (1), Postdocs (2), and Graduate Students (5)
- North Carolina State University (7)
  - Jack Edwards — CFD (multiblock structured w/ implicit solvers)
  - Hong Luo — CFD (unstructured grid / compressible)
  - Frank Mueller — Parallel Computing (languages, compilers, scalability)
  - Postdocs (2), Graduate Students (2)

Why Synergistic Co-Design? Why Now?

- Increasing heterogeneity in computing resources

Altera FPGA



Intel MIC

NVIDIA GPU

TI DSP

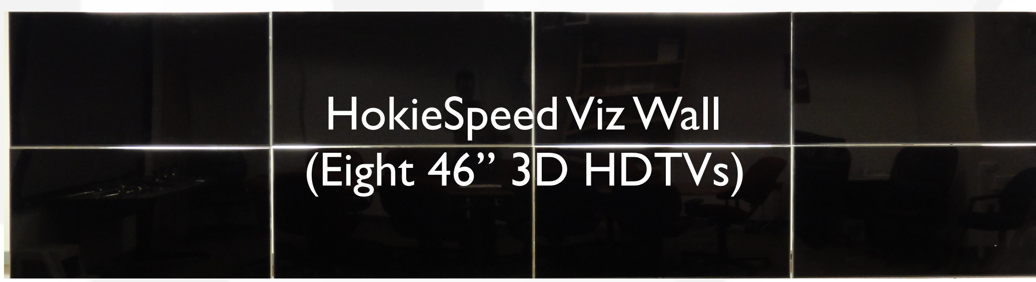
… across a wide variety of environments

Heterogeneous Systems in HPC

- Statistics
  - *Four* out of top 10 systems
  - Performance share in Top500 systems
    5% (2009) → 35% (2013)
- HokieSpeed
  - CPU+GPU heterogeneous supercomputer with large-scale visualization wall

Debuted as *GREENEST* commodity supercomputer in the U.S. in Nov. 2011 on THE GREEN 500

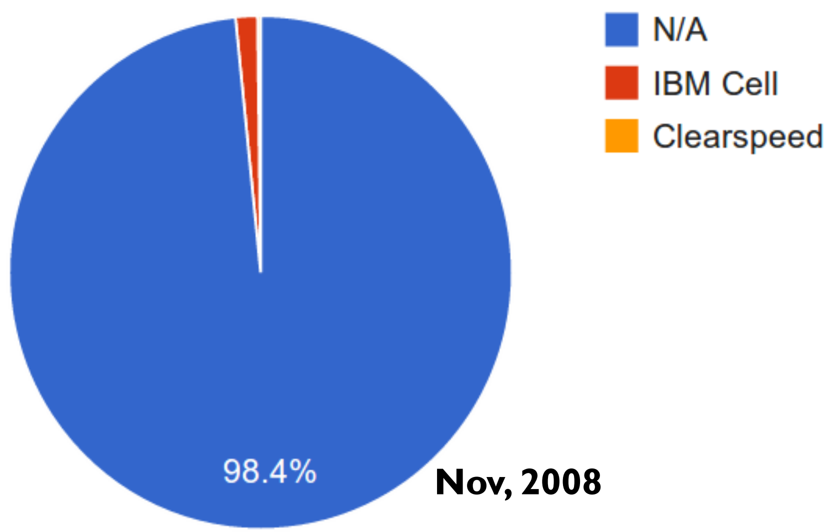
Tianhe-2

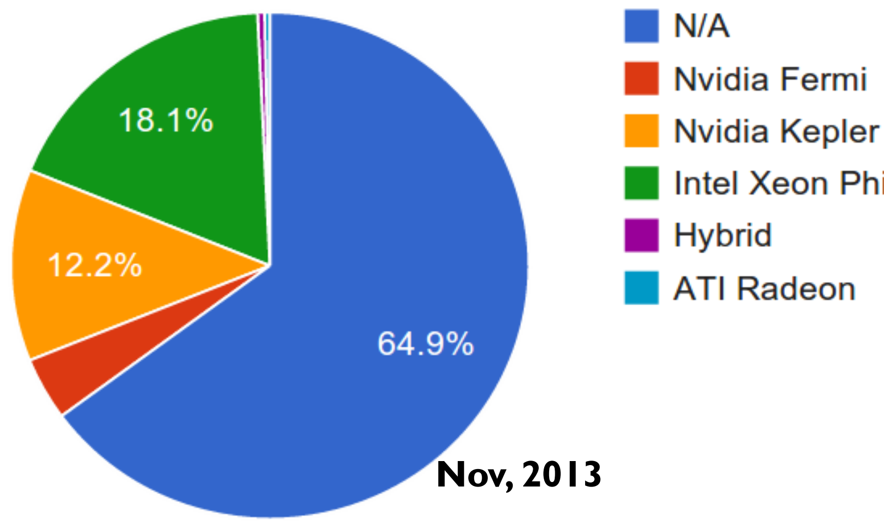


Top 10 Source: top500.org

1. **Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 ... ess-2, Intel Xeon Phi 31S1P
   NUDT
2. **Titan** - Cray XK7 , Opteron 6274 16C ... interconnect, NVIDIA K20x
   Cray Inc.
3. **Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom
   IBM
4. K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect
   Fujitsu
5. **Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom
   IBM
6. **Piz Daint** - Cray XC30, Xeon E5-2670 ... interconnect , NVIDIA K20x
   Cray Inc.
7. **Stampede** - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband ... DR, Intel Xeon Phi SE10P
   Dell
8. **JUQUEEN** - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect
   IBM
9. **Vulcan** - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect
   IBM
10. **SuperMUC** - iDataPlex DX360M4, Xeon E5-2680 8C 2.70GHz, Infiniband FDR

Diversity of Heterogeneous Systems



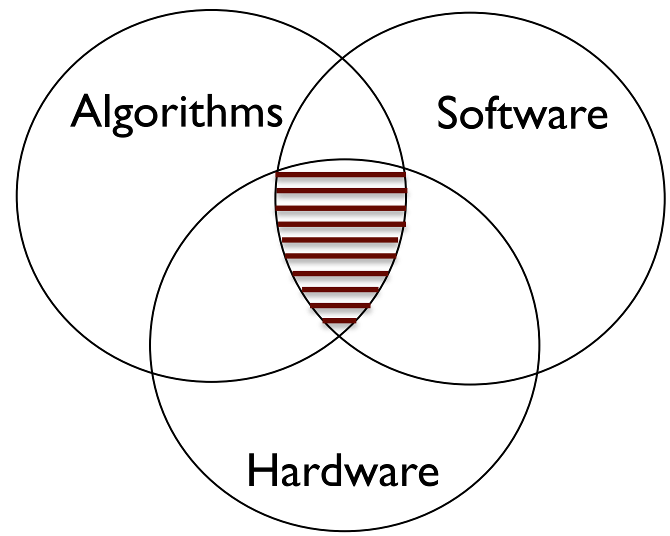


Performance Share of Accelerators in Top500 Systems

W. Feng, wfeng@vt.edu, 540.315.1545

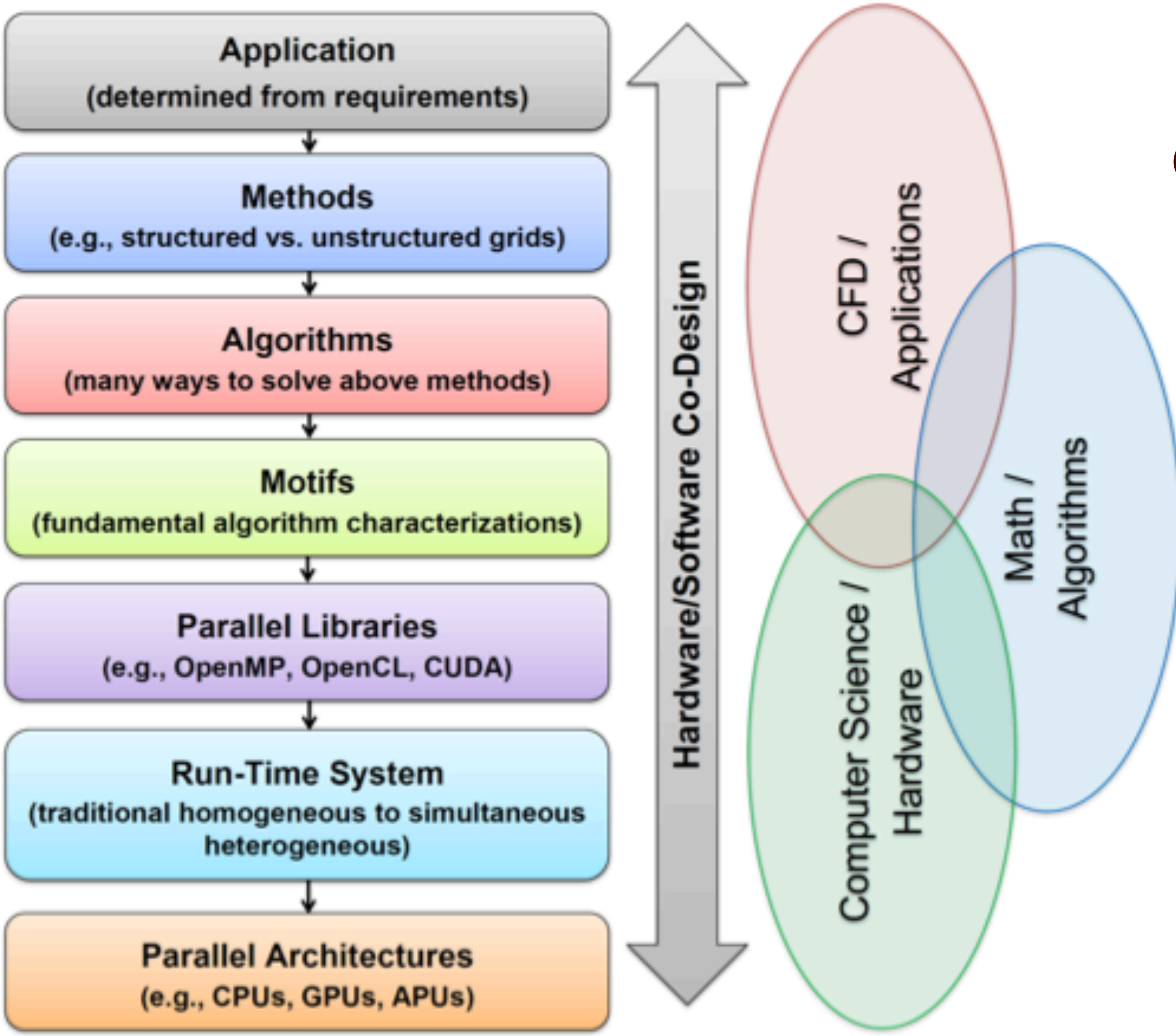
Roadmap

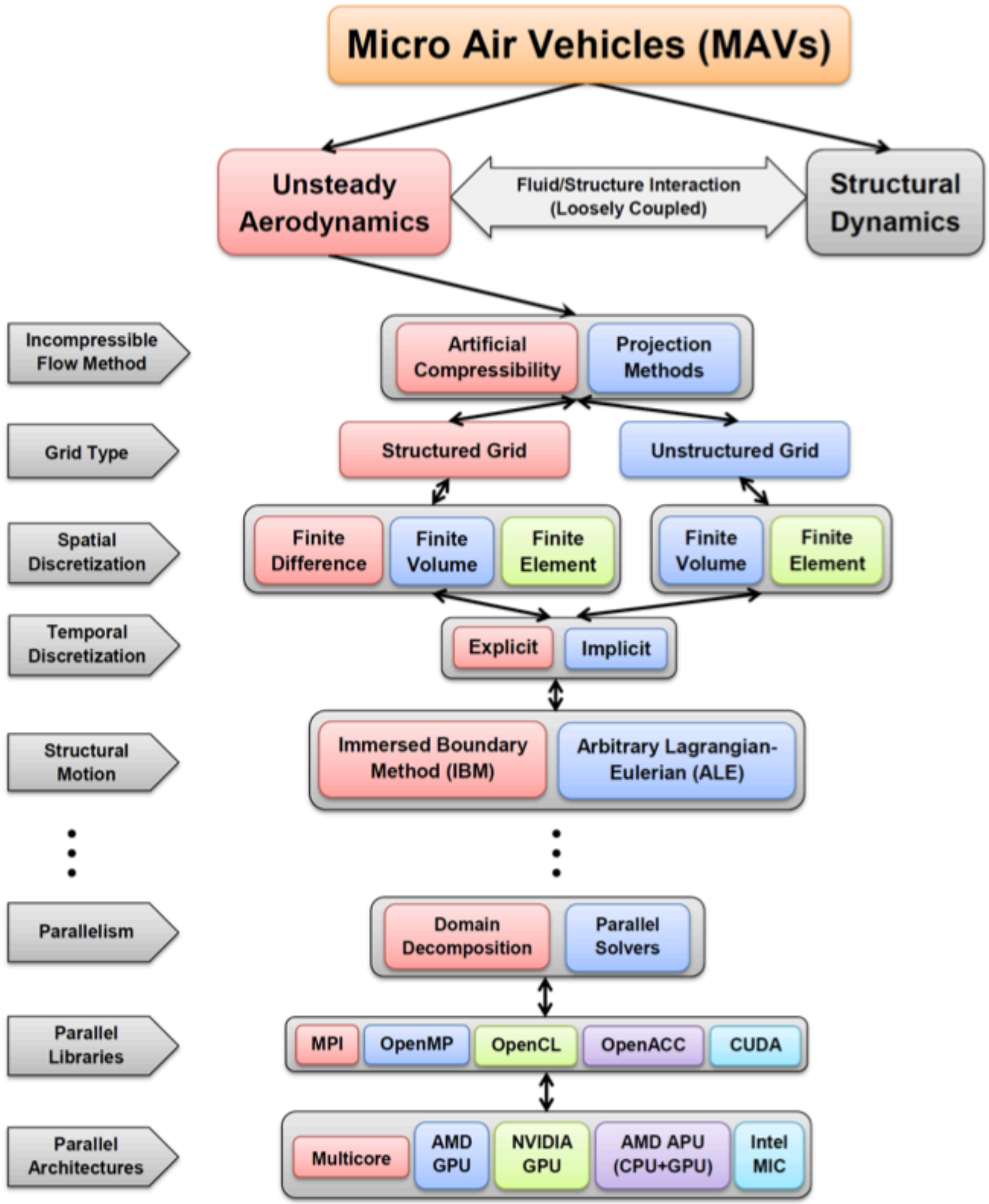
- Vision
- Team
- **Approach**
- Infrastructure
- Co-Design Research
  - Computer Science
  - CFD Codes (4)
  - Mathematics
- Achievements & Publications
- Next Steps



Concept presented at a White House BIGDATA Event in May 2013

Layered Co-Design





Co-Design for Micro Air Vehicles (MAVs)

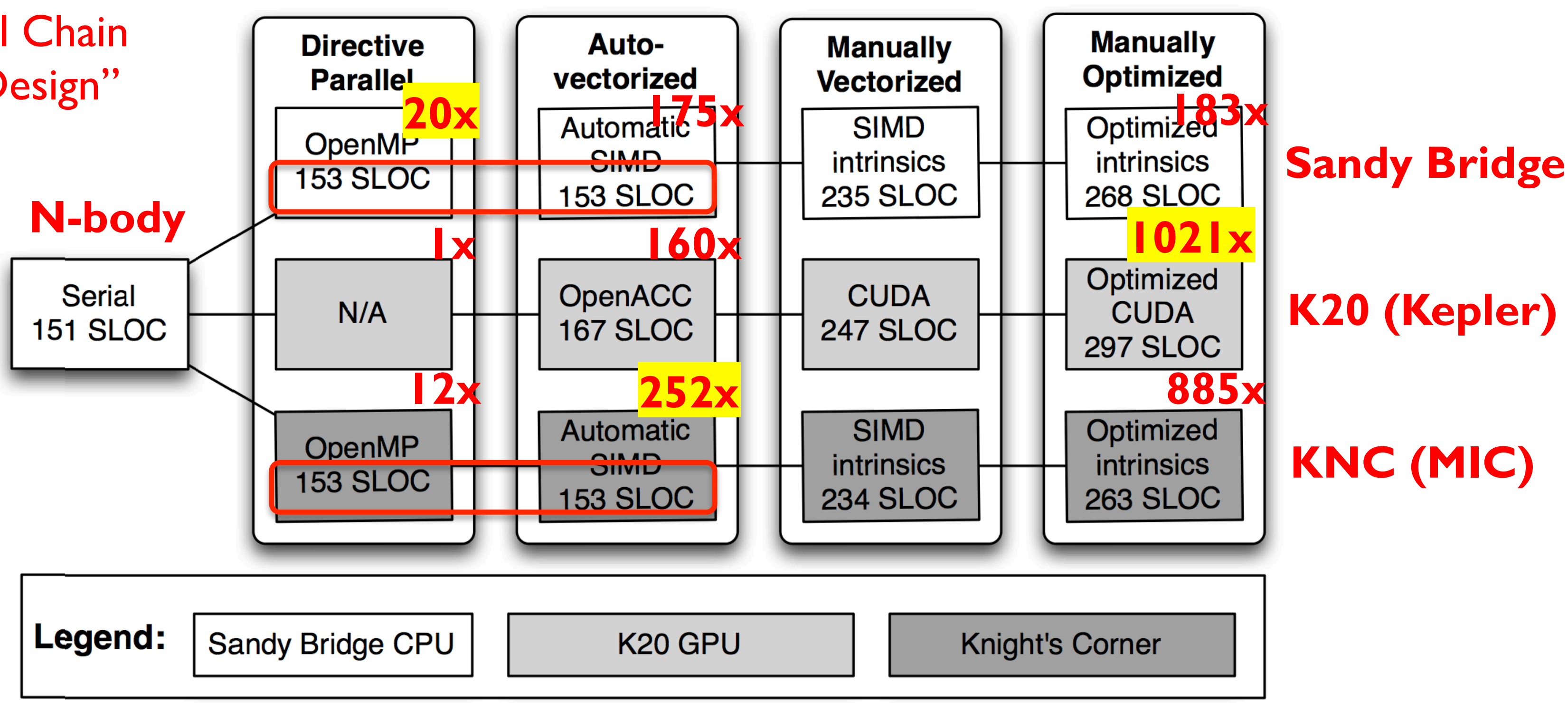
Co-Design Around Three P's:
Performance, Programmability, Portability

"Productivity = Performance + Programmability + Portability"

– Multi-dimensional optimization across two or more P's

… first *manual co-design* … then *automated co-design*

See "Tool Chain for Co-Design"

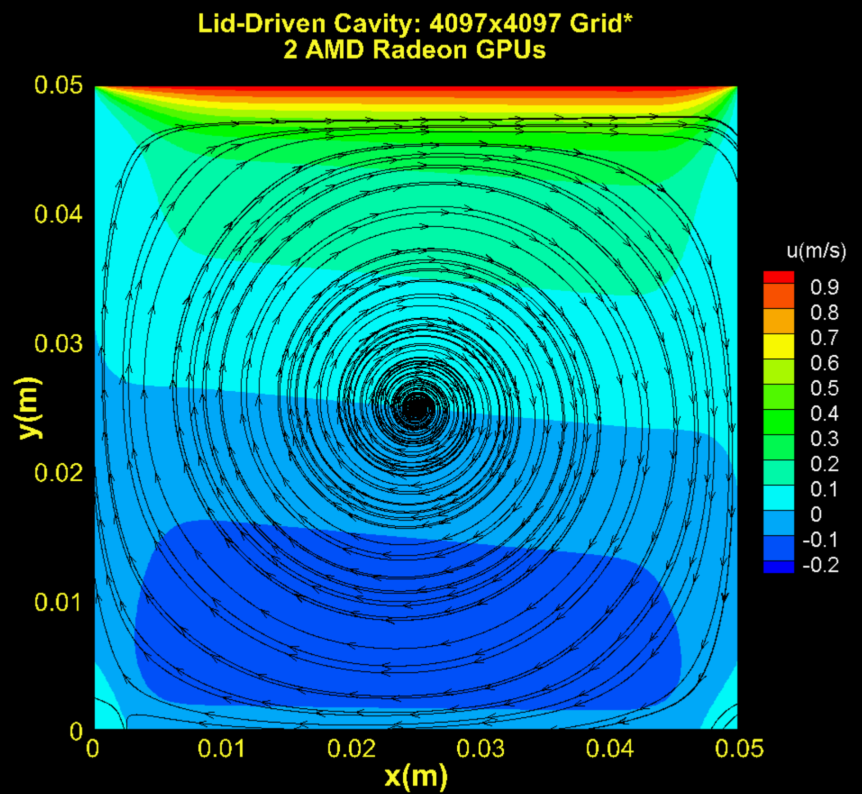


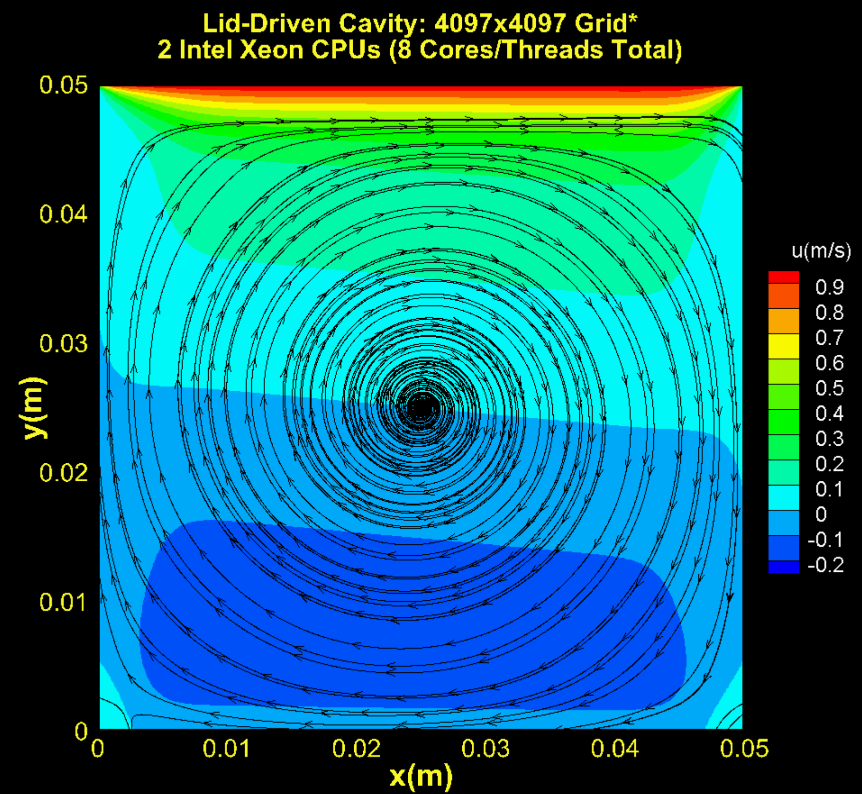
On the Programmability & Performance of Heterogeneous Platforms,
19th IEEE Int'l Conf. on Parallel and Distributed Systems, Dec. 2013.

Lid-Driven Cavity: 2 GPUs vs. 2 CPUs

Brent Pickering and Christopher Roy, Virginia Tech

8X Speed-Up



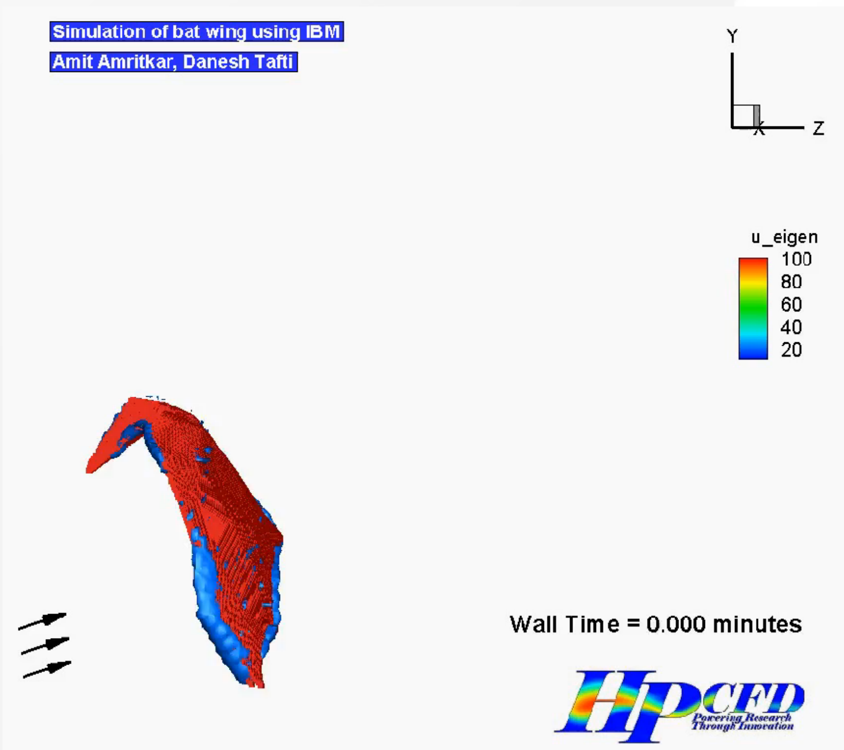


*Results shown on a 257x257 mesh to reduce file size

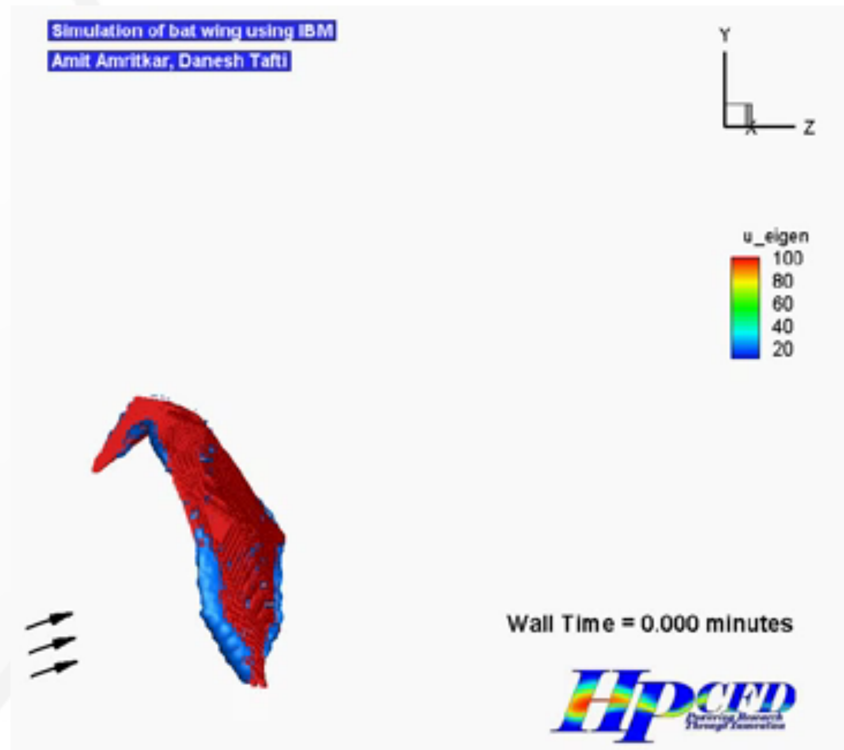
Bat Wing Simulation on GPU and CPU

Amit Amritkar and Danesh Tafti, Virginia Tech

GPU

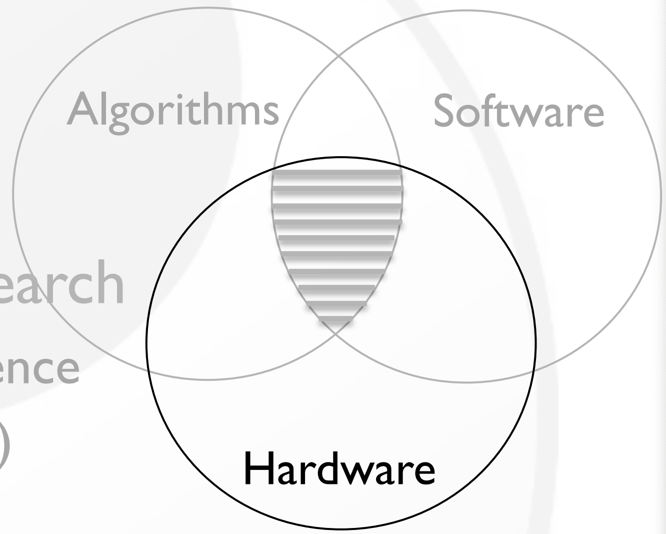


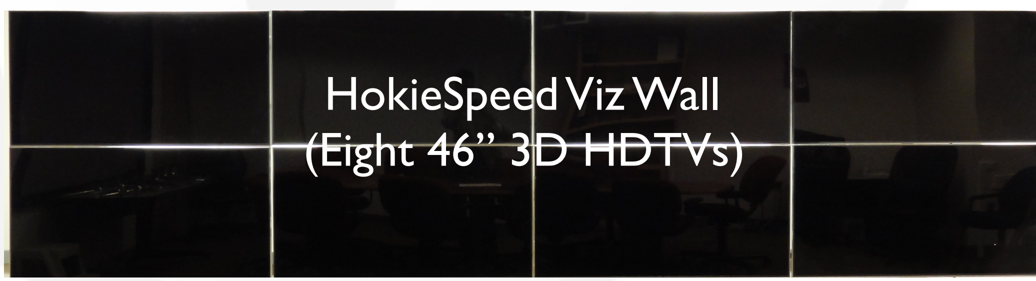
CPU



Roadmap

- Vision
- Team
- Approach
- **Infrastructure**
- Co-Design Research
  - Computer Science
  - CFD Codes (4)
  - Mathematics
- Achievements & Publications
- Next Steps





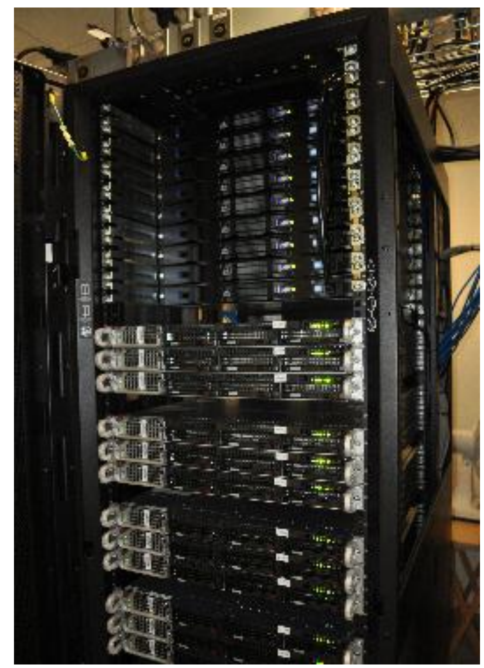
Experimental Systems

Add Intel MIC!

Add descriptive captions

ARC Cluster

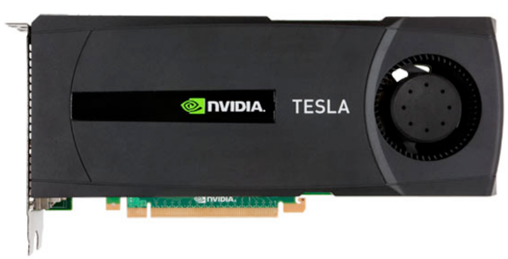
- Hardware
  - 2x AMD Opteron 6128 (8 cores each)
    - 120 nodes = ~2000 CPU cores
  - NVIDIA GTX480, GTX680 , C2050, K20c: 108 GPUs
  - Mellanox QDR InfiniBand: 40Gbit/s
- Software
  - CUDA 5.5
  - PGI Compilers V13.9 w/ CUDA Fortran & OpenACC support
  - OpenMPI & MVAPICH2-1.9 w/ GPUDirect V2 capability
  - Torque/Maui job management system

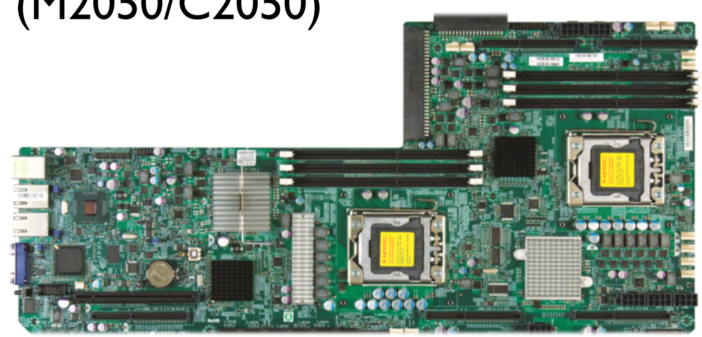


HOKIESPEED

A GPU-Accelerated Supercomputer for the Masses

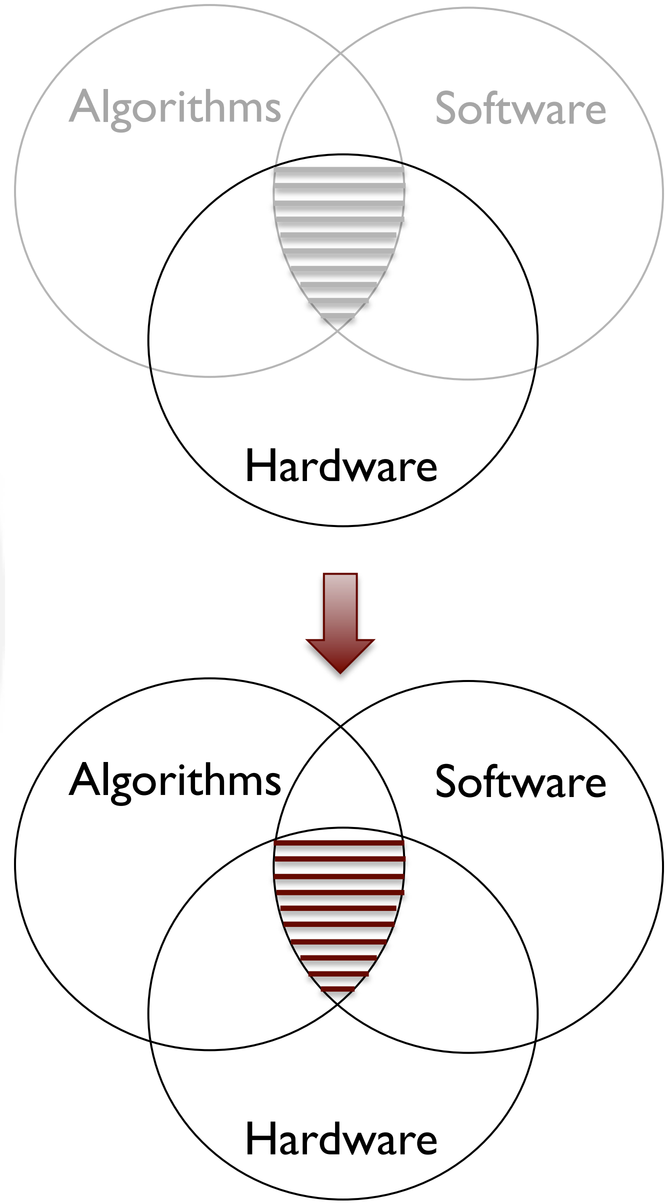
- Purpose
  - To catalyze new approaches for conducting research via the synergistic amalgamation of heterogeneous CPU-GPU hardware and software
- Profile
  - Total Nodes: 209, where each compute node consists of
    - Motherboard: Supermicro 2026GT-TRF Dual Intel Xeon
    - CPUs: Two 2.4-GHz Intel Xeon E5645 6-core (12 CPU cores per node)
    - GPUs: Two NVIDIA Tesla Fermi GPUs (M2050/C2050)





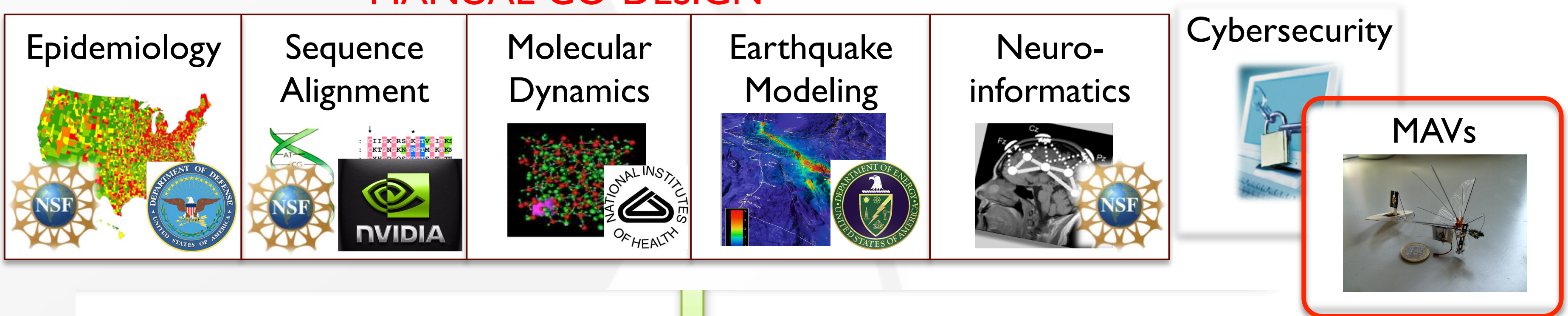
Roadmap

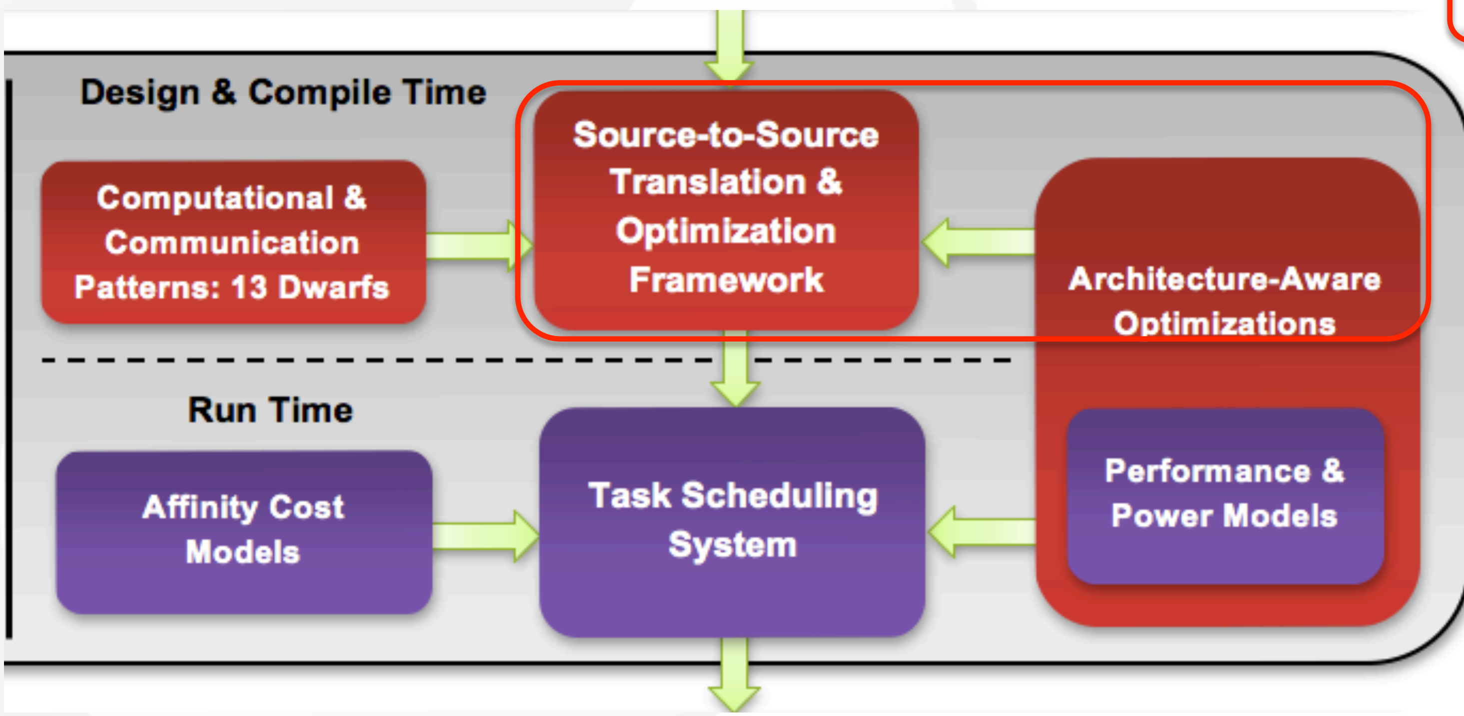
- Vision
- Team
- Approach
- Infrastructure
- Co-Design Research
  - Computer Science (Feng, Mueller)
  - CFD Codes (Edwards, Luo, Roy, Tafti)
  - Mathematics (de Sturler, Sandu)
- Achievements & Publications
- Next Steps



Intra-Node Ecosystem for Heterogeneous Parallel Computing

MANUAL CO-DESIGN





Heterogeneous Parallel Computing Platform

Manual Co-Design → Automated Co-Design

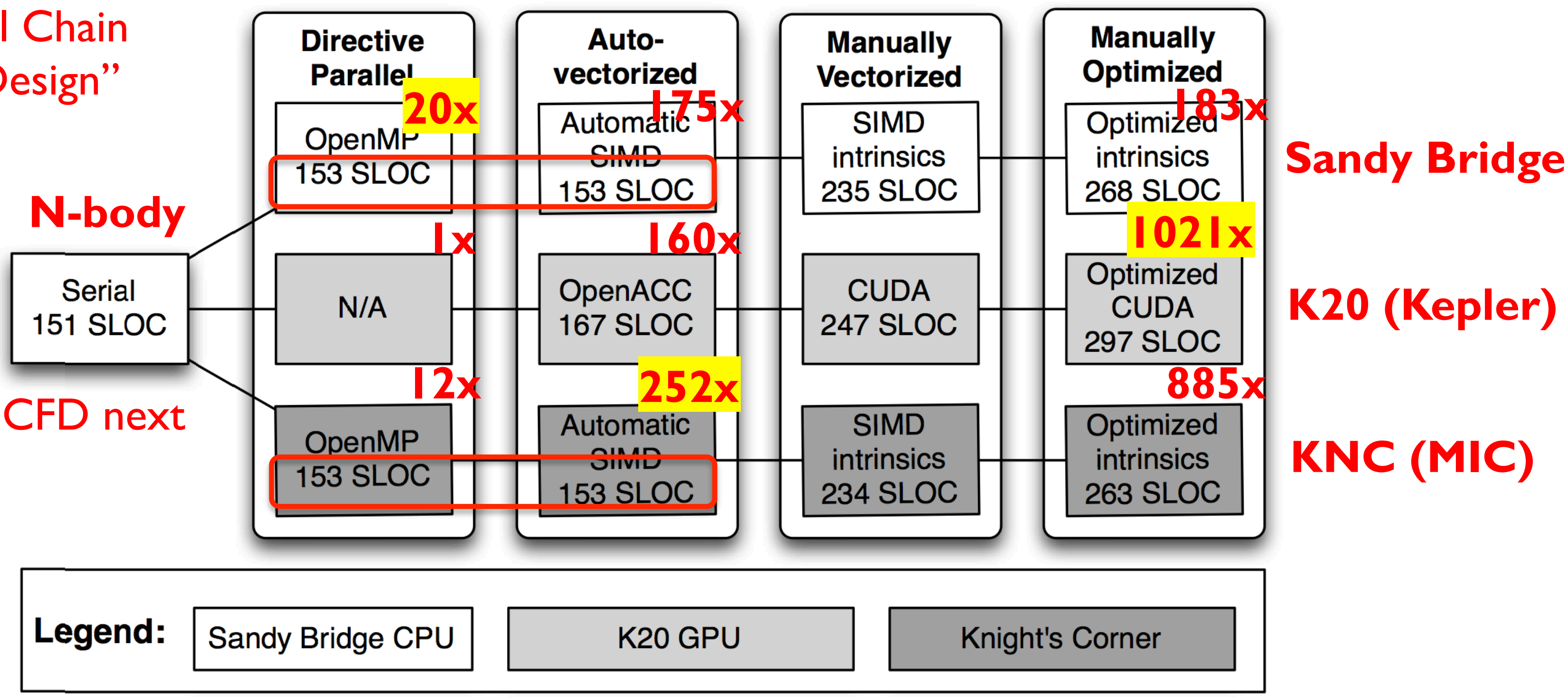
Co-Design Around Three P's:
Performance, Programmability, Portability

"Productivity = Performance + Programmability + Portability"

– Multi-dimensional optimization across two or more P's

… first *manual co-design* … then *automated co-design*

See "Tool Chain for Co-Design"

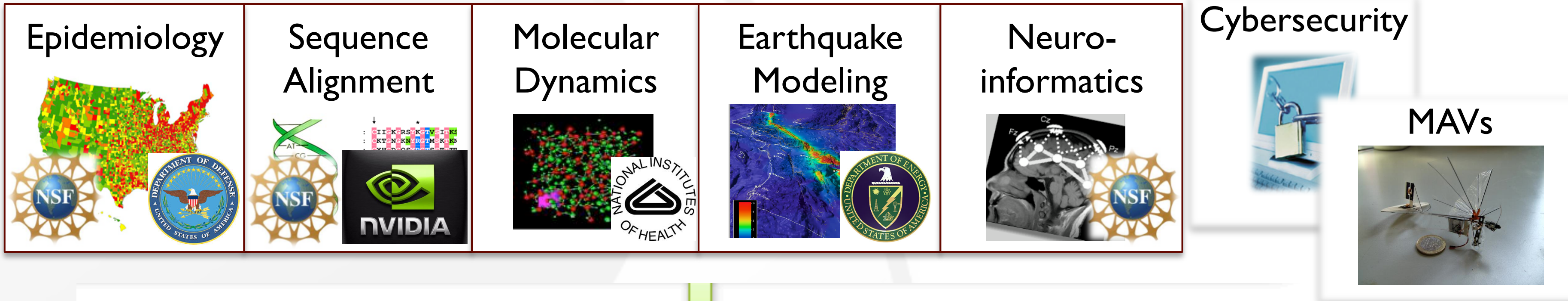


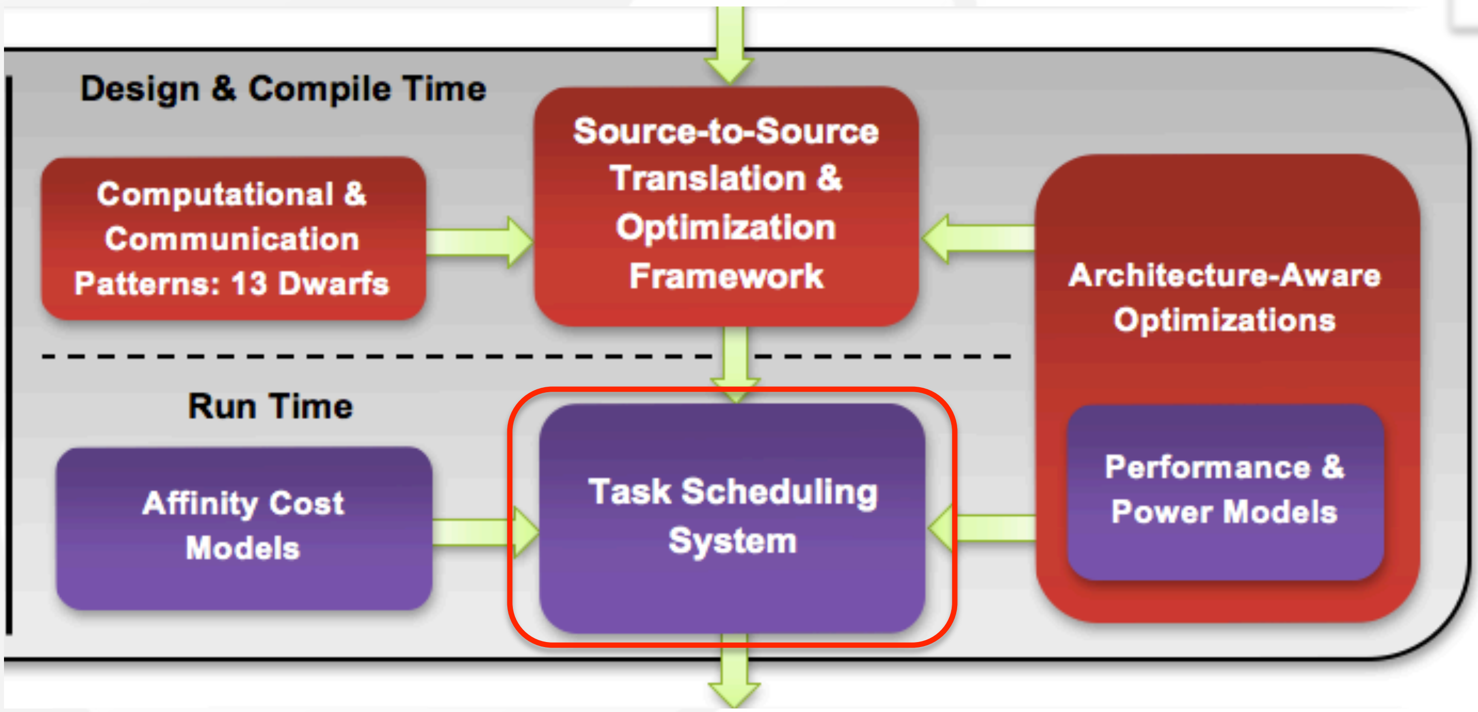
Apply to CFD next

On the Programmability & Performance of Heterogeneous Platforms, *19th IEEE Int'l Conf. on Parallel and Distributed Systems*, Dec. 2013.

Intra-Node Ecosystem for Heterogeneous Parallel Computing

MANUAL CO-DESIGN



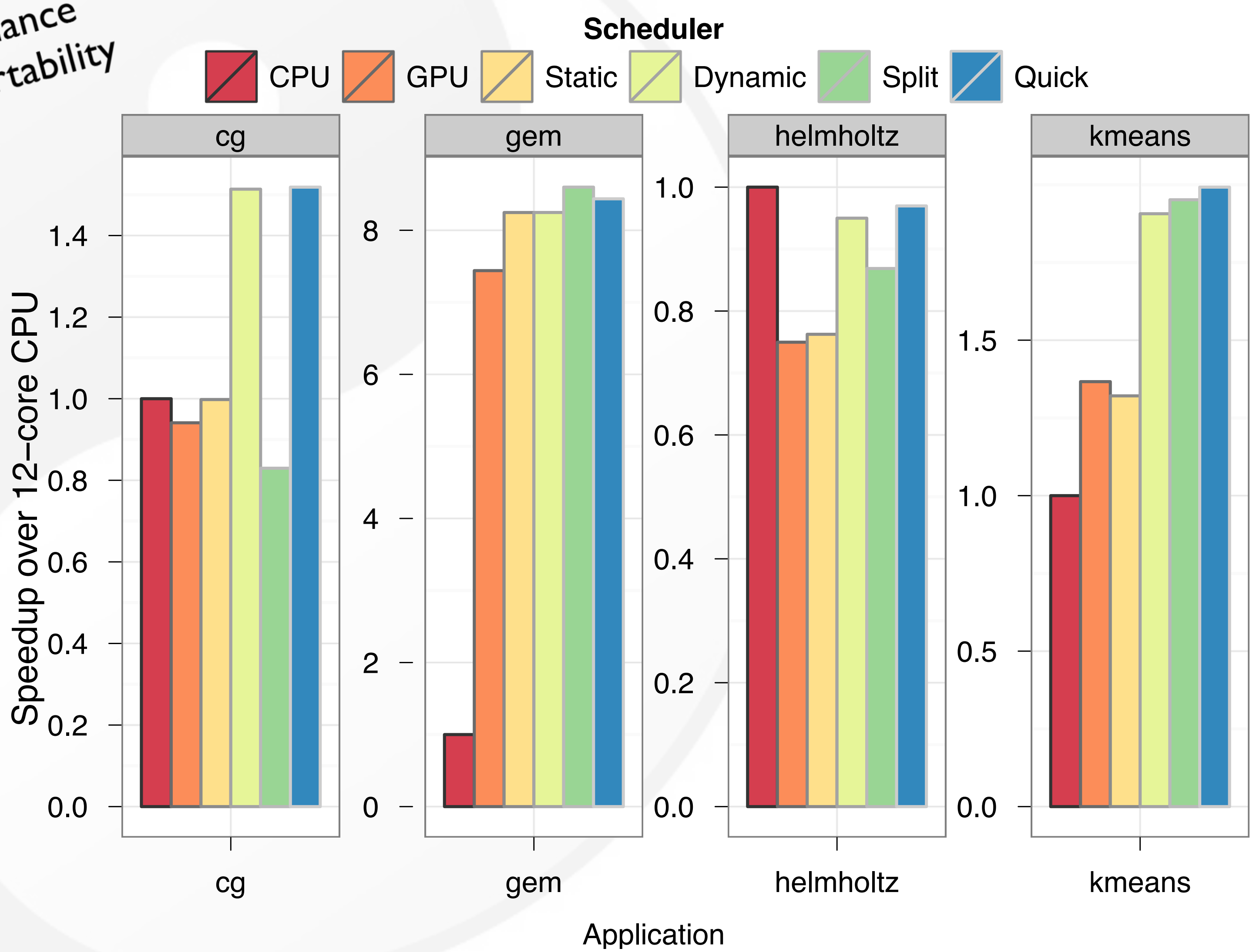


Manual Co-Design → Automated Co-Design

Heterogeneous Parallel Computing Platform

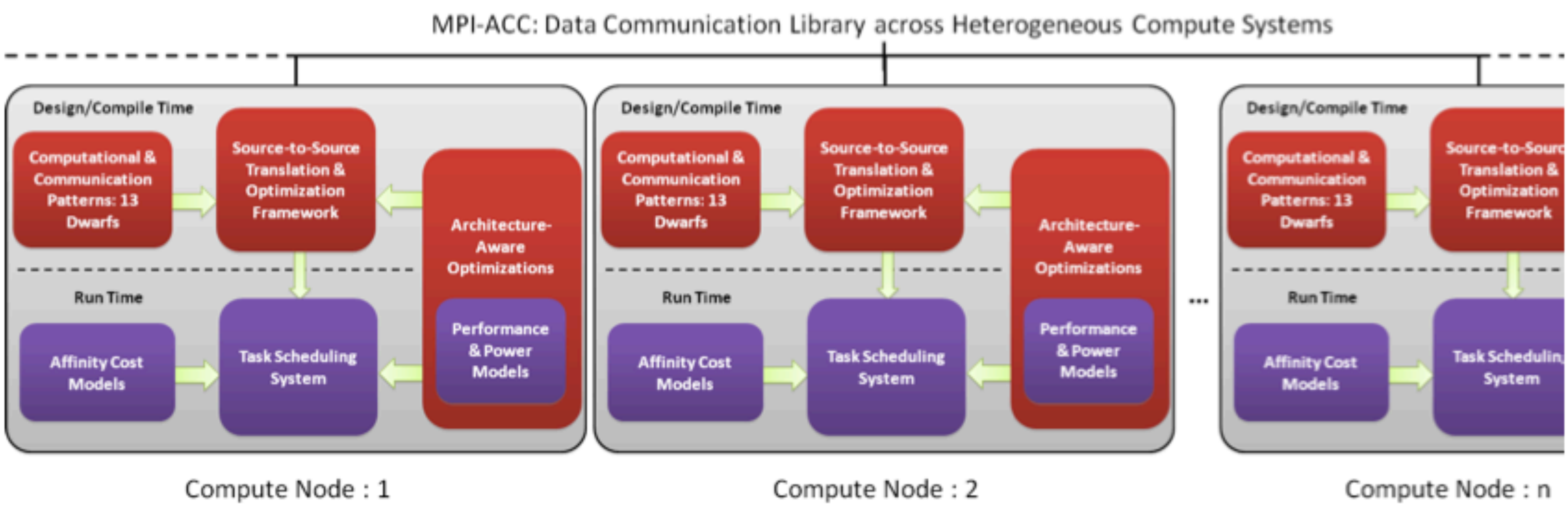
Preliminary Results of Automated Task Schedulers

Performance and Portability



Inter-Node Ecosystem for Heterogeneous Parallel Computing

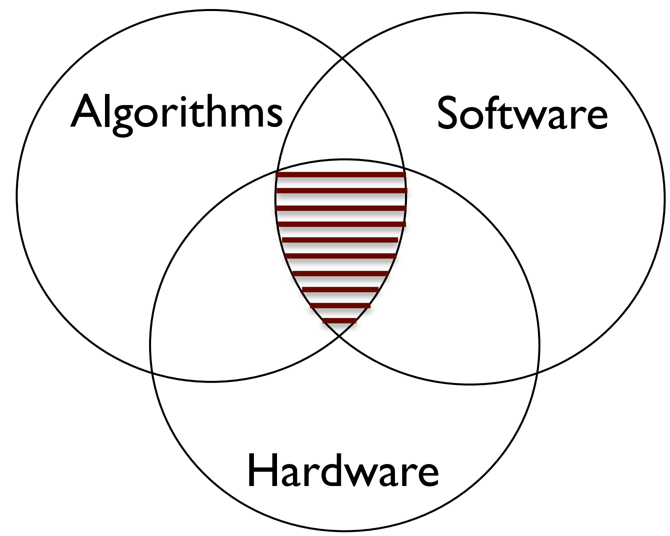
- Goal
  - Unified data movement library that hides all the hardware and system software details from the algorithm developer while supporting a multitude of environments



MPI-ACC: An Integrated and Extensible Approach to Data Movement in Accelerator-Based Systems by A. Aji, J. Dinan, D. Buntinas, P. Balaji, W. Feng, K. Bisset, R. Thakur. In Proc. 14th IEEE International Conference on High Performance Computing and Communications, Liverpool, UK, June 2012.

Roadmap

- Vision
- Team
- Approach
- Infrastructure
- Co-Design Research
  - Computer Science
  - CFD Codes (4)
  - Mathematics
- Achievements & Publications
- Next Steps



Overall Objective

- Extend and port our (four) CPU-based CFD methods to modern heterogeneous computing systems, particularly those with GPUs, via a synergistic hardware/software co-design approach that seeks to significantly improve performance, thus significantly enhancing the computational capabilities of current CFD tools.
  - Characterize and implement the following methods:
    - Projection and artificial compressibility methods for incompressible flows
    - Structured, unstructured, and Cartesian grids
    - Arbitrary Lagrangian-Eulerian (ALE) and immersed boundary methods (IBM) for boundary deformation
    - Finite volume and finite element methods based on the reconstructed discontinuous Galerkin (RDG) technique.
  - Co-design, test, verify, and assess the above methods for solving a variety of low Reynolds number incompressible flow problems of interest to the U.S. Air Force in general and for predicting MAV aerodynamics in particular.

Targeted CFD Codes

SENSEI (C. Roy, Virginia Tech)

- Structured, multiblock, 2nd order, finite-volume code
- Artificial compressibility method
- Arbitrary Lagrangian/Eulerian (ALE) 2nd or higher order spatial accuracy
- Artificial compressibility (AC) and immersed boundary (IB) methods

GenIDLEST (D. Tafti, Virginia Tech)

- Structured, multiblock, 2nd order, finite-volume code
- Pressure projection method
- ALE and immersed boundary methods (IBM)

RDGFLO (H. Luo, NCSU)

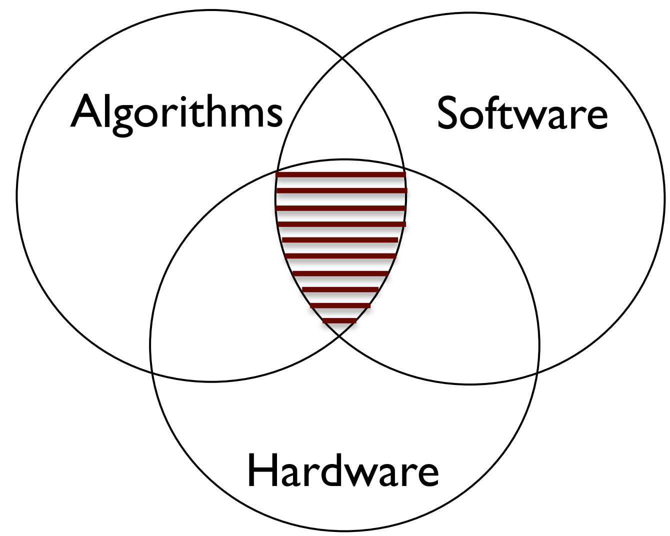
- Unstructured, discontinuous Galerkin (DG) method
- High-order solution of compressible flows

INCOMP3D (J. Edwards, NCSU)

- Structured, multiblock finite-volume code
- Second or higher order spatial accuracy
- ALE and IBM

Roadmap

- Vision
- Team
- Approach
- Infrastructure
- Co-Design Research
  - Computer Science
  - CFD Codes (4)
  - Mathematics (de Sturler, Sandu)
- Achievements & Publications
- Next Steps

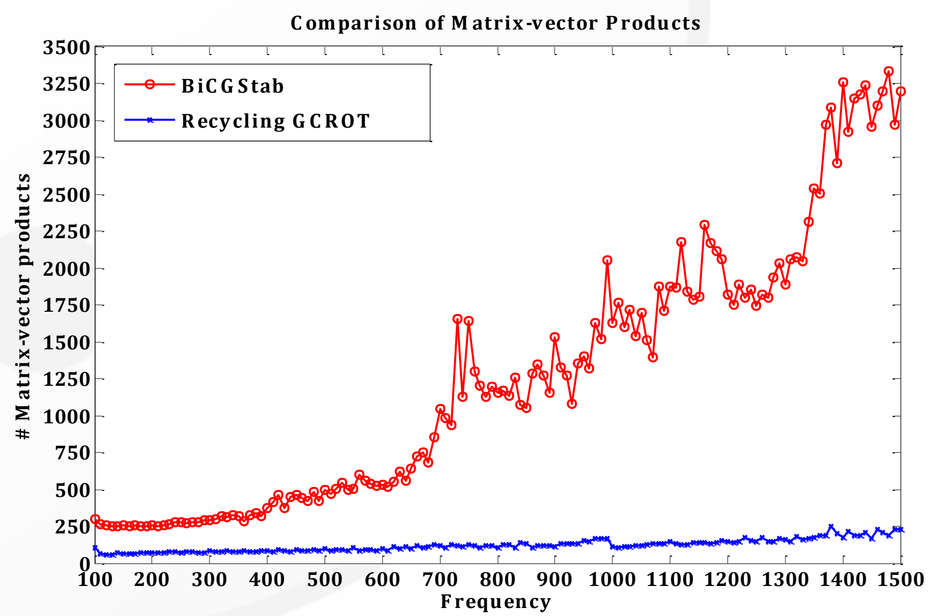


Solvers on GPUs and Multicore Processors

- Goal: Minimum Total Execution Time (vs Accuracy)
  - Fewer *expensive* iterations vs more *cheap* iterations
- Current General Effort
  - Efficient implementation of each kernel (data layout)
  - Efficient implementation of standard preconditioners
  - Better performance by varying parameters/preconditioners, memory-intensive algorithms have poor cache performance
- Future Efforts
  - Combine kernels/iterations, latency hiding, more arithmetic vs. data movement
  - Vary precision, analyze accuracy, and convergence
  - Alternative preconditioners
    - Faster convergence, better hardware utilization, data storage/access
    - Modify algorithms (solvers/preconditioners) – need to maintain good convergence

New(er) Solvers with Better Trade-offs

- Faster convergence (fewer iterations) for additional orthogonalizations and upfront matvecs, but all at once
  - Higher level BLAS/more work per data movement – trade off with sparse matvec
  - Better convergence allows cheaper preconditioner
  - Only one sync per extra vector in space
  - Especially useful for sequences of linear systems



Acoustics problem: Tire noise

Collaboration Jan Bierman, BMW

New Rosenbrock-Krylov Matrix-Free Integrators Using Minimal Implicitness

- New implicit matrix-free Rosenbrock-Krylov time integration methods avoid solution "to convergence" and guarantee order of consistency with small Krylov spaces (e.g., four-dimensional)
- Full-space linear algebra parallelized using cuBLAS. Arnoldi iteration parallelized with a custom CUDA kernel for the orthogonalization step.
- Tested on a parallel implementation of the shallow water equations using different grid sizes and tolerances.

Selected Achievements

- Computer Science
  - Manual Co-Design
    - GenIDLEST: 18-fold speed-up with four C2070 GPUs vs. serial CPU
    - Lid-Driven Cavity → SENSEI-Lite
      - 8-fold speed-up with a dual-GPU 7990 vs 8-core Intel Xeon CPU
      - 7-fold speed-up with four C2070 GPUs vs 8-core Intel Xeon CPU
  - At Compile/Design Time
    - Infer speedup potential before refactoring code via memory trace compression algorithms to estimate parallelization benefits
  - At Run Time
    - Initial evaluation of our automated run-time system prototype (for Accelerated OpenMP and OpenACC)
  - Tool Chain
    - Library for Computational Fluid Dynamics (CFD)
      - Initial scaffolding based on identifying commonality for library for CFD codes: generalized GPU-to-GPU communication (via MPI), ghost cell exchange between GPUs, …
    - Co-design collaboration with PGI on PGI Compiler Suite (including 13.6 beta → 13.10 beta → 14.1)

Selected Achievements

- Computational Fluid Dynamics (CFD)
  - CPU parallelization of prototype CFD codes with MPI+OpenMP
  - GPU parallelization of prototype CFD codes with MPI+OpenACC and MPI+CUDA Fortran
  - Multiple publications on different CFD codes: GenIDLEST, Lid-Driven Cavity → SENSEI-Lite, RDFLO3D, and INCOMP3D
- Math
  - Initial GPU-parallelized Rosenbrock-Krylov method
  - Integrated initial CUDA-parallelized solvers with CFD

Publications

- C. Li, Y. Yang, H. Dai, S. Yan, F. Mueller, H. Zhou, "Understanding the Tradeoffs between Software-Managed vs. Hardware-Managed Caches in GPUs", IEEE Int'l Symp. on Performance Analysis of Systems and Software, Mar. 2014.
- L. Luo, J. R. Edwards, H. Luo, F. Mueller, "Performance Assessment of Multi-block LES Simulations using Directive-based GPU Computation in a Cluster Environment," 52nd AIAA Aerospace Sciences Meeting (SciTech), Jan. 2014.
- Y. Xia, H. Luo, L. Luo, J. Edwards, J. Lou, F. Mueller "OpenACC-based GPU Acceleration of a 3-D Unstructured Discontinuous Galerkin Method", 52nd AIAA Aerospace Sciences Meeting (SciTech), Jan. 2014.
- Y. Xia, L. Luo, H. Luo, J. Edwards, F. Mueller, "GPU Acceleration of a Reconstructed Discontinuous Galerkin Method for Compressible Flows on Unstructured Grids", 52nd AIAA Aerospace Sciences Meeting (SciTech), Jan. 2014.
- B. Pickering, C. Jackson, T. Scogland, W. Feng, and C. Roy, "Directive-Based GPU Programming for Computational Fluid Dynamics," 52nd AIAA Aerospace Sciences Meeting (SciTech), Jan. 2014.
- S. R. Glandon, P. Tranquilli, A. Sandu, "Acceleration of matrix-free time integration methods", Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems at SC13, Nov. 2013.
- J. M. Derlaga, T. S. Phillips, and C. J. Roy, "SENSEI Computational Fluid Dynamics Code: A Case Study in Modern Fortran Software Development," AIAA Paper 2013-2450, 21st AIAA Computational Fluid Dynamics Conference, San Diego, CA, June 2013.
- P. Tranquilli, A. Sandu, "Rosenbrock-Krylov Methods for Large Systems of Differential Equations" http://arxiv.org/abs/1305.5481, May 2013.

What's Next?

- Platforms
  - AMD & Intel CPU, AMD APU, AMD & NVIDIA GPUs, Intel MIC
- Towards Ease of Use and Automation
  (for Performance, Programmability, and Portability)
  - Web resource for *tenets of synergistic co-design*
    … between algorithms, software, and hardware → automation (long term)
  - Towards a CFD library for heterogeneous computing systems
    - GPU-integrated MPI vs. GPUDirect, ghost cell exchange, bounds checking, …
  - Code repositories for production codes
- GPU-Integrated MPI Evaluation
  - Experimental platforms (MIC and next-generation APU w/ "infinite memory")
- GPU mixed-precision solvers, GPU-efficient preconditioners
- GPU-efficient accurate and stable high-order time stepping

Agenda

10:30-11:00 Opening Remarks and Overview of Project:
Co-Design of Hardware/Software for Predicting MAV Aerodynamics

11:00-11:30 Development of a Portable, GPU-Accelerated High-Order Discontinuous Galerkin CFD Code for Compressible Flows on Hybrid Grids

11:30-12:00 Performance Assessment of a Multi-block Incompressible Navier-Stokes Solver using Directive-based GPU Programming in a Cluster Environment

12:00-12:30 Cache Performance Prediction

12:30-13:30 Working Lunch

13:30-14:00 GPU Acceleration of the SENSEI CFD Code Suite

14:00-14:30 GENIDLEST Co-Design

14:30-14:45 Break

14:45-15:15 Accelerated Solvers for CFD

15:15-15:45 Co-design of Time-Stepping Algorithms for Large Aerodynamics Simulations

15:45-16:00 Break

16:00-16:30 Codifying and Applying a Methodology for Manual Co-Design and Developing an Accelerated CFD Library (Co-Design: CFD/Math/CS)

16:30-17:00 Tool Chain for Co-Design (Co-Design: CS/Tools)

17:00-17:30 Discussion

Acknowledgements

- This work was funded by the Air Force Office of Scientific Research (AFOSR) Computational Mathematics Program
  - Program Manager: Fariba Fahroo
  - Grant No. FA9550-12-1-0442