Analysis and Implementation of Asynchronous Finite Difference Scheme for Advection - Diffusion Equation

Kumar Saurabh [1] Prof. S. Sundar[1] Prof. Dr. Martin Frank [2]

[1] Department of Mathematics
IIT Madras

[2] Math CCES
RWTH Aachen

August 31, 2016

Acknowledgement: DAAD, RWTH IT Center

Motivation

- Many natural and engineering systems can be described with PDEs
  - Fluid mechanics, Electromagnetism, Quantum Mechanics
- Analytical Solution not known. Need to solve these problems numerically.

[1] Maitham Alhubail, Qiqi Wan. The swept rule for breaking the latency barrier in time advancing PDEs

[2] Jagannathan & Donzis (XSEDE 2012), Sankaran et al. (SC 2012), Lee et al. (SC 2013)

Motivation

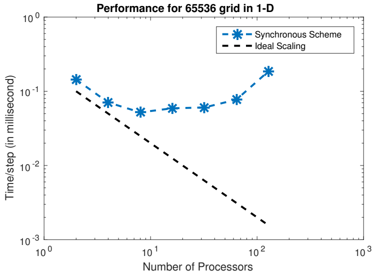
- Many natural and engineering systems can be described with PDEs
  - Fluid mechanics, Electromagnetism, Quantum Mechanics
- Analytical Solution not known. Need to solve these problems numerically.
- Large number of numerical methods: Finite Difference, Finite Volume, Spectral Method, etc.
  - The problem is decomposed into a large number of Processing Element (PEs).
  - Extreme-scale computer clusters can solve PDEs using over 1,000,000 cores. [1]
  - The communication is required between the PEs to solve the PDEs to compute spatial derivatives.

[1] Maitham Alhubail, Qiqi Wan. The swept rule for breaking the latency barrier in time advancing PDEs

[2] Jagannathan & Donzis (XSEDE 2012), Sankaran et al. (SC 2012), Lee et al. (SC 2013)

Motivation

- Many natural and engineering systems can be described with PDEs
  - Fluid mechanics, Electromagnetism, Quantum Mechanics
- Analytical Solution not known. Need to solve these problems numerically.
- Large number of numerical methods: Finite Difference, Finite Volume, Spectral Method, etc.
  - The problem is decomposed into a large number of Processing Element (PEs).
  - Extreme-scale computer clusters can solve PDEs using over 1,000,000 cores. [1]
  - The communication is required between the PEs to solve the PDEs to compute spatial derivatives.
- Computation rates are much faster than communication
  - Exascale: Communication likely to be bottleneck.
  - Direct Numerical Simulation (DNS): Communication or synchronization takes upto 50 - 70 % of the total computation time. [2]
- Petascale computation: What if one of the node fails?



[1] Maitham Alhubail, Qiqi Wan. The swept rule for breaking the latency barrier in time advancing PDEs

[2] Jagannathan & Donzis (XSEDE 2012), Sankaran et al. (SC 2012), Lee et al. (SC 2013)

Asynchronous Schemes

Asynchronous Scheme for Advection Diffusion Equation

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^{\tilde{n}} - u_{i-1}^n) = \frac{\alpha}{\triangle x^2}(u_{i+1}^{\tilde{n}} - 2u_i^n + u_{i-1}^n) \tag{1}$$

- Regarding $\tilde{n}$
  - Synchronous when $\tilde{n} = n$
  - $\tilde{n}$ can be $n, n-1, n-2, ...$
  - Concrete value of $\tilde{n}$ depends on hardware, network, traffic, (possible) unpredictable factors,...
  - $\tilde{n}$ is in fact a principle random variable.

[3] Diego, A. Donzis and Konduri, Aditya, 2004 "Asynchronous Finite Difference Scheme for Partial Difference Equations", Journal of Computational Physics. 274(0), pp: 370-392.

[4] Mudigree, D., Sherleker,S., Ansumali, S., 2014 "Delayed Difference scheme for large scale scientific simulations", Physical Review Letters, 113(21), 218701.

[5] Assumptions and constraints stated in thesis.

Asynchronous Schemes

Asynchronous Scheme for Advection Diffusion Equation

$$\frac{1}{\triangle t}(u_i^{n+1} - u_i^n) + \frac{c}{2\triangle x}(u_{i+1}^{\tilde{n}} - u_{i-1}^n) = \frac{\alpha}{\triangle x^2}(u_{i+1}^{\tilde{n}} - 2u_i^n + u_{i-1}^n) \tag{1}$$

- Regarding $\tilde{n}$
  - Synchronous when $\tilde{n} = n$
  - $\tilde{n}$ can be $n, n-1, n-2, \ldots$
  - Concrete value of $\tilde{n}$ depends on hardware, network, traffic, (possible) unpredictable factors,...
  - $\tilde{n}$ is in fact a principle random variable.

Analysis

- Stability Stable if the Synchronous Scheme is stable, irrespective of delay statistics. [3]
- Truncation Error: Not homogeneous in space and random.
  - Need for statistical description of the truncation error.
  - Strong Scaling: $\langle \overline{E} \rangle \sim O(\Delta x)$
  - Weak Scaling: $\langle \overline{E} \rangle \sim O(1)$
  - Verified by numerical experiments.
  - $\langle \overline{E} \rangle \propto \overline{\tilde{k}}$: Higher the delay, higher will be the error.
- Asynchrony Tolerant Schemes
  - Need for higher order schemes that are capable to maintain accuracy.
  - Truncation Error Analysis.
  - Previous work by Mudigree et al. [4], Donzis and Aditya[3] proposed such schemes.[5]

[3] Diego, A. Donzis and Konduri, Aditya, 2004 “Asynchronous Finite Difference Scheme for Partial Difference Equations”, Journal of Computational Physics. 274(0), pp: 370-392.

[4] Mudigree, D., Sherleker,S., Ansumali, S., 2014 “Delayed Difference scheme for large scale scientific simulations”, Physical Review Letters, 113(21), 218701.

[5] Assumptions and constraints stated in thesis.

Implementation of Asynchronous Scheme

Algorithms

- Deterministic Asynchronous Scheme
  - Error: Deterministic - Independent of runs
  - Exchange the information after a certain amount of steps. (SYNC_STEP)
  - Naive way of Implementation
- Stochastic Asynchronous Scheme
  - Error: Different for different runs.
  - Do not wait for the communication to complete.
  - Use the latest time values.
  - Dependent on delay statistics.

Implementation of Asynchronous Scheme

Algorithms

- Deterministic Asynchronous Scheme
  - Error: Deterministic - Independent of runs
  - Exchange the information after a certain amount of steps. (SYNC_STEP)
  - Naive way of Implementation
- Stochastic Asynchronous Scheme
  - Error: Different for different runs.
  - Do not wait for the communication to complete.
  - Use the latest time values.
  - Dependent on delay statistics.

MPI Implementation

- Two Sided Non - blocking Communication
  - Achieved using MPI_Isend / MPI_Irecv / MPI_Test
  - Data to be transferred is stored in the buffer.
  - Buffering of data takes place.
  - Handshaking occurs - Matching tags and rank.
- One Sided RMA
  - Achieved using MPI_Put / MPI_Lock / MPI_Unlock
  - Target processor exposes its location to memory.
  - No buffering - Source processor writes into the target location before returning.
  - Redundancy of data is prevented by MPI_Lock.
  - No handshaking - Consent of target processor is not required.

Results

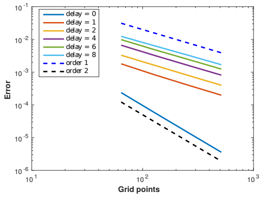


Figure : Asynchronous Schemes

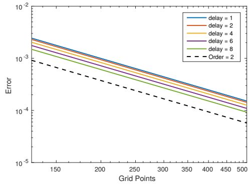


Figure : Asynchrony Tolerant (AT) Schemes

Results

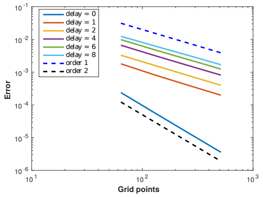


Figure : Asynchronous Schemes

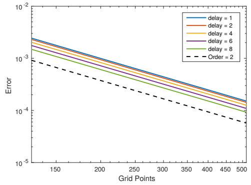


Figure : Asynchrony Tolerant (AT) Schemes

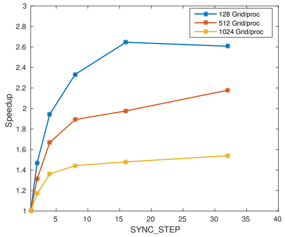


Figure : Effect of load and SYNC_STEP on speedup

Results

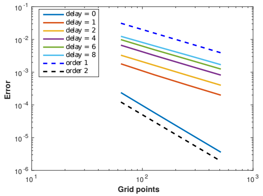


Figure : Asynchronous Schemes

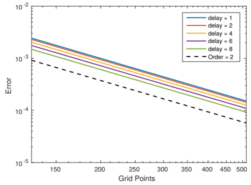


Figure : Asynchrony Tolerant (AT) Schemes

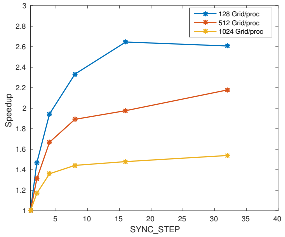


Figure : Effect of load and SYNC_STEP on speedup

Deterministic Implementation with 2048 Grid Points per processor		Stochastic Implementation with 2048 Grid Points per processor	
SYNC_STEP	Speedup	Maximum Allowable Delay	Speedup
5	1.3615	5	1.8589
10	1.4474	10	2.6174
20	1.4889	20	3.7192
30	1.5300	30	6.0154

Table : Comparison of Deterministic and Stochastic Asynchronous Implementation

Results

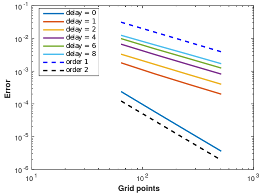


Figure : Asynchronous Schemes

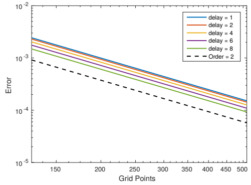


Figure : Asynchrony Tolerant (AT) Schemes

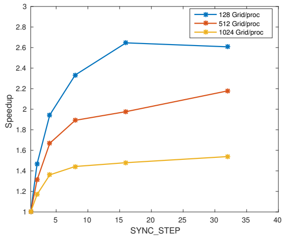


Figure : Effect of load and SYNC_STEP on speedup

Deterministic Implementation with 2048 Grid Points per processor		Stochastic Implementation with 2048 Grid Points per processor	
SYNC_STEP	Speedup	Maximum Allowable Delay	Speedup
5	1.3615	5	1.8589
10	1.4474	10	2.6174
20	1.4889	20	3.7192
30	1.5300	30	6.0154

Table : Comparison of Deterministic and Stochastic Asynchronous Implementation

Effect of Computer Architecture on High Load

- RMA operation

Processor per node	Time (s)	Maximum Delay	Average Delay
8	204.288266	38	0.152375
4	162.511951	42	0.013750
2	122.102558	44	0.075125
1	93.667287	20	0.132000

- Two-sided Non blocking Send and Receive

Processor per node	Time (s)	Maximum Delay	Average Delay
8	151.922679	65	10.7976
4	112.257639	123	20.1246
2	111.381462	137	17.8507
1	84.915095	110	13.0063

Conclusion

- Original Scheme:
  - Second Order Convergence without delays.
  - First Order Convergence in presence of delays.
- Second Order Convergence in presence of delays.
  - Newer Scheme - Asynchrony Tolerant Scheme.
- Fewer Communication calls $\Rightarrow$ More speedup.
- Higher Load on processor $\Rightarrow$ Communication cost less significant.
- Stochastic Implementation - More Speedup.
- Study of Delay Statistics.
  - MPI_RMA: better for asynchronous case in terms of average delay statistics.
  - Dependence of load on the processor. More frequent communication $\Rightarrow$ Larger Delays.
  - Impact of Computer Architecture.