Parallel Runtime Environments with Cloud Database: Performance Study for HMM with Adaptive Sampling

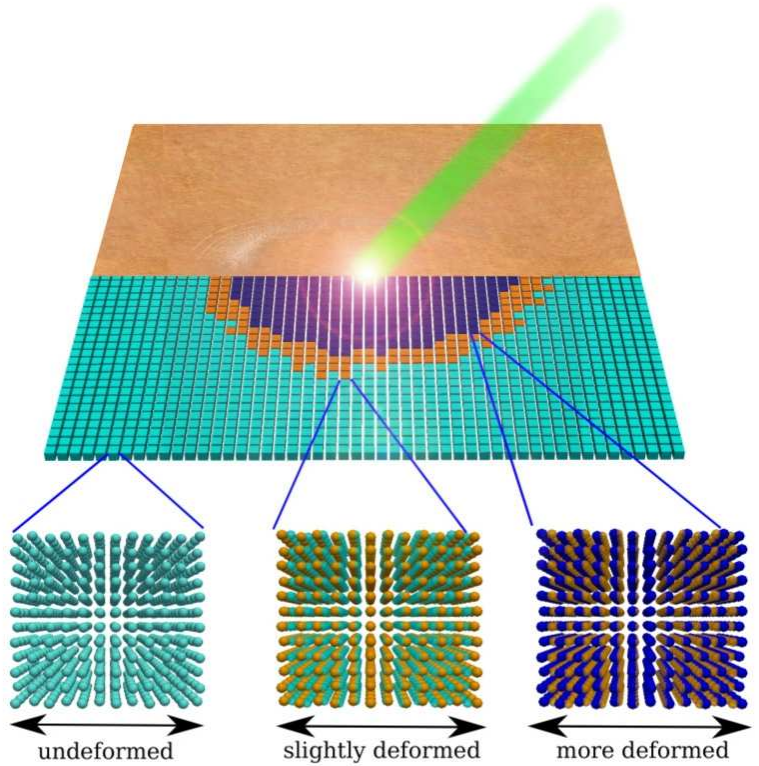
D. Roehm, R. S. Pavel, T. C. Germann, A. L. McPherson and C. Junghans

Los Alamos National Laboratory
NM, USA

May 8, 2015

Motivation

- Material modeling: time and length scale challenge
- Micro-structure matters, but is computationally expensive
- HMM: Combination of macro- and micro-scale simulations
- Adaptive sampling techniques: take micro-structure into account when necessary
- Prediction (kriging) based on database values instead of executing MD simulation



Model problem: Laser impact on a copper plate

Elastodynamics

- Non-oscillatory central scheme (predictor-corrector)
- Continuum mechanics $\Rightarrow$ conservation PDEs in Lagrangian coordinates
- Evolution of deformation, momentum and energy density computed by finite volume solver
- Stress and energy flux evaluated with MD

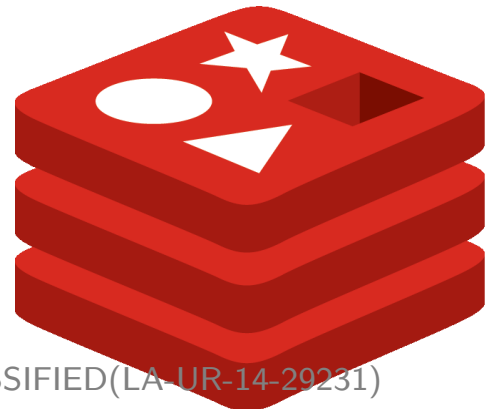
On Macro level: Conservation laws for mass, momentum, and energy:

$$\rho \partial_t \mathbf{A} - \bigtriangledown \mathbf{q} = 0$$
$$\partial_t \mathbf{q} - \bigtriangledown \cdot \tau = 0$$
$$\partial_t e + \bigtriangledown \cdot \mathbf{j} = 0$$

On Micro level: Take strain, momentum density, and energy density and return stress, momentum, and energy density flux.

Database: Redis

- Key-value storage
- High performance
- Support for distribution/cluster mode
- NoSQL Redis: open-source, networked, in-memory, key-value data store
- Users of Redis: GitHub, Twitter, Stackoverflow, Craigslist, ... (info: http://redis.io)
- Locality-aware hashes: a range along all seven dimensions of our conserved vector
- Truncated hash based upon the specified range
- Sort values by distance to requested input



redis

Kriging

- "Optimally predicting", originated in geostatistics
- Prediction of $Z(\mathbf{s}_0)$ at a certain position in the high dimensional space by computing a weighted average of the known vectors in the neighborhood of the point

$$Z'(\mathbf{s}_0) = \sum_{i=1}^{n} \lambda_i Z(\mathbf{s}_i)$$

  at location $\mathbf{s}_0$ that minimizes the mean-square error

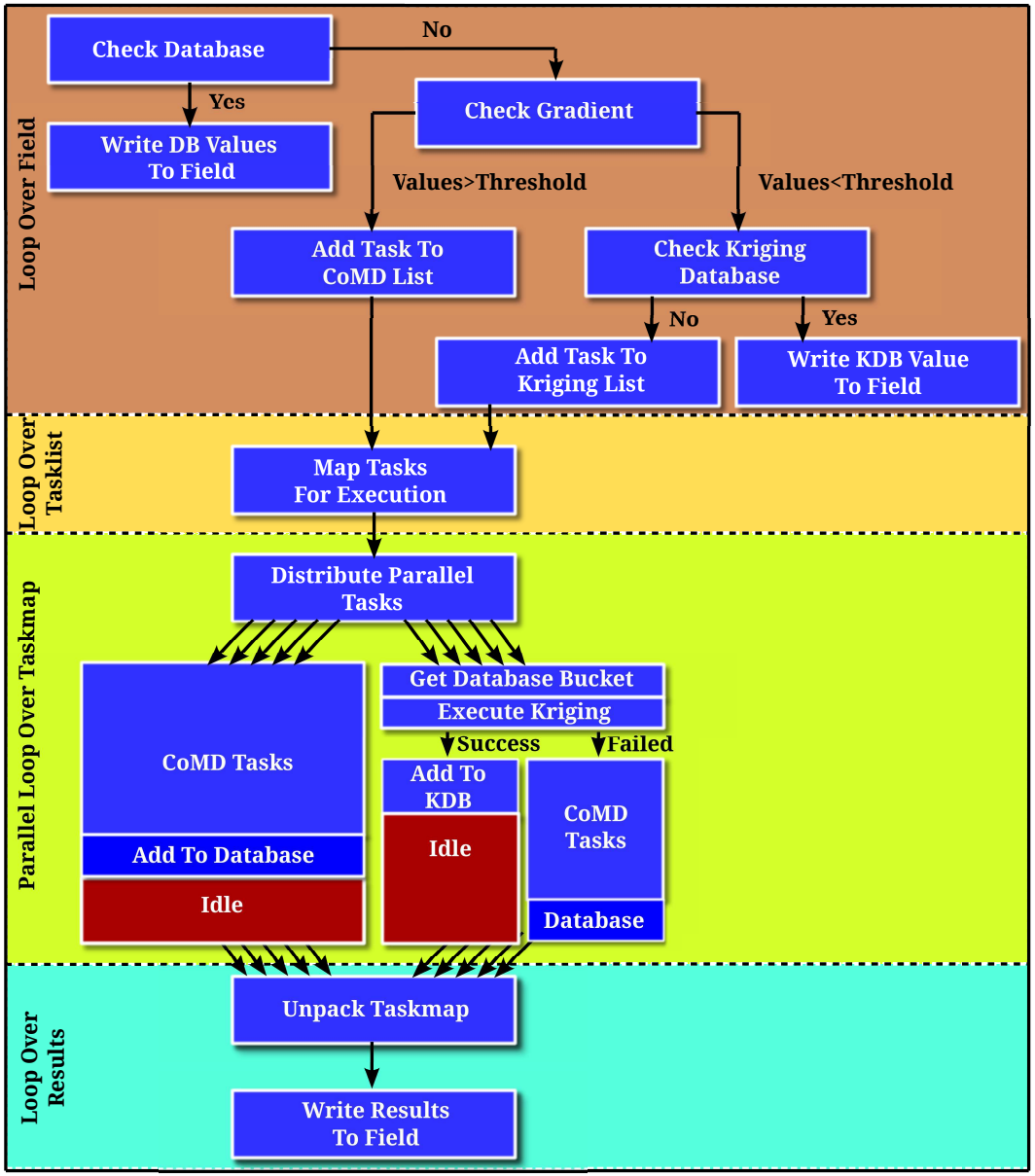
$$E[Z(\mathbf{s}_0) - Z'(\mathbf{s}_0)]^2.$$

- Calculates an error of the prediction at the same time
- Store simulation results in key-value database

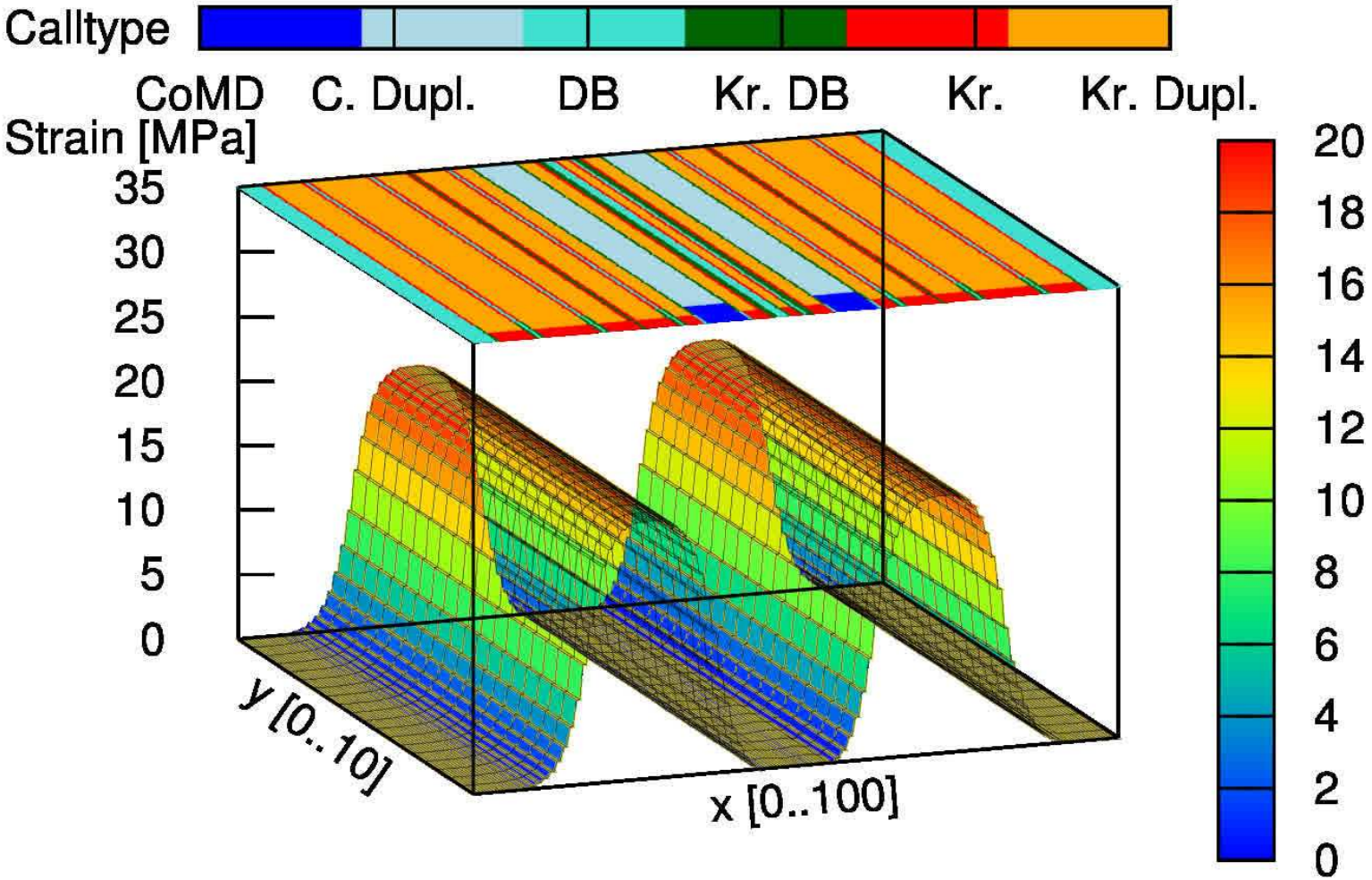
Implementation

- Macrosolver frameworks (github.com/exmatex/CoHMM)
  - Charm++ 6.6.0 (University of Illinois: charm.cs.uiuc.edu)
  - Intel CnC 1.0.002 (Intel: icnc.github.io)
  - OpenMP
  - Libcircle v0.2.1 (github.com/hpc/libcircle)
- MD miniapp CoMD (github.com/exmatex/CoMD)
  - serial C Version
- Compilers and Libraries:
  - GCC 4.8.x
  - ICPC 15.0.0
  - Boost 1.55
  - Blas and Lapack

Schematics

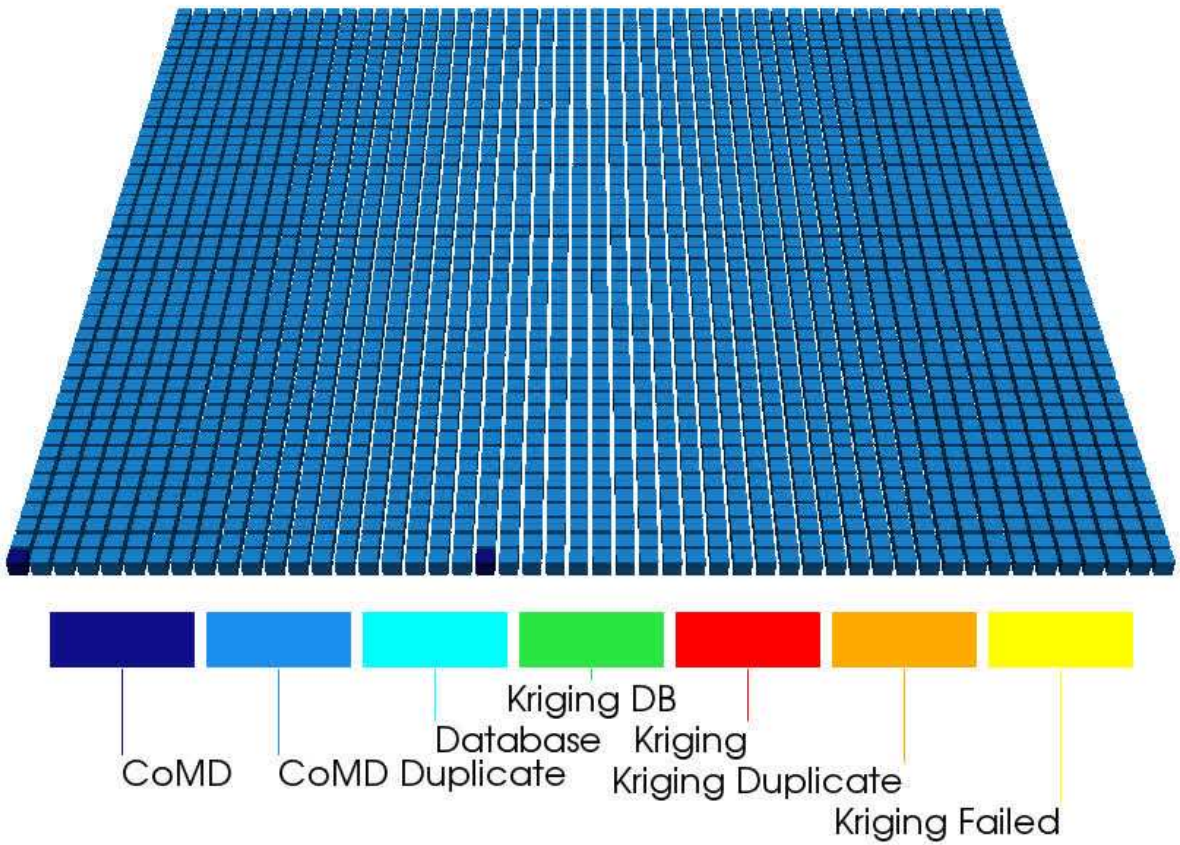

Loop Over Field
Check Database
No
Yes
Write DB Values To Field
Check Gradient
Values>Threshold
Values<Threshold
Add Task To CoMD List
Check Kriging Database
No
Yes
Add Task To Kriging List
Write KDB Value To Field
Loop Over Tasklist
Map Tasks For Execution
Parallel Loop Over Taskmap
Distribute Parallel Tasks
CoMD Tasks
Add To Database
Idle
Get Database Bucket
Execute Kriging
Success
Failed
Add To KDB
Idle
CoMD Tasks
Database
Loop Over Results
Unpack Taskmap
Write Results To Field

Flat Wave Test Case



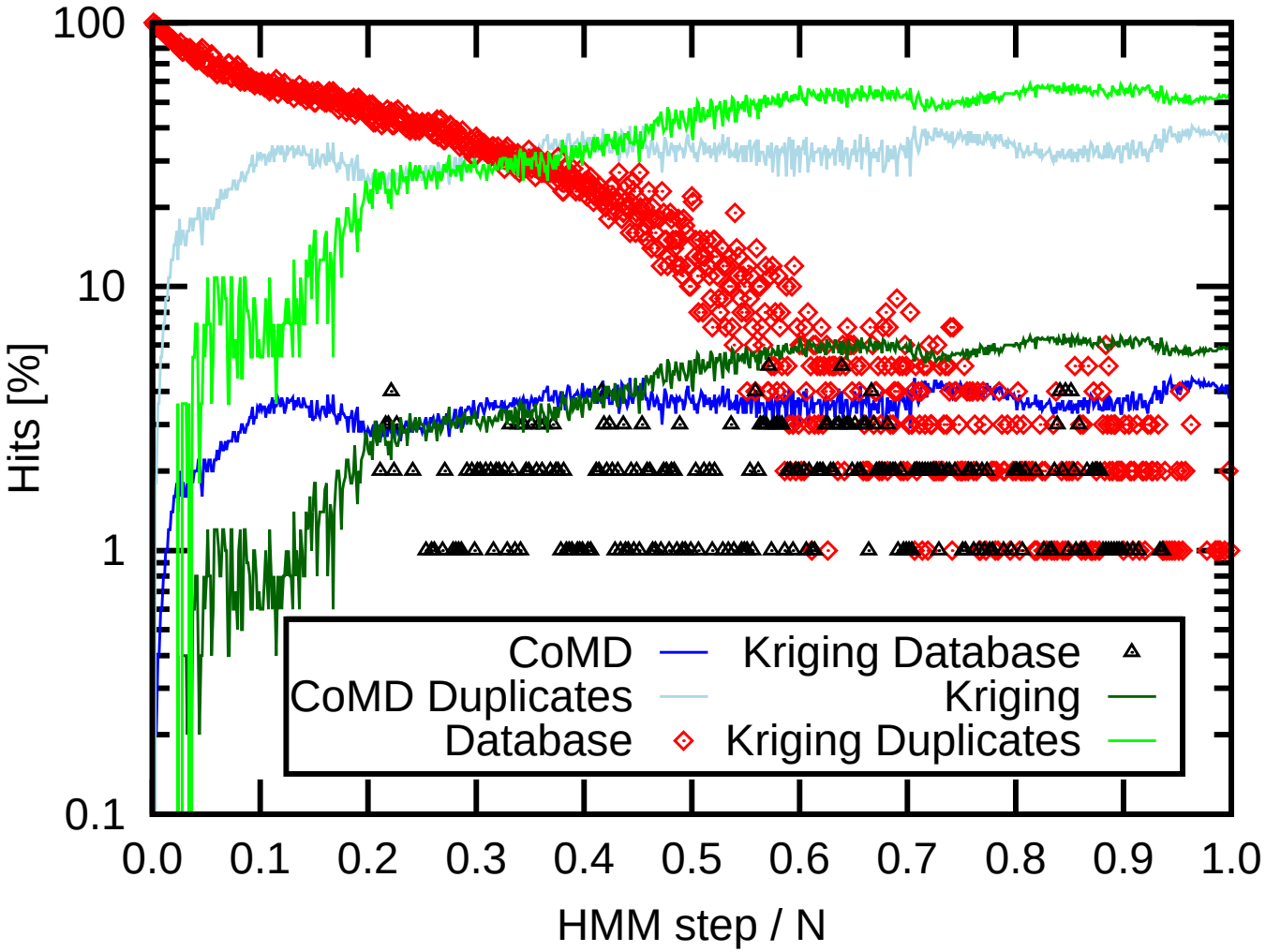
Color bar right hand side: Strain. Color bar top: Type of call.

Flat Wave Test Case



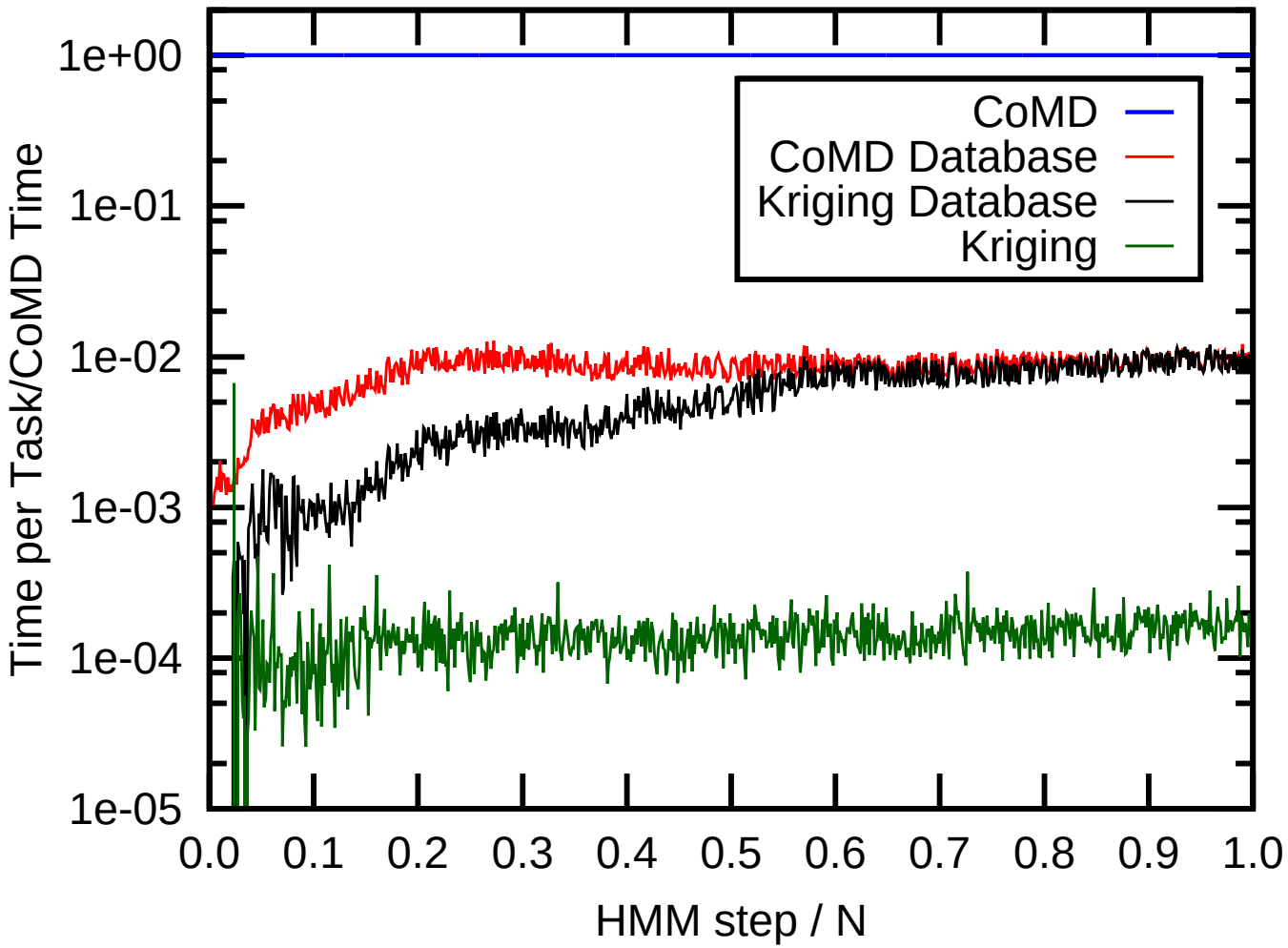
Colors: Type of call.

Adaptive Sampling Performance



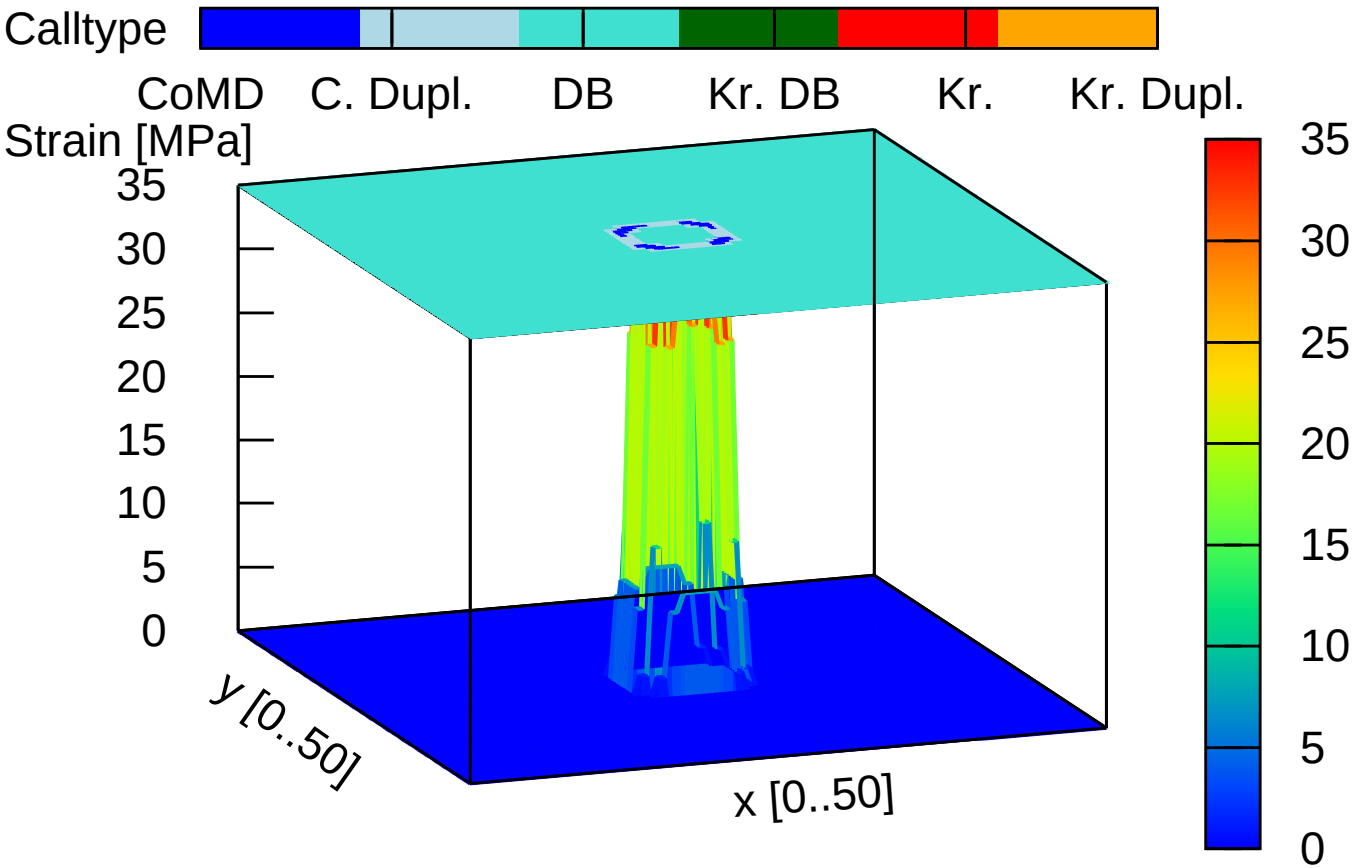
Overall less than 5% of CoMD calls ⇒ speedup of 25

Adaptive Sampling Performance: Flat Wave

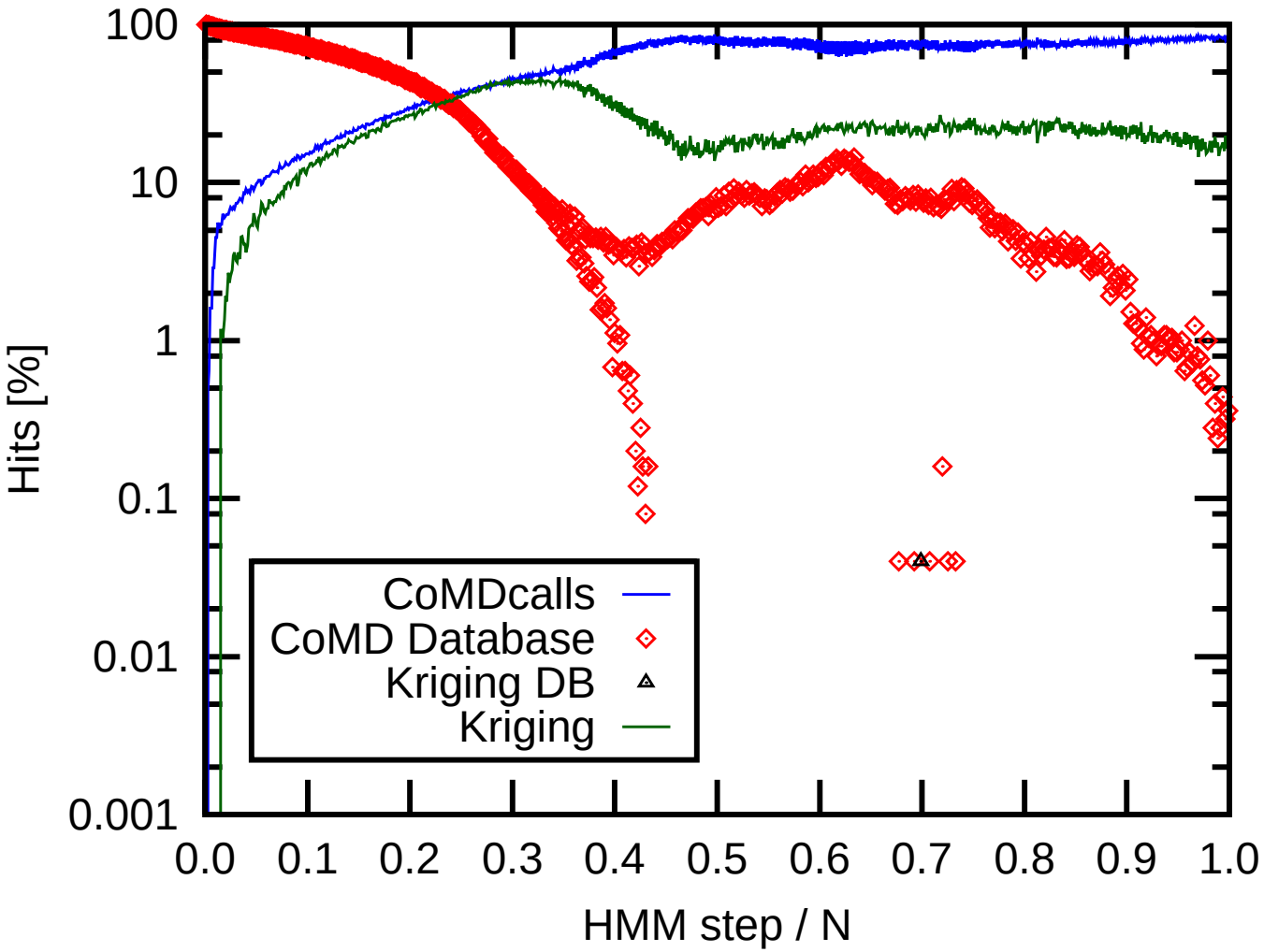


Absolute time per task

Circular Impact Test Case

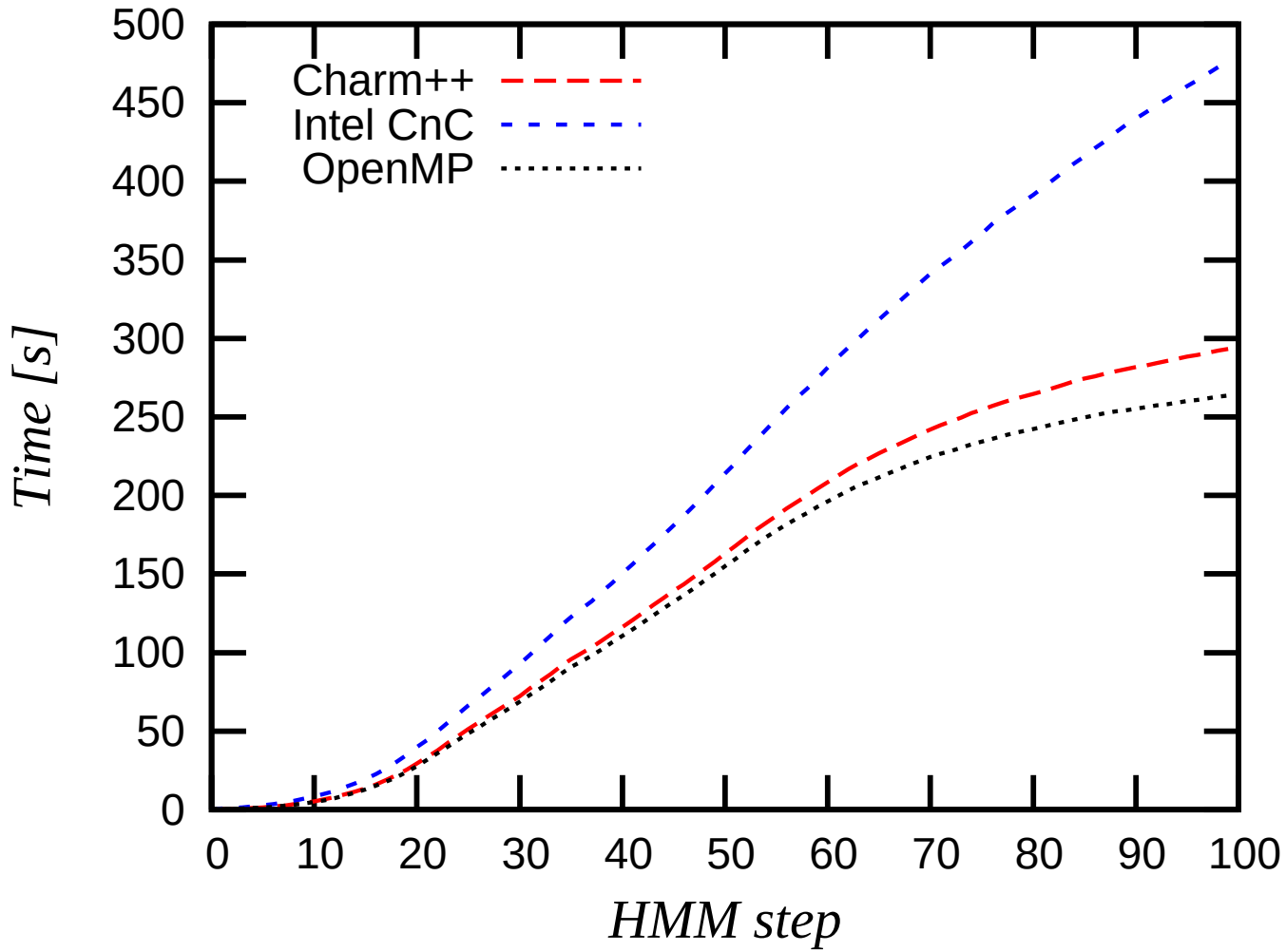


Adaptive Sampling Performance: Circular Impact



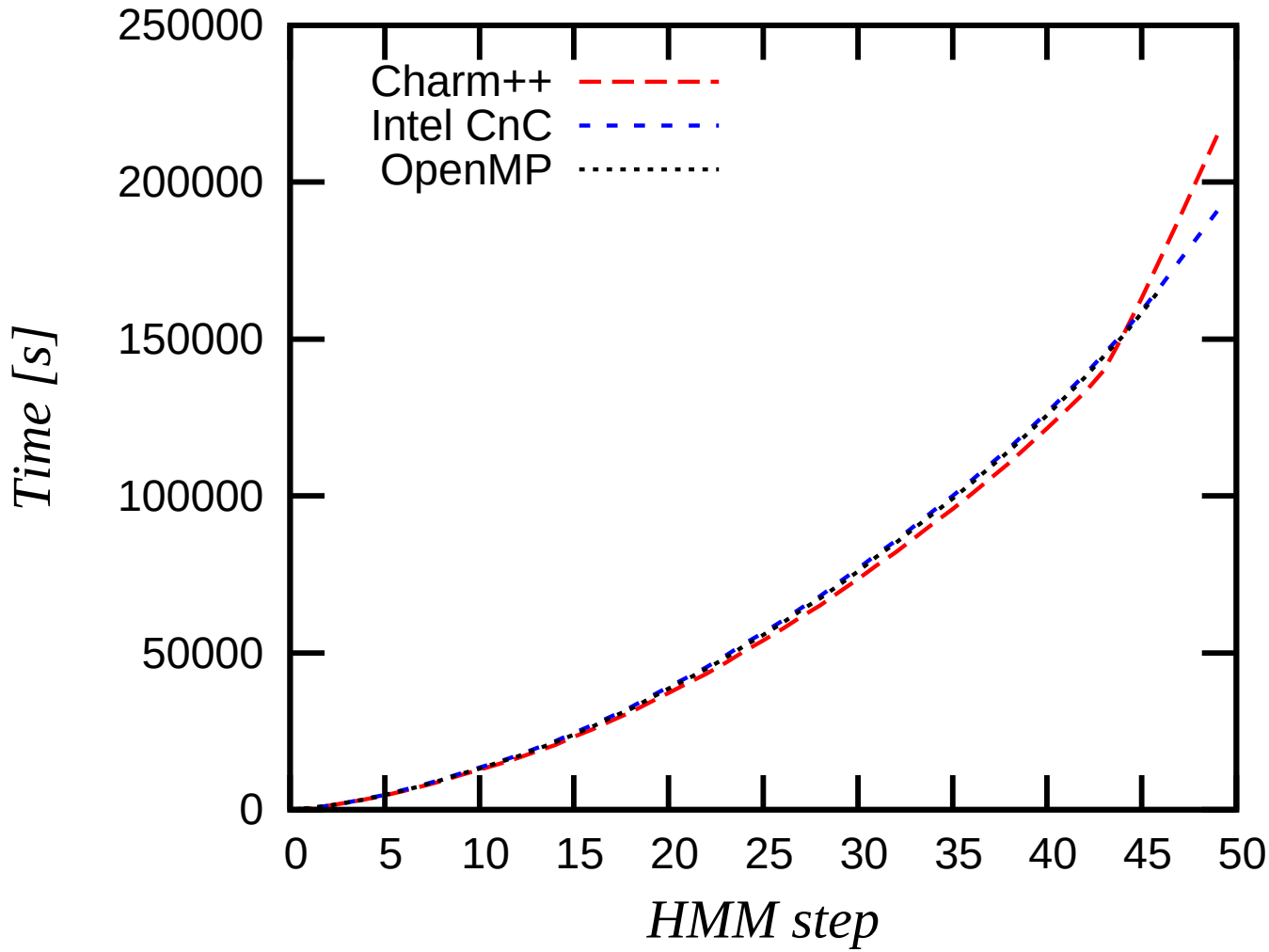
Save approx. 10% of calls long term ⇒ speedup of 2.5

Framework Performance: Circular Impact Analytic



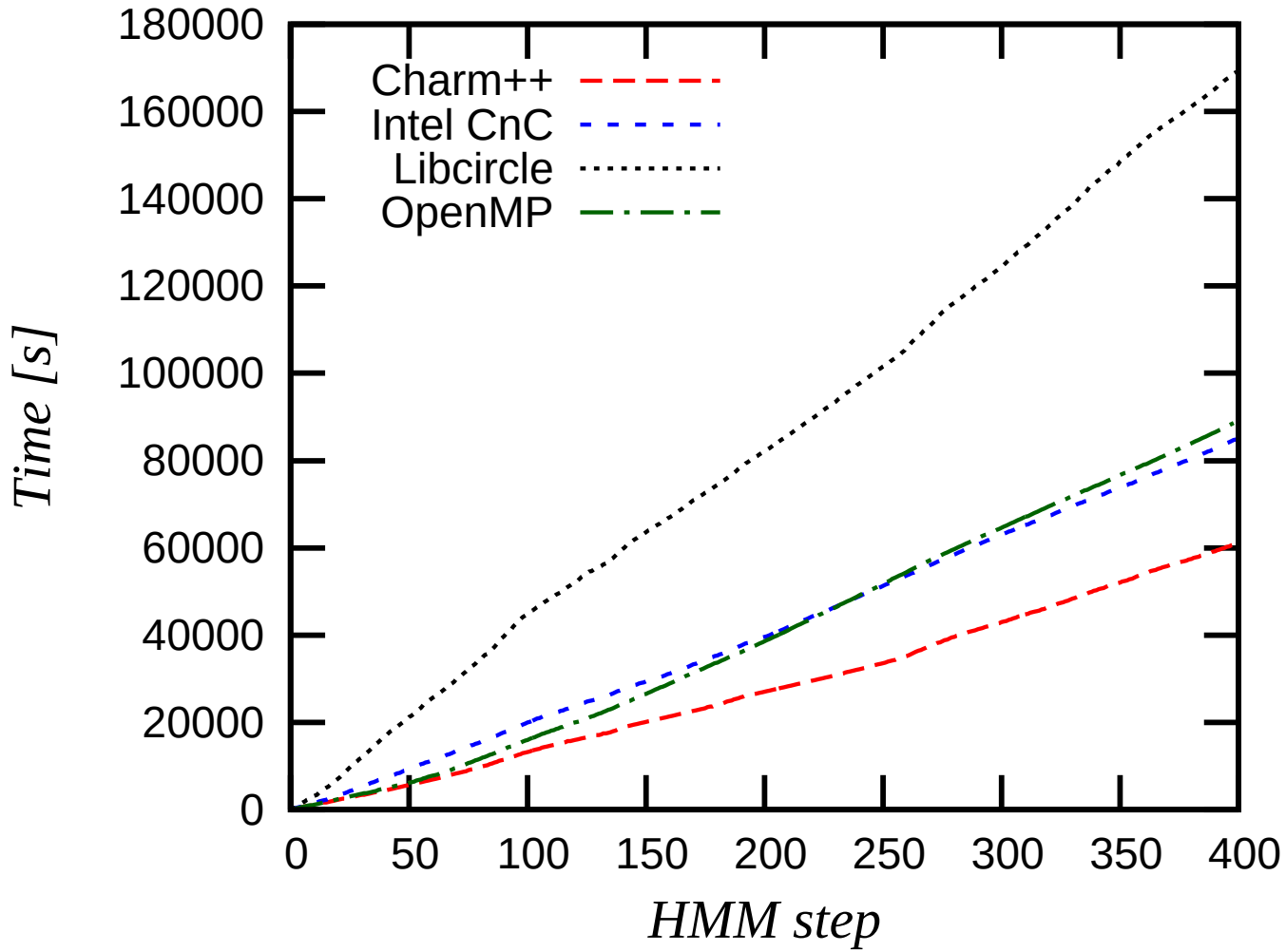
16 cores single database shared memory

Framework Performance: Circular Impact



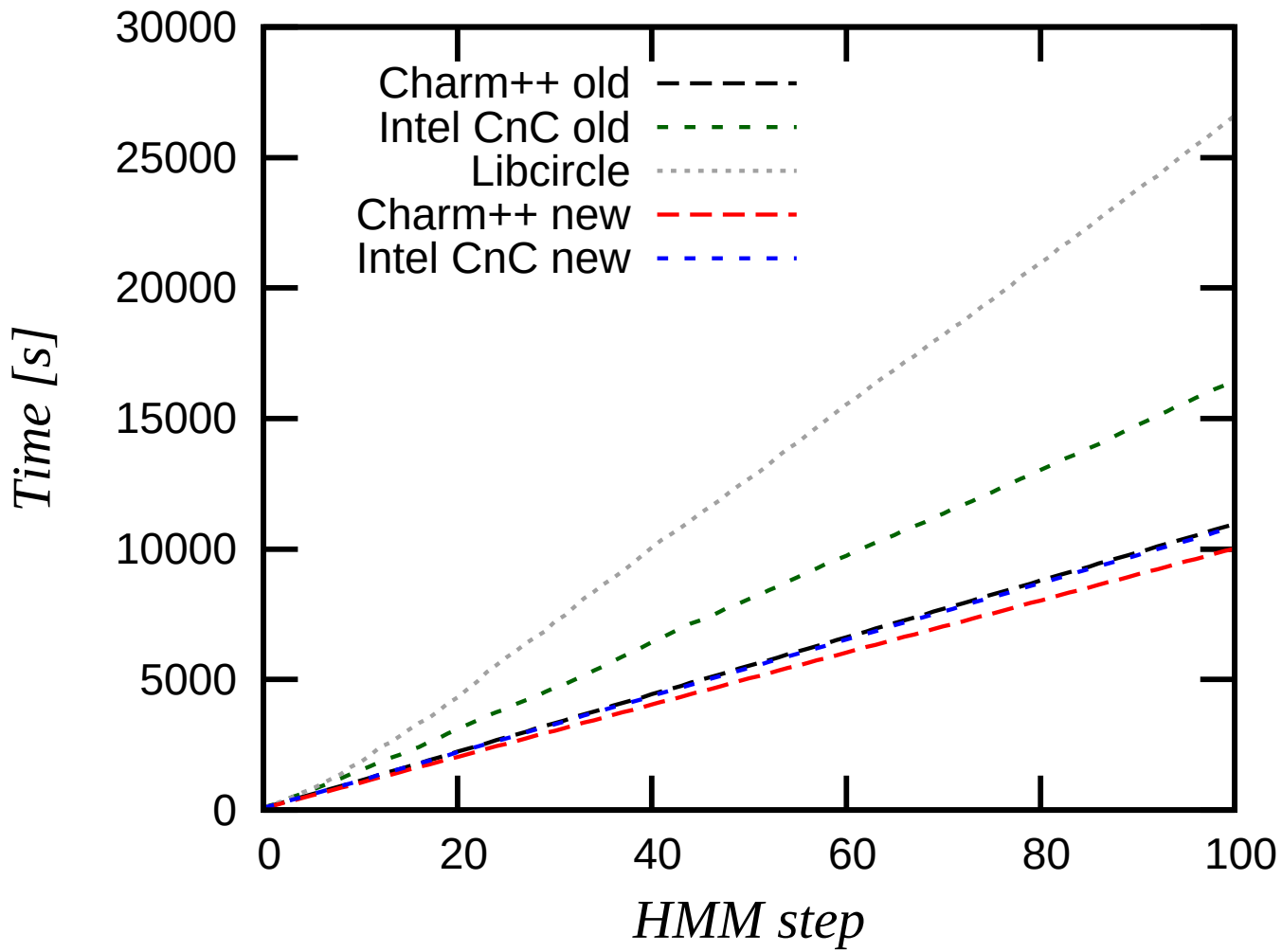
16 cores single database shared memory

Framework Performance: Flat Wave



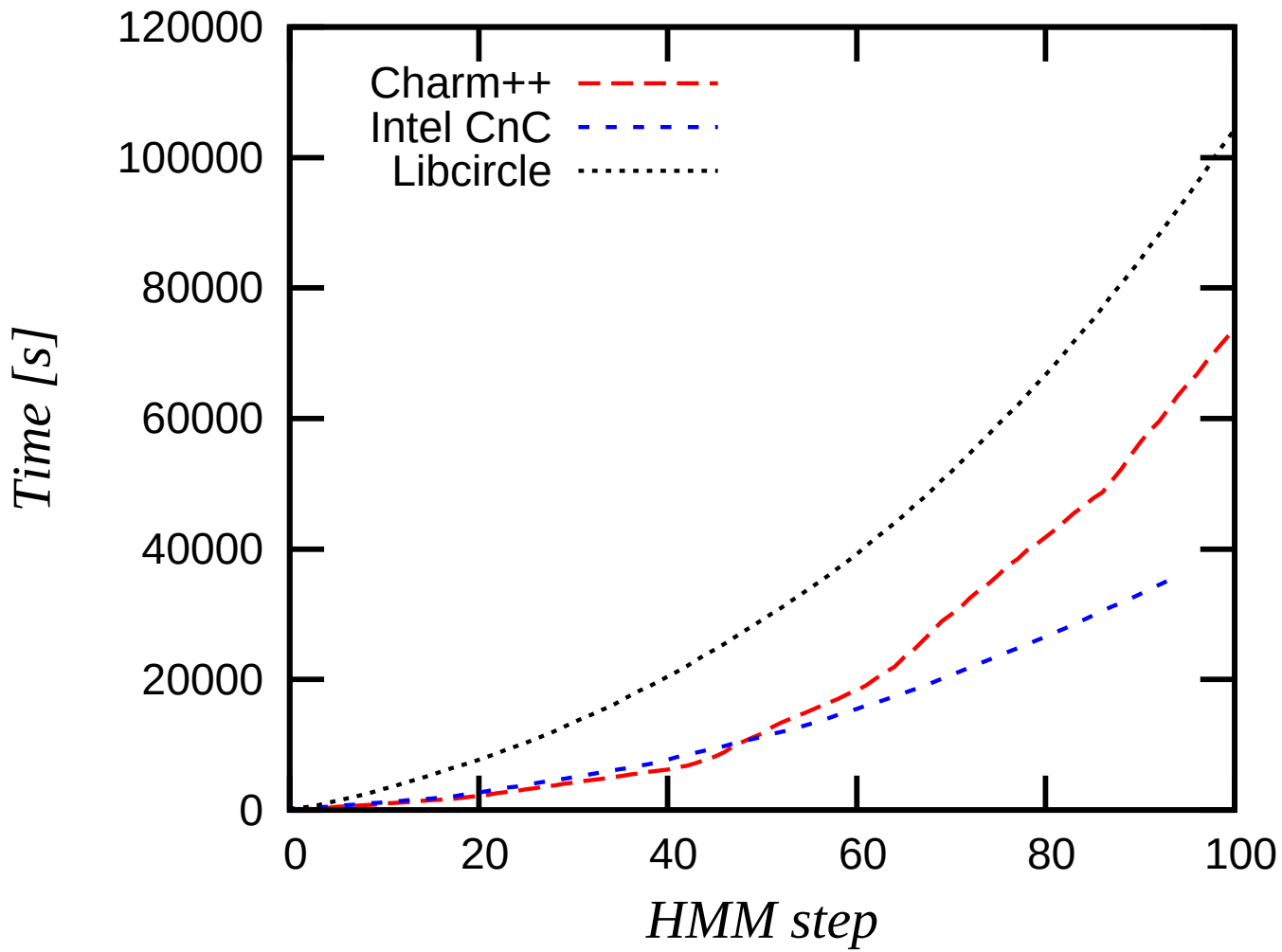
48 cores single database shared memory

Framework Performance: Flat Wave



144 cores single database

Framework Performance: Circular Impact



480 cores with single database

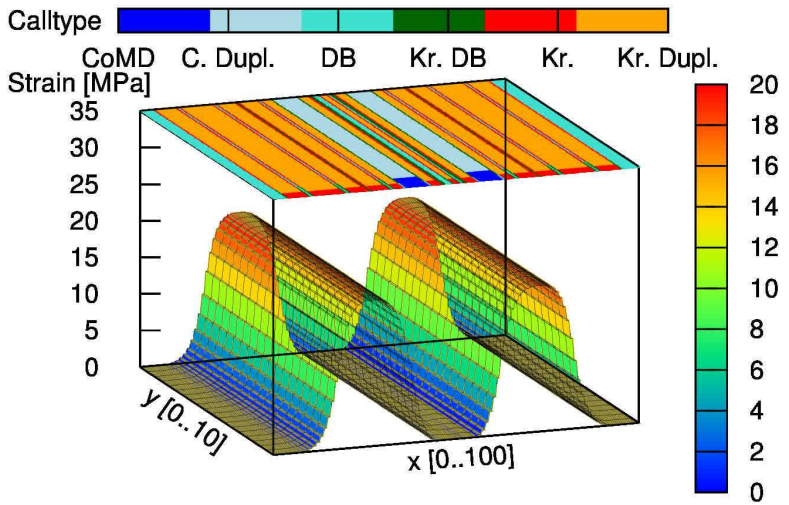
Framework Pros and Cons

Based on preliminary results!

- Charm++
  - More complex to implement (.ci files)
  - Great platform support, but uncommon build system
  - Good performance
  - Good documentation and support on mailing list
- Intel CnC
  - Straightforward to implement
  - Needs Intel MPI or MPICH
  - Good performance with optimization efforts
  - Good documentation for basics
  - Tuners mainly undocumented
- Libcircle
  - Trivial to implement
  - Great platform support
  - Performance NOT comparable
  - Manual serialization of input and output data

Summary and Outlook

- Implemented Distributed Database Kriging for Adaptive Sampling (D^2KAS) for HMM (elastodynamics) using different frameworks
- Our adaptive scheme achieves a speedup of $2.5 - 25$[1]
- Enables inclusion of defects, crystal domains or phase boundaries
- One code base: Charm++, CnC, OpenMP, Libcircle (github.com/exmatex/CoHMM)



Color bar right hand side: Strain. Color bar top: Type of call.

[1] Comp. Phys. Comm. **192**, 138 (2015)

Thanks to
Phil Miller (Audience, UI-UC)
Frank Schlimbach (Intel)

This work was supported by the Los Alamos Information Science & Technology Center (IS&T) Co-Design Summer School, the U.S. Department of Energy (DOE), Office of Advanced Scientific Computing Research (ASCR) through the Exascale Co-Design Center for Materials in Extreme Environments (ExMatEx), and the Center for Nonlinear Studies (CNLS).

The End

Thank you for your attention !

Los Alamos NATIONAL LABORATORY EST. 1943