SAND2007-7493C

Reducing Data Migration in the Context of Adaptive Partitioning for AMR

The 19th IASTED International Conference on Parallel and Distributed Computing and Systems 2007

November 19, 2007

Johan Steensland

Advanced Software R&D, Sandia National Laboratories

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

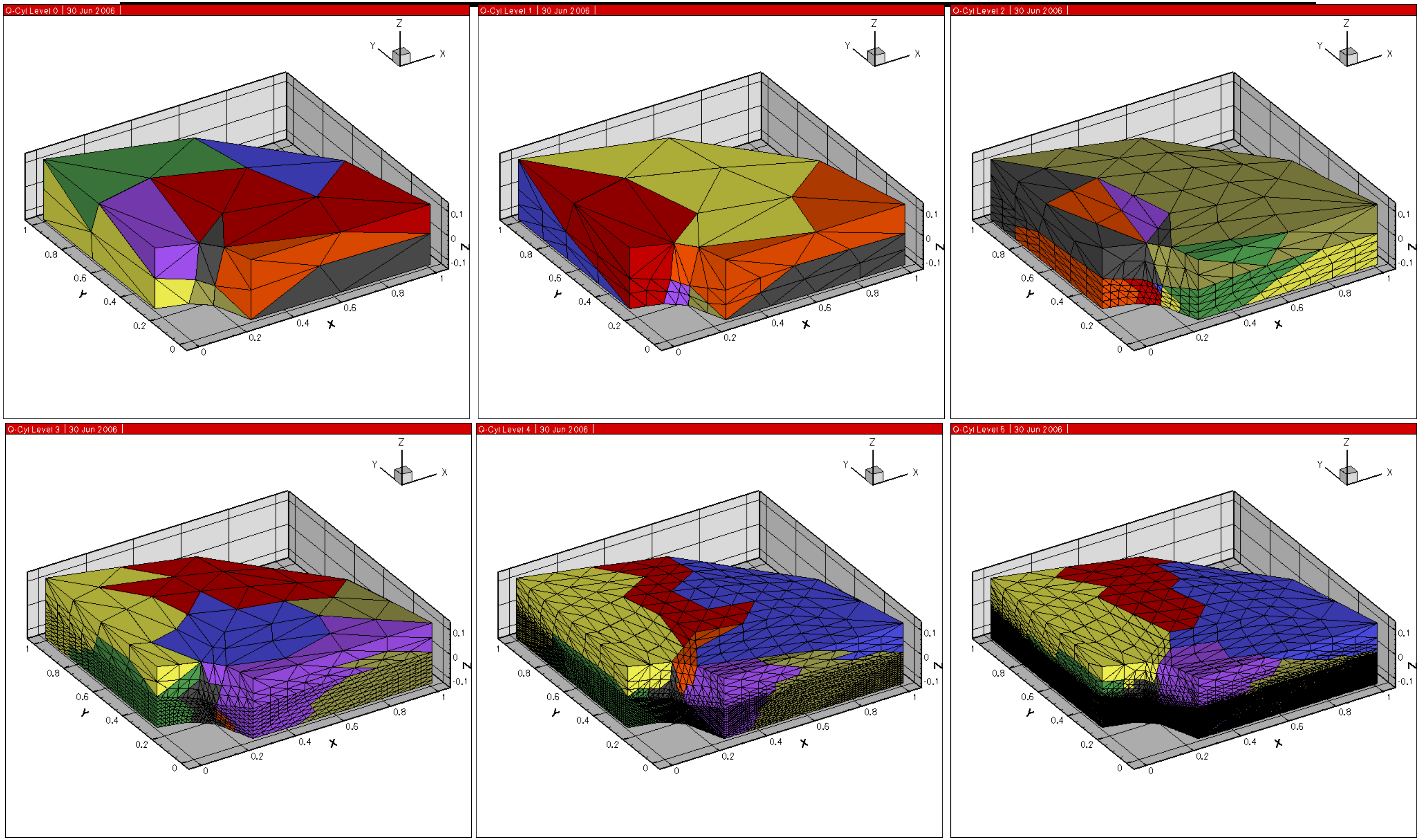
Credits

- John Peterson, CFDLab, University of Texas
- Karen Devine, Richard Drake, and James Overfelt, Sandia, NM
- Henrik Johansson, Uppsala University, Sweden
- Charles Norton, Jet Propulsion Lab, NASA, CA
- Jaideep Ray, Sandia, CA

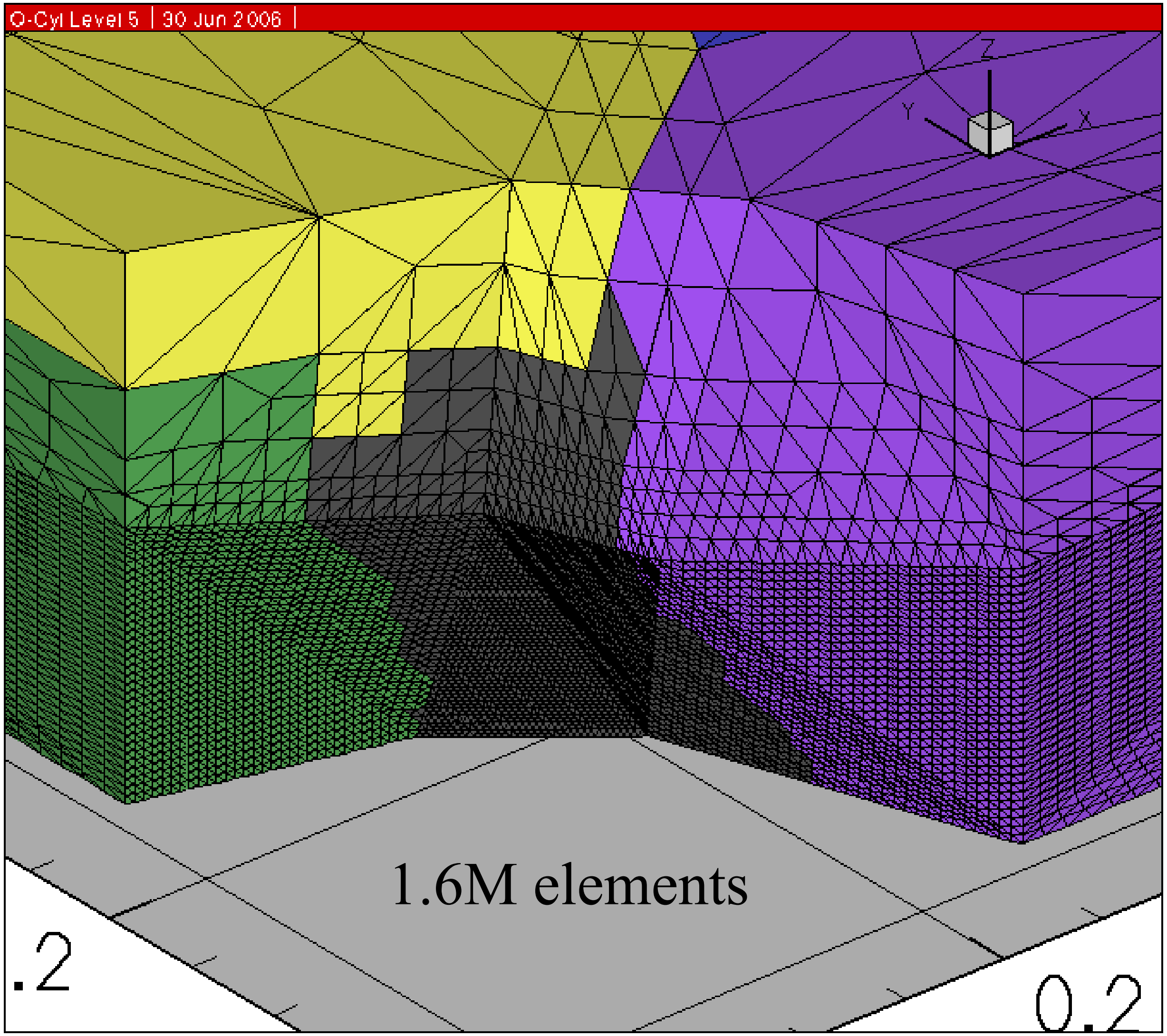
Outline

- Adaptive mesh refinement (AMR)
- Parallel AMR: Performance, scalability, and adaptive partitioning (AP)
- Reducing data migration at algorithm switching: uniform starting points (USP) and switching penalties (SP)
- Hypothesis: AP w/ USP/SP outperforms both static partitioning and “regular” AP
- Experimental setup: Applications, partitioners, the simulator, the cost function, and parameter space
- Results, conclusion and future work

AMR Case Study: Quake [Norton, Jet Propulsion Lab, NASA]



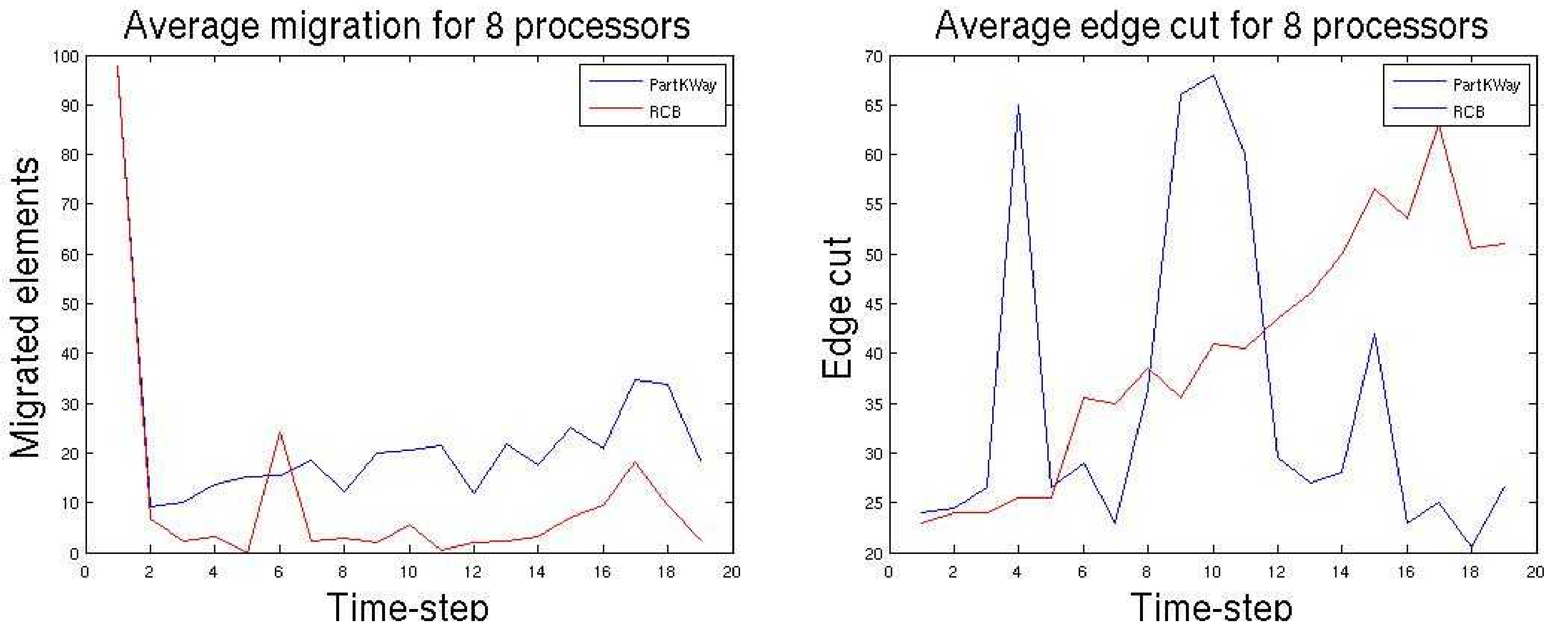
Quake: A Closeup



Data Partitioning and Scalability

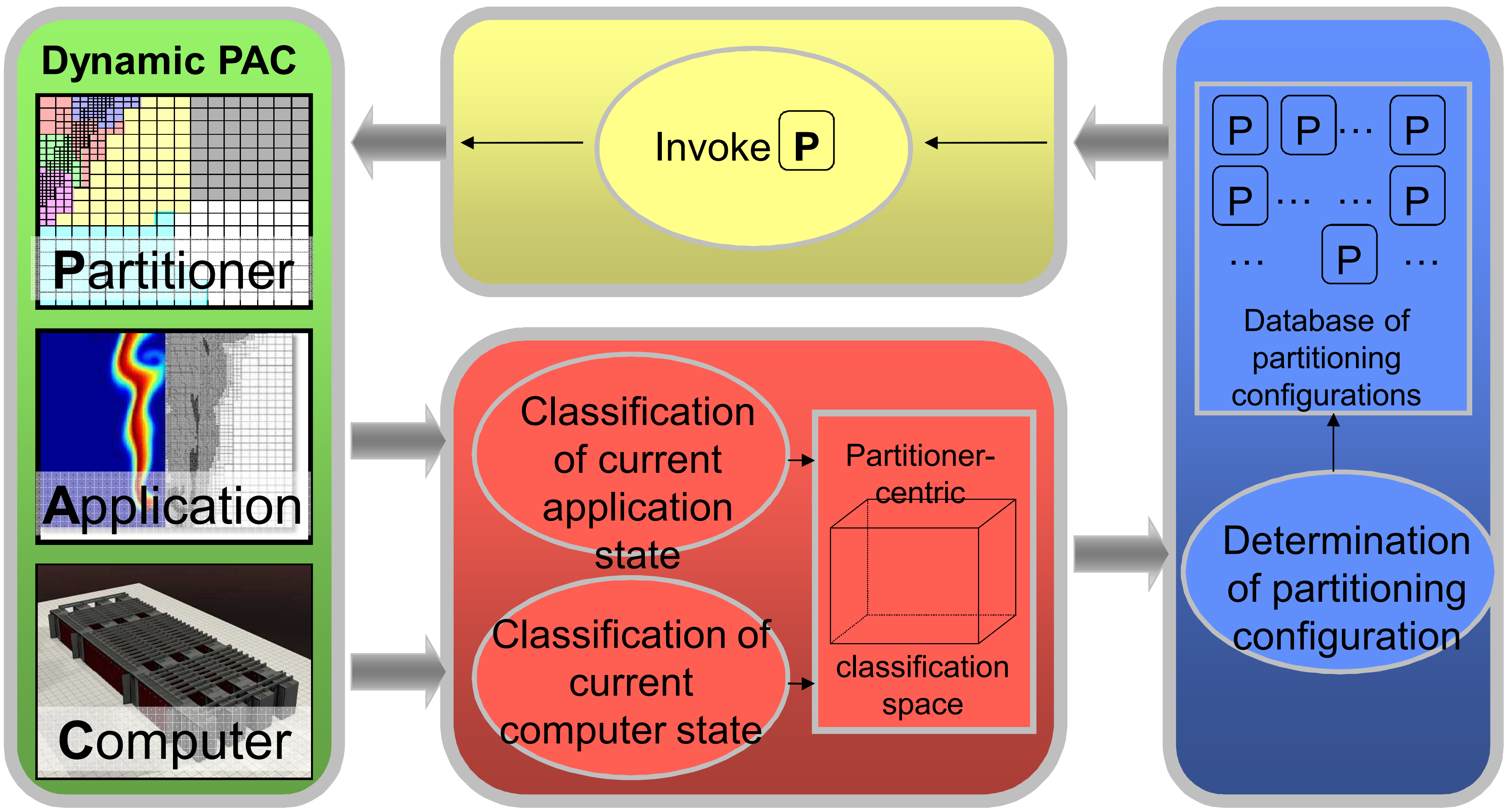
- Solution features drive the mesh dynamics
- As the mesh changes, the partitioning requirements change
- Consider the current run-time state for selecting, configuring, and invoking the most suitable partitioning algorithm →
- Dynamically adaptive partitioning
- Considerable amount of work for *structured* AMR
- How to make effective for *unstructured* AMR?

Example: RCB vs. PartKWay for Laser-Raster



→ To get the best possible parallel efficiency, we need adaptive partitioning

The Meta-Partitioner



Adaptive Partitioning: A Database Approach

Scientific application S with m timesteps: $S = \{s_1, s_2, s_3, \ldots, s_m\}$ sequence of meshes

A set of n partitioning algorithms operating on S:

$$P_1(S) = \{\partial_1^1, \partial_2^1, \partial_3^1, \ldots, \partial_m^1\}$$

$$P_2(S) = \{\partial_1^2, \partial_2^2, \partial_3^2, \ldots, \partial_m^2\}$$

$$\vdots$$

$$P_n(S) = \{\partial_1^n, \partial_2^n, \partial_3^n, \ldots, \partial_m^n\}$$

sequences of partitioned meshes

Quality METRICS

Cost function:

$$\text{Opt}\ (S,P) = \text{index} \min_{\substack{i=1:n \\ j=2:m}} \Pi(\partial_j^i, \partial_{j-1}^i) = \{P_{o_1}, P_{o_2}, P_{o_3}, \ldots, P_{o_m}\}$$

sequence of partitioners

Adaptive partitioning: $A = \{P_{o_1}(s_1), P_{o_2}(s_2), P_{o_3}(s_3), \ldots, P_{o_m}(s_m)\}$

Sandia National Laboratories

Suspicious Metrics (Data Migration)

Scientific application *S* with *m* timesteps: $S = \{s_1, s_2, s_3, \ldots, s_m\}$ sequence of meshes

A set of *n* partitioning algorithms operating on *S*:

$$P_1(S) = \{\partial_1^1, \partial_2^1, \partial_3^1, \ldots, \partial_m^1\}$$

$$P_2(S) = \{\partial_1^2, \partial_2^2, \partial_3^2, \ldots, \partial_m^2\}$$

$$\vdots$$

$$P_n(S) = \{\partial_1^n, \partial_2^n, \partial_3^n, \ldots, \partial_m^n\}$$

sequences of partitioned meshes

Quality METRICS

Cost function:

$$\text{Opt}\ (S,P) = \text{index} \min_{\substack{i=1:n \\ j=2:m}} \Pi(\partial_j^i, \partial_{j-1}^i) = \{P_{o_1}, P_{o_2}, P_{o_3}, \ldots, P_{o_m}\}$$

sequence of partitioners

Adaptive partitioning: $A = \{P_{o_1}(s_1), P_{o_2}(s_2), P_{o_3}(s_3), \ldots, P_{o_m}(s_m)\}$

Sandia National Laboratories

Data Migration Problem for Adaptive Partitioning

- Partitioning algorithms fundamentally different→
- Their native mapping of data onto processors might be fundamentally different→
- At run-time, switching to a theoretically superior algorithm, might incur substantial data migration

Remedies

- Uniform starting point (USP)
- Switching penalties (SP)

Uniform Starting Point (USP)

Create more predictable sequence differences by

a) Selecting a scratch/remap technique P_k

b) Forcing $\partial_1^1 = \partial_1^2 = \partial_1^3 \ldots = \partial_1^n = \partial_1^k$

A set of *n* partitioning algorithms operating on *S*:

$$P_1(S) = \{\partial_1^k, \partial_2^1, \partial_3^1, \ldots, \partial_m^1\}$$

$$P_2(S) = \{\partial_1^k, \partial_2^2, \partial_3^2, \ldots, \partial_m^2\}$$

$$P_n(S) = \{\partial_1^k, \partial_2^n, \partial_3^n, \ldots, \partial_m^n\}$$

sequences of partitioned meshes

→ Greater probability for ∂_i^t and ∂_j^t being more similar

Switching Penalties (SP)

Original function for estimating the cost of invoking partitioner i at timestep t:

$$\Pi(\partial_i^t, \partial_j^{t-1}) = \mathrm{CCR} \times \mathrm{loadimb}(\partial_i^t) + \mathrm{ITR} \times \mathrm{edgec}(\partial_i^t) + \mathrm{migr}(\partial_i^t, \partial_j^{t-1})$$

P^I = Set of incremental algorithms

P^S = Set of scratch/remap algorithms

$$F(f) = \begin{cases} f \text{ for } P_i^I \rightarrow P^S \\ 1 \text{ otherwise} \end{cases}$$

New cost function:

$$\Pi(\partial_i^t, \partial_j^{t-1}) = \mathrm{CCR} \times \mathrm{loadimb}(\partial_i^t) + \mathrm{ITR} \times \mathrm{edgec}(\partial_i^t) + F(f) \times \mathrm{migr}(\partial_i^t, \partial_j^{t-1})$$

Uniform Starting Point + Switching Penalties

Scientific application S with m timesteps: $S = \{s_1, s_2, s_3, \ldots, s_m\}$ sequence of meshes

A set of n partitioning algorithms operating on S:

$$P_1(S) = \{\partial_1^k, \partial_2^1, \partial_3^1, \ldots, \partial_m^1\}$$

$$P_2(S) = \{\partial_1^k, \partial_2^2, \partial_3^2, \ldots, \partial_m^2\}$$

$$P_n(S) = \{\partial_1^k, \partial_2^n, \partial_3^n, \ldots, \partial_m^n\}$$

sequences of partitioned meshes

Quality METRICS

Cost function:

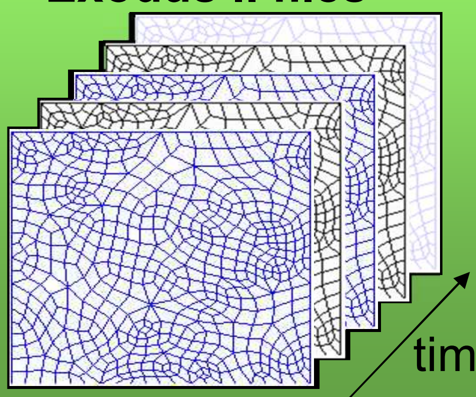
$$\text{Opt}\ (S,P) = \text{index} \min_{\substack{i=1:n \\ j=2:m}} \Pi(\partial_j^i, \partial_{j-1}^i) = \{P_{o_1}, P_{o_2}, P_{o_3}, \ldots, P_{o_m}\}$$

sequence of partitioners

Adaptive partitioning: $A = \{P_{o_1}(s_1), P_{o_2}(s_2), P_{o_3}(s_3), \ldots, P_{o_m}(s_m)\}$

The Parallel Mesh Application Simulator

Adaptive mesh as a sequence of *serial* Exodus II files



Exodus reader

First timestep?

Yes!

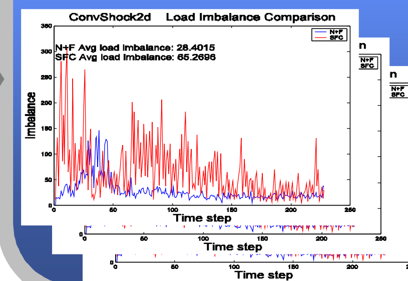
No!

Pre-partition with respect to current partitioning

Repartition with the settings of choice

Derive metrics:
- Data migration
- Edge cut
- Load imbalance
- Border elements
- Partitioning time

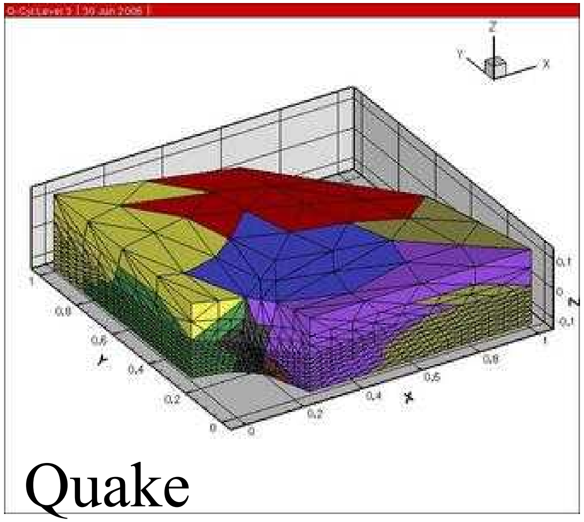
Statistics data files and plots

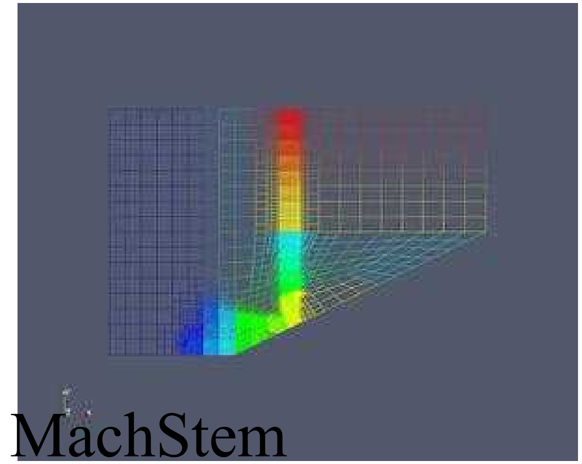


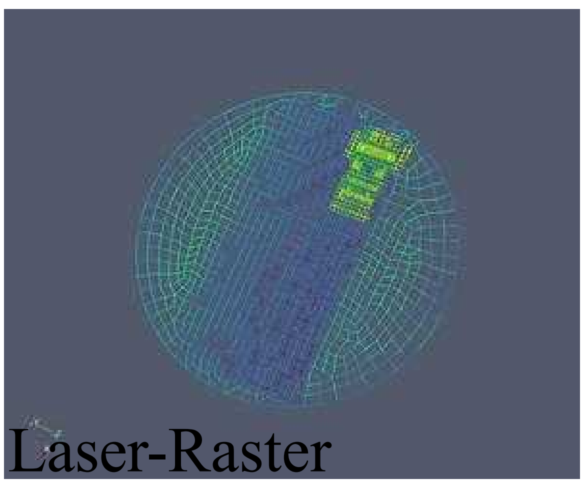
Zoltan:
- ParMetis (…)
- RCB
- RIB
- HSFC

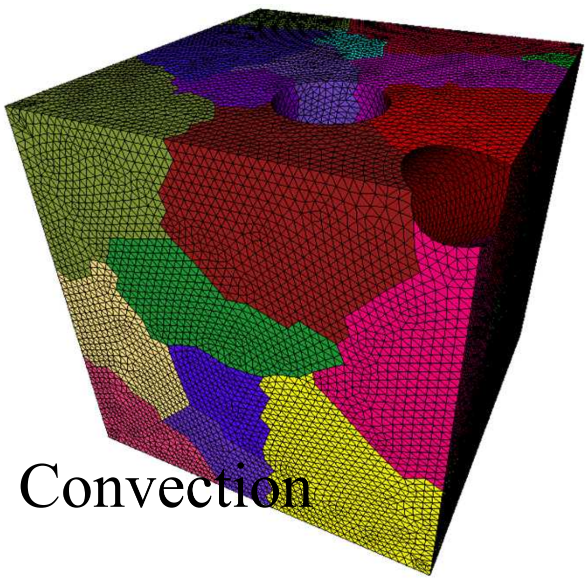
Partitioning parameters

Real-World AMR Applications









Sandia National Laboratories

Application Specifics

Applic.	Elmnt	Dim	Steps	Avg *E*	Max *E*
Quake	Tetra	3	6	308K	1.6M
Mach-Stem	Quad	2	109	8.2K	13K
Laser-Raster	Cube	3	65	4.4K	10K
Conv.	Tetra	3	73	87K	94K

Sandia National Laboratories

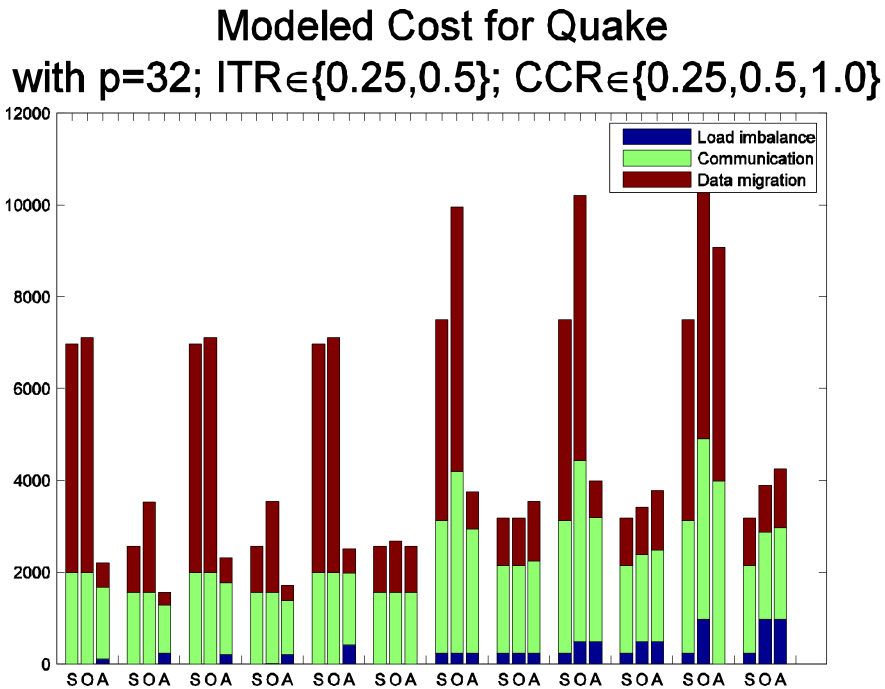
The Partitioners

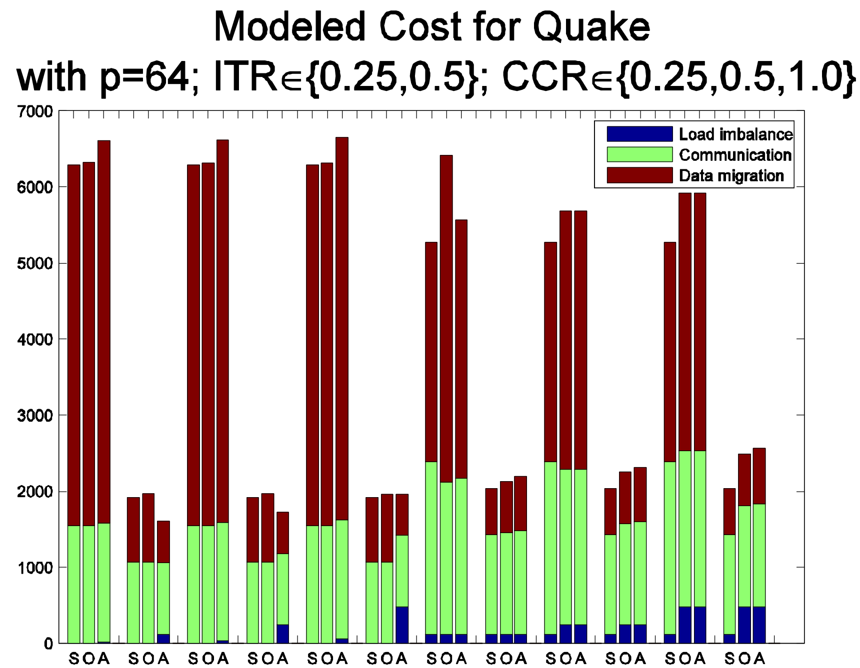
- Need partitioners from different classes with fundamentally different properties:
- RCB: Zoltan native; geometric
- RCB+Remap: Zoltan native; geometric; scratch/remap
- AdaptiveRepart: ParMetis; graph-based; incremental or scratch/remap (adaptive)

Experimentation

- **Hypothesis: Adaptive partitioning with USP/SP is beneficial within our experimental parameters**
- **Measure partitioner impact with cost function**
- **Chose parameters for "equal contribution" →**
- ***f*=1; 2; 4; ITR=0.25; 0.5; CCR=0.25; 0.5; 1.0;**
- **Two sets of data; two sets of #*p***
- **Static: RCB, RCB+Remap, AdaptiveRepart →**
- **Opt(*f*) (min cost for each timestep for *f*) →**
- **Adaptive(*f*) (true adaptive for *f*, determined by Opt)**

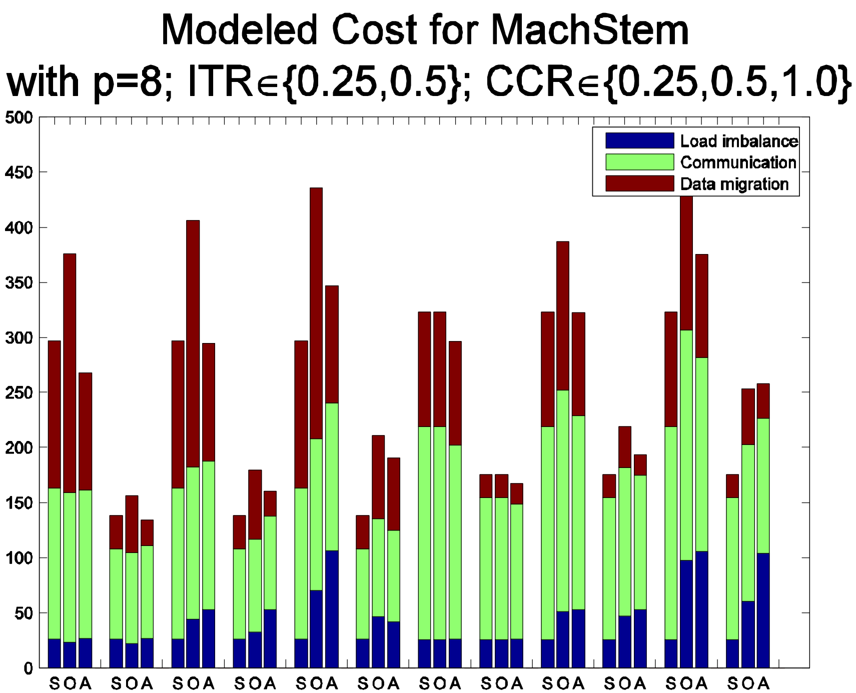
Results: Quake

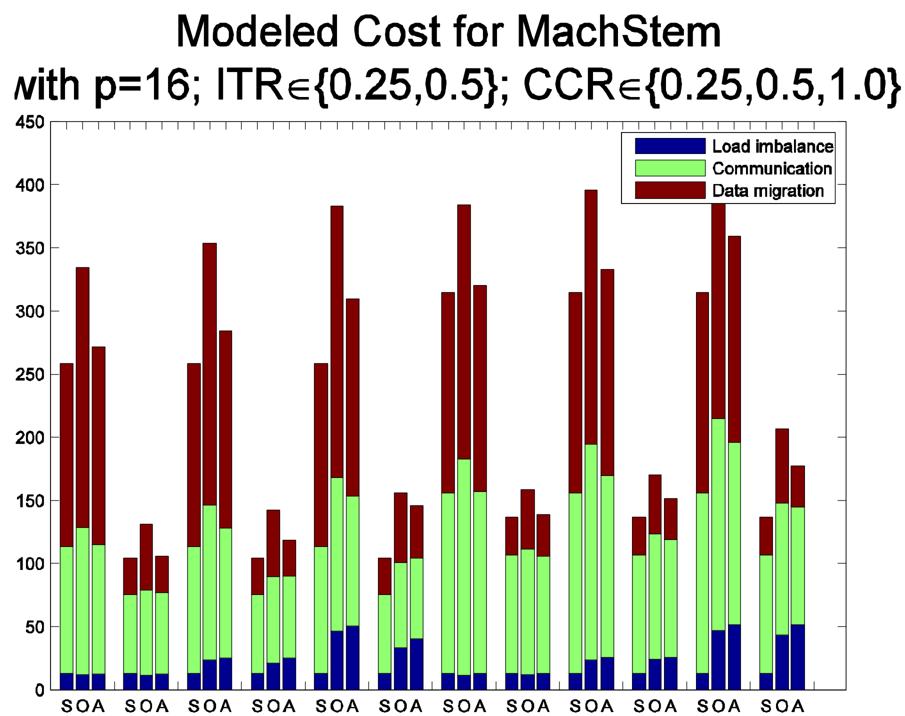




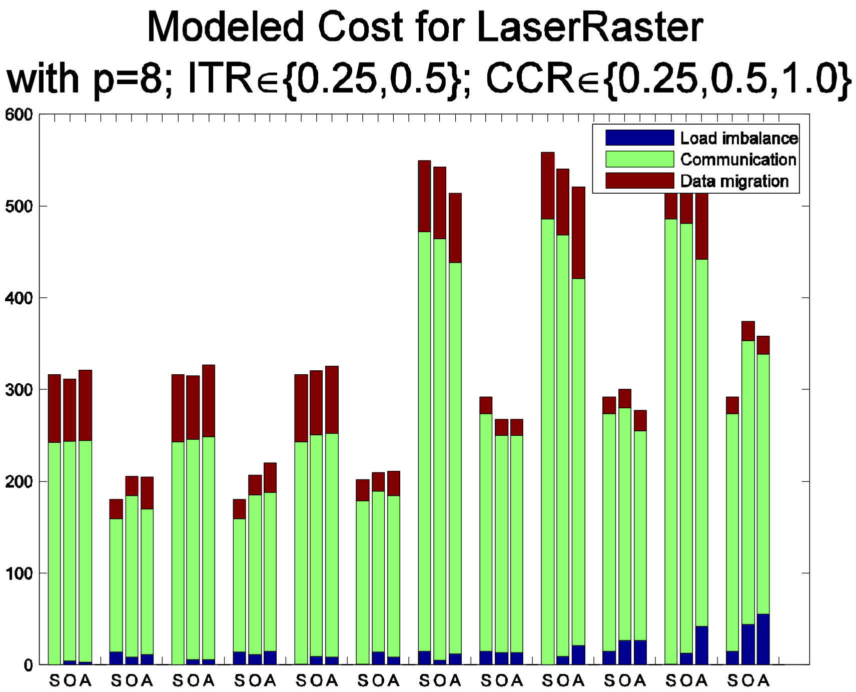
Sandia National Laboratories

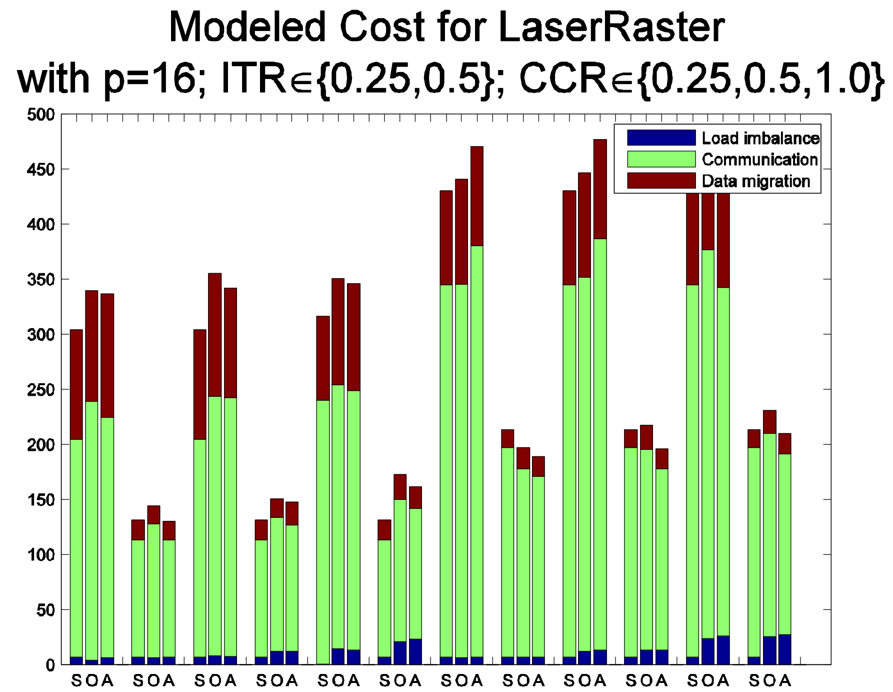
Results: MachStem





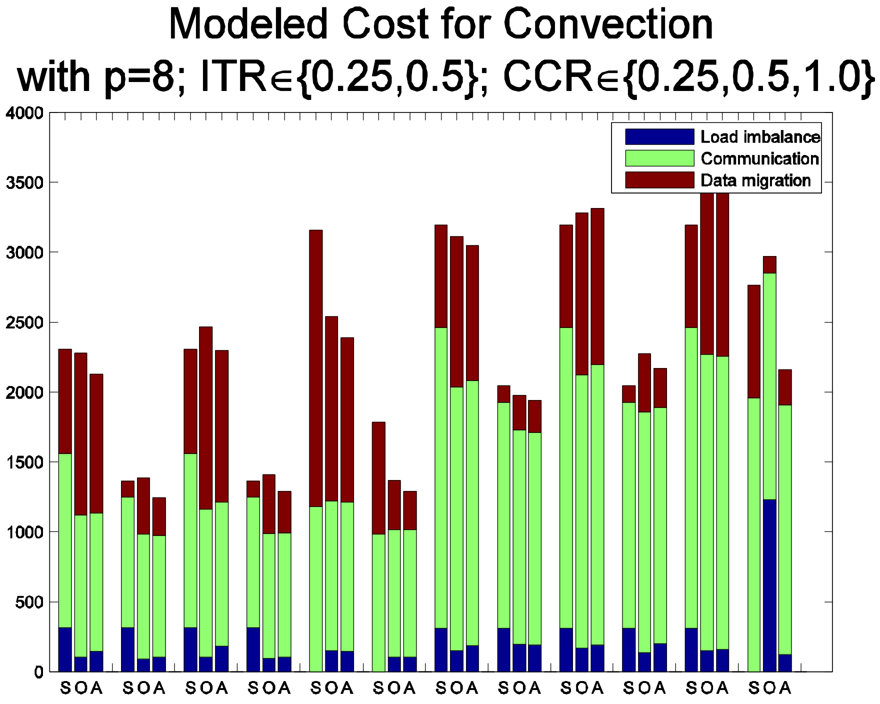
Results: Laser-Raster

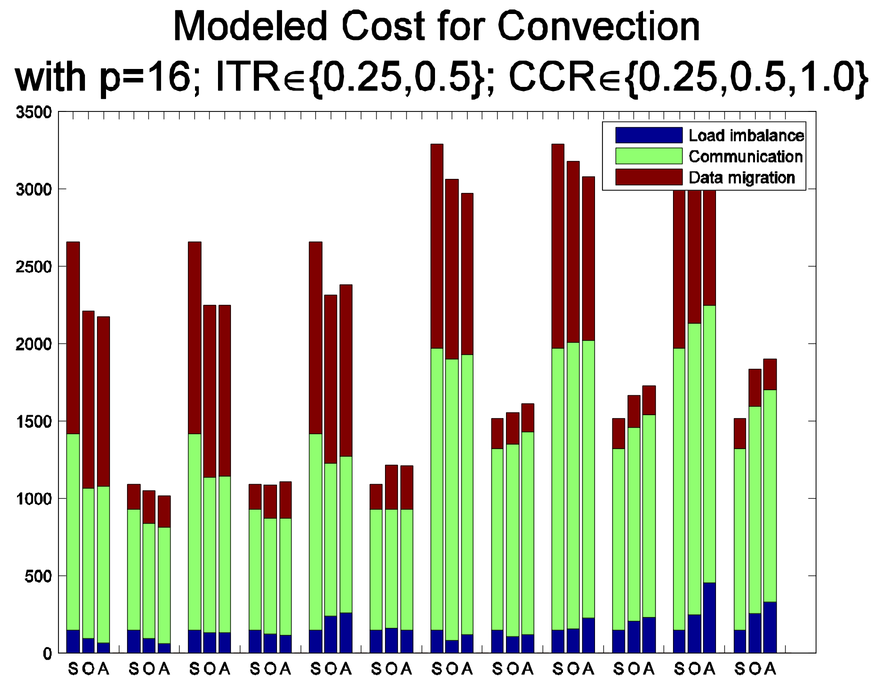




Sandia National Laboratories

Results: Convection





Sandia National Laboratories

Results: Summary

- Adaptive partitioning with USP/SP outperformed the best static algorithm in about 40% of the experiments
- AP with USP/SP generally improved on “regular” AP (about 75% of the experiments).
- Adaptive partitioning performs slightly better for 8 (32) processors than for 16 (64).
- No clear correlation between CCR, ITR, and *f* compared to the effectiveness of the AP.

Sandia National Laboratories

Conclusions and Future Work

- AP seems applicable in a variety of (hard to define) AMR conditions
- Future work includes (a) finding better ways to estimate data migration due to different native data mappings, and (b) gaining a better understanding of conditions beneficial for AP
- New hypothesis: Smarter migration estimation & leveraging better understanding of conditions → AP beneficial for vast gamut of application-computer combinations

Questions?

- Read more on http://hpcn.sandia.gov/~jsteens