Asynchronous PDE Solver for Computing at Extreme Scales

Konduri Aditya[1], Diego A. Donzis[1] and Torsten Hoefler[2]
[1]Department of Aerospace Engineering, Texas A&M University
[2]Department of Computer Science, ETH Zurich

Motivation

- Many natural and engineering systems can be described with PDEs
- Large number of numerical methods: finite difference, finite volume and spectral methods, along with parallel programming
  - Real life problem are complex, demand massive computations
  - Trend has been to increase degree of parallelism, currently at $O(10^5)$ processing elements (PEs)
- Computation rates are much faster than communication
- Challenge: communication between PEs at very large scales
  - Designing codes that scale to extreme degrees of parallelism is critical to exploit resources and simulate real-scale problems
- Exascale: communication likely to be the bottleneck
- Need to relax synchronization of data not just in communication algorithm, but also at mathematical level (Asynchronous schemes)

Literature

- Extensive work on iterative algorithms for algebraic systems (Bertsekas and Tsitsiklis 1997, Frommer and Szyld 2000)
- Algorithms restricted to parabolic PDEs (Amitai et al. 1998)

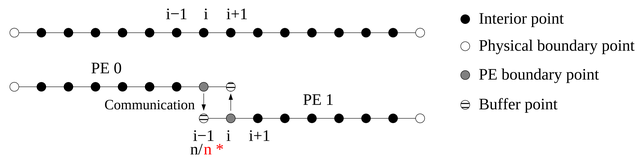
Concept

Consider the simple 1D heat equation

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Finite difference discretization

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + O(\Delta t, \Delta x^2)$$



Asynchronous computing

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} \approx \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^{n^*}}{\Delta x^2}$$

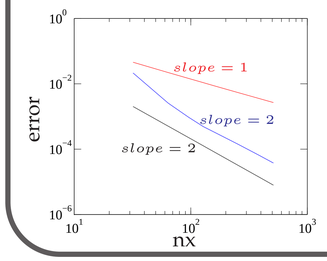
- Interior points use values at nth level
- Processing Element (PE) boundary points at n^*th level
- n^* will be a random variable and could be n, $(n-1)$, $(n-2)$, etc. based on communication time
- Schemes continue to remain stable (Konduri and Donzis, SC12, manuscript under preparation)
- Accuracy drops to first order in space for $n^* < n$

Asynchronous Schemes

- Can the accuracy of schemes be recovered?
- Analyzing the truncation error, we derive new schemes that will maintain the accuracy under time asynchronicity.

Now, if $n^* = n - k^*$, the second order scheme will be:

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2} = \frac{1}{2\Delta x^2}[&u_{i+3}^n - 2u_{i+2}^n + u_{i+1}^n \\ &+(k^*+1)(u_{i-1}^{n-k^*} - 2u_{i-2}^{n-k^*} + u_{i-3}^{n-k^*}) \\ &-k^*(u_{i-1}^{n-k^*-1} - 2u_{i-2}^{n-k^*-1} + u_{i-3}^{n-k^*-1})] + O(\Delta x^2)\end{aligned}$$



Need stencil values at multiple time levels
$k^* = 0$: synchronous value
$k^* > 0$: delayed
$k^* < 0$: value from future
– : synchronous
– : asynchronous
– : asynchronous new scheme

Algorithm

Requirements

- Non-blocking communication calls (overlap computation-communication)
- Communicate time stamp of the data to obtain the value of k
- Use buffer arrays to
  - store data at multiple time levels
  - allow simultaneous multiple communication messages

Algorithm

```
Initialization and communication setup
Time loop
  - communication test
    + check status of communication at different time levels
    + Synchronize data if required
    + Choose the latest time levels for computations
  - computation of field variables
  - initiate communication of updated field
  - auxillary computations
```

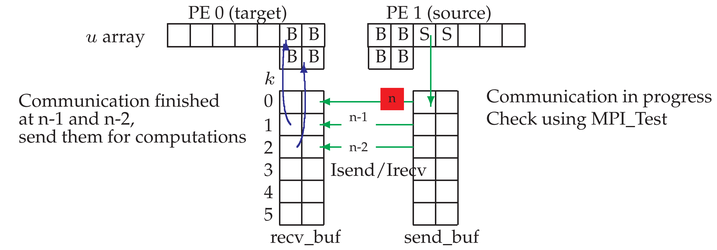
Conclusions

- Scheme that maintain accuracy under asynchronous computing have been presented
- The potential to eliminate virtually all communication overheads has been demonstrated
- A possible path to improve scalability at exascale

Acknowledgement: We are grateful to NERSC for computing time on Hopper and support.

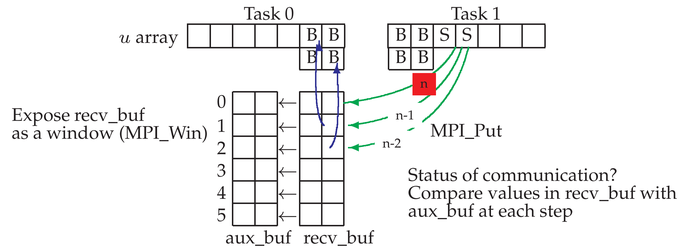
Communication

Two-sided non-blocking



- Both source and destination PEs are involved
- As synchronization is relaxed, adds an overhead time

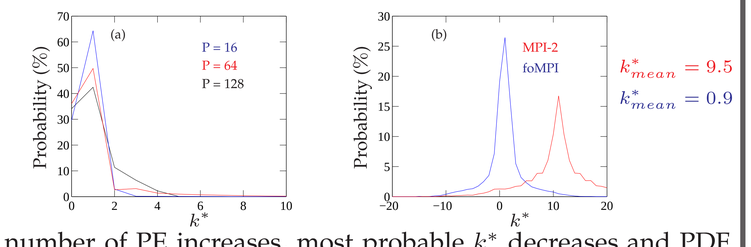
One-sided RMA



- Only source PEs are involved communication
- Better communication performance
- Atomicity of data is important

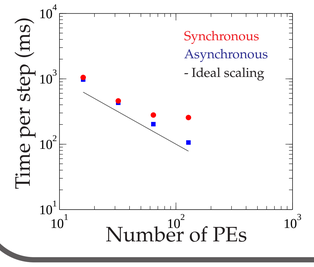
Results

Measured distribution of k^* on Hopper



- As number of PE increases, most probable k^* decreases and PDF becomes wider
- Fast one-sided MPI (foMPI, Gerstenberger et al. SC13) implementation gives lower values of k^* and hence, its mean

Strong scaling on Hopper



- Burgers' equation simulations
- Nearly ideal scaling
- Global synchronization completely avoided