A scalable asynchronous computing SAND2018-6990C g PDEs at extreme scale

Konduri Aditya

Sandia National Laboratories, Livermore, CA

Sandia National Laboratories

Acknowledgments

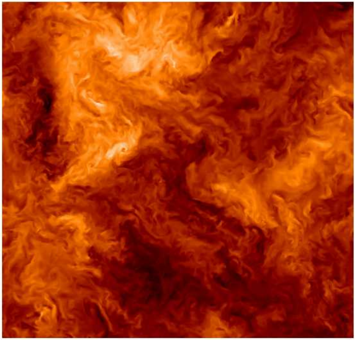
Emmet Cleary (CalTech)
Dr. Diego A. Donzis (Texas A&M University)
Dr. Jacqueline H. Chen (SNL)
Dr. Torsten Hoefler (ETH Zurich)

Funding:

Computing resources:

XSEDE
Extreme Science and Engineering Discovery Environment

Turbulence



A. F. Maqui and D. A. Donzis (2011)

- DNS: resolve all scales both in time and space
    - Large scale (L)
    - Kolmogorov small scale (η)
    - Using classical scaling (Kolmogorov 1941)

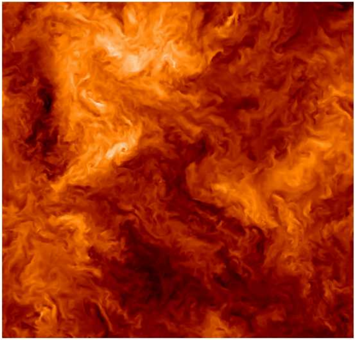
$$\frac{L}{\eta} \sim R_\lambda^{3/2}$$

where $R_\lambda = u\lambda/\nu$ is the Reynolds number

- Typically very high in applications
- Computational effort (C)

$$C \sim R_\lambda^6$$

Turbulence



A. F. Maqui and D. A. Donzis (2011)

- DNS: resolve all scales both in time and space
  - Large scale (L)
  - Kolmogorov small scale (η)
  - Using classical scaling (Kolmogorov 1941)

$$\frac{L}{\eta} \sim R_\lambda^{3/2}$$

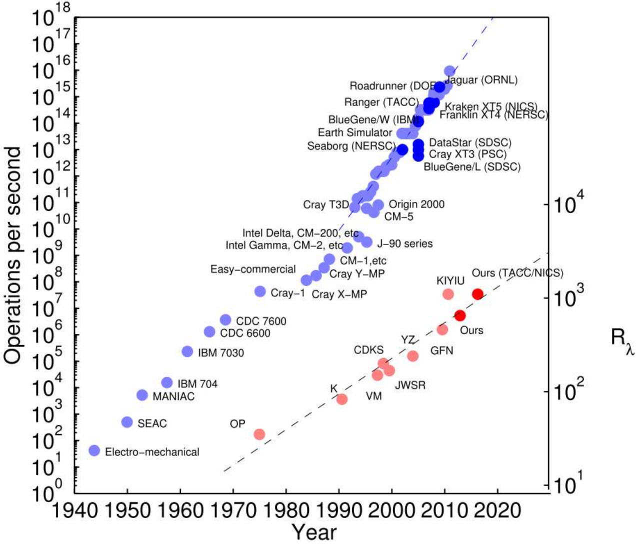
where $R_\lambda = u\lambda/\nu$ is the Reynolds number

- Typically very high in applications
- Computational effort (C)

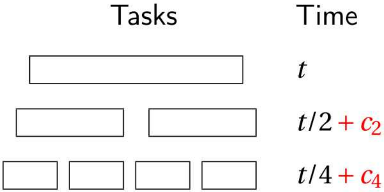
$$C \sim R_\lambda^6$$

- Parallel computing

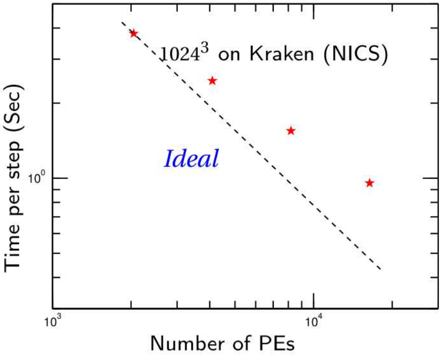
Turbulence simulations



Parallel programming



Strong scaling[1]



Ideal scaling:

Number of processing elements (P)

Computational effort $C \sim 1/P$

- Communication affects scalability

[1] S. Jagannathan and D. A. Donzis (XSEDE 2012)

Exascale

Issues

- Large number of communications
  - increase communication time
  - likely to be a bottleneck
- Performance variations in processing elements
  - potential issues: noise, inadequate cooling
- Synchronizations enforced at multiple levels (communication, mathematics, etc.)
- system faults
  - failure of a PE, Node, etc.

J. Dongarra et al. (IJHPCA 2011)

Exascale

Issues

- Large number of communications
  - increase communication time
  - likely to be a bottleneck
- Performance variations in processing elements
  - potential issues: noise, inadequate cooling
- Synchronizations enforced at multiple levels (communication, mathematics, etc.)
- system faults
  - failure of a PE, Node, etc.

will affect scalability further

J. Dongarra et al. (IJHPCA 2011)

Literature

- Extensive work on iterative algorithms for algebraic systems (D. P. Bertsekas and J. N. Tsitsiklis 1997, A. Frommer and D. B. Szyld 2000)
  - can be used in implicit solvers for $\boldsymbol{B}\boldsymbol{V}^{n+1} = \boldsymbol{V}^n$ solution at every time step
  - still need data synchronization at the end of time step
- Finite difference algorithms restricted to parabolic PDEs (D. Amitai et al. 1993, 1998)
  - use corrections based on Greens function at PE boundaries
  - restricted to lower order
  - not suitable for fundamental studies

Literature

- Extensive work on iterative algorithms for algebraic systems (D. P. Bertsekas and J. N. Tsitsiklis 1997, A. Frommer and D. B. Szyld 2000)
    - can be used in implicit solvers for $\boldsymbol{B}\boldsymbol{V}^{n+1} = \boldsymbol{V}^n$ solution at every time step
    - still need data synchronization at the end of time step
- Finite difference algorithms restricted to parabolic PDEs (D. Amitai et al. 1993, 1998)
    - use corrections based on Greens function at PE boundaries
    - restricted to lower order
    - not suitable for fundamental studies

Need a new method!

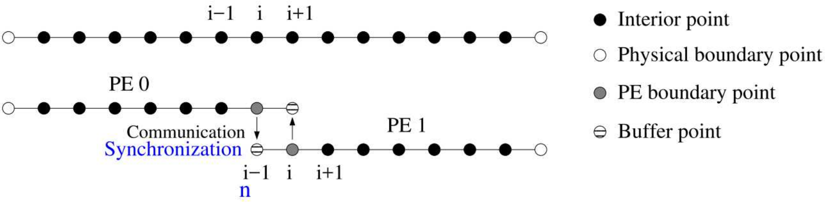
Finite difference method

Consider the simple 1D heat equation

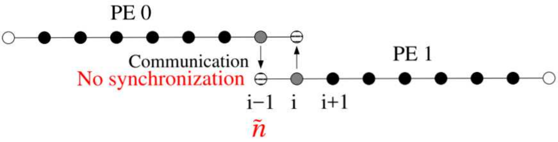
$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

FD equation - forward in time and central in space

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



Asynchronous computing



- $u_{i-1}^{\tilde{n}}$ is the available data at $(i-1)$th grid point
- $\tilde{n}$ could be n, $(n-1)$, $(n-2)$, etc. time levels

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

- Interior points used in computations are at nth level
- Buffer points are at $\tilde{n} = n - \tilde{k}$ th level
- What are the properties of such a scheme?

Parameters

Numerical properties depend on

- Grid resolution (N)
- Stencil size (S)
- Number of PE (P)
- Number of delay levels (L)
- Probability of level k (p_k)
  - Communication performance: Network latency, bandwidth; message characteristics; MPI implementation; ETC.

$\tilde{n} = n - \tilde{k}$ is a random variable at PE boundaries

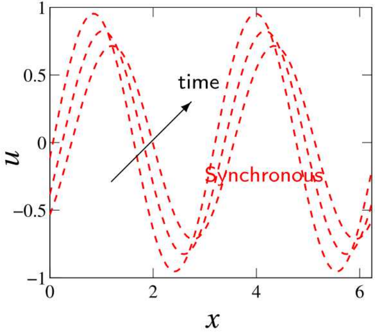
Example: $L = 2$, $\tilde{k} = \{0, 1\}$

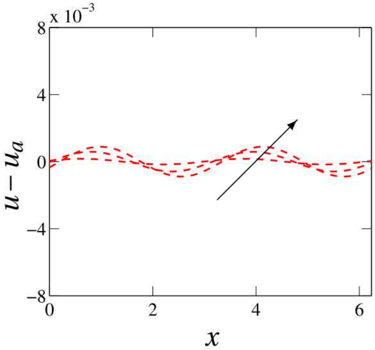
If $p_0 = 0.6$, then

$p_1 = 1 - p_0 = 0.4$

Results

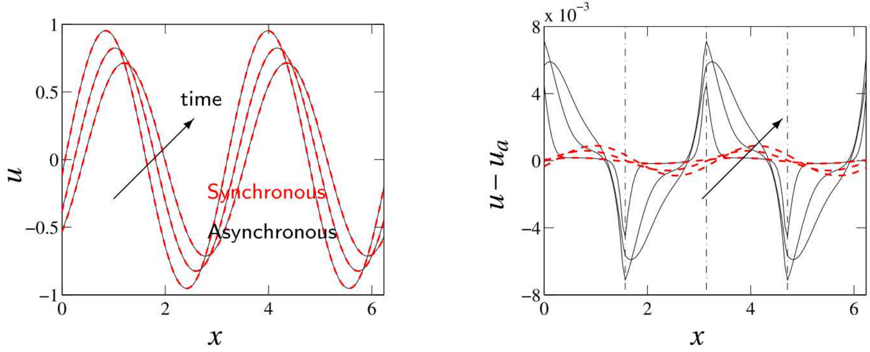
- Linear advection-diffusion equation
- Time evolution of solution and error





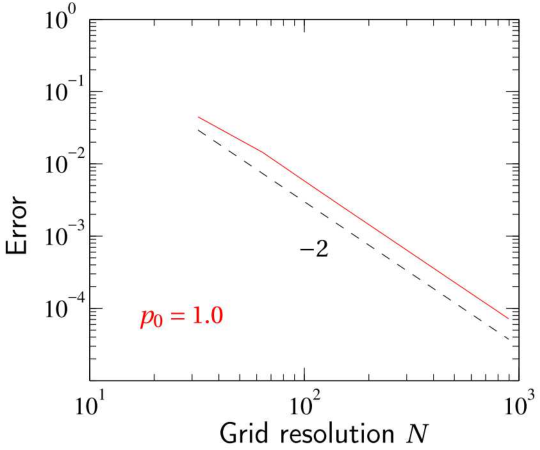
Results

- Linear advection-diffusion equation
- Time evolution of solution and error



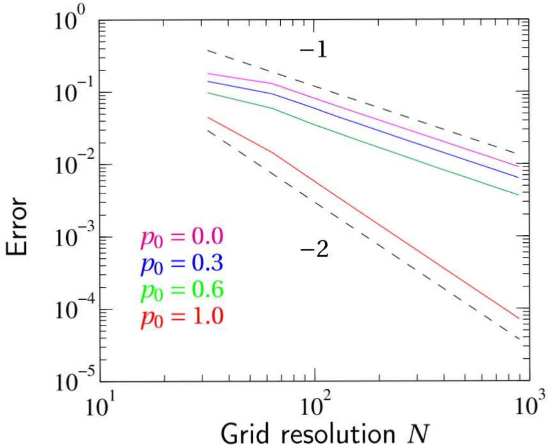
Parameters: $P = 4,\ L = 2,\ p_k = \{0.0, 1.0\}$

Accuracy



[1]K. Aditya and D. A. Donzis (SC 2012), [2]D. A. Donzis and K. Aditya (JCP 2014)

Accuracy



- Solution is stable
- Accuracy drops to first order with asynchrony

[1]K. Aditya and D. A. Donzis (SC 2012), [2]D. A. Donzis and K. Aditya (JCP 2014)

Accuracy

- Using Taylor series

$\tilde{\mathbf{n}} = \mathbf{n}$

$$E_i^n = -\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 + \mathcal{O}(\Delta t^2, \Delta x^4).$$

Accuracy

- Using Taylor series

$\tilde{\mathbf{n}} = \mathbf{n}$

$$E_i^n = -\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 + \mathcal{O}(\Delta t^2, \Delta x^4).$$

$\tilde{\mathbf{n}} = \mathbf{n-1}$

$$\begin{aligned} \tilde{E}_i^n|_1 \quad = \quad & -\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 - \alpha \dot{u}\frac{\Delta t}{\Delta x^2} + \alpha \dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha \dot{u}''}{2}\Delta t \\ & + \mathcal{O}(\Delta x^3, \Delta t^2, \Delta x^p \Delta t^q) \end{aligned}$$

Accuracy

- Using Taylor series

$\tilde{\mathbf{n}} = \mathbf{n}$

$$E_i^n = -\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 + \mathcal{O}(\Delta t^2, \Delta x^4).$$

$\tilde{\mathbf{n}} = \mathbf{n} - \mathbf{1}$

$$\begin{aligned}\tilde{E}_i^n|_1 \quad = \quad & -\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 - \alpha \dot{u}\frac{\Delta t}{\Delta x^2} + \alpha \dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha \dot{u}''}{2}\Delta t \\ & + \mathcal{O}(\Delta x^3, \Delta t^2, \Delta x^p \Delta t^q)\end{aligned}$$

$\tilde{\mathbf{n}} = \mathbf{n} - \tilde{\mathbf{k}}$

$$\begin{aligned}\tilde{E}_i^n|_k \quad = \quad & -\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 - \alpha k \dot{u}\frac{\Delta t}{\Delta x^2} + \alpha k \dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha k \dot{u}''}{2}\Delta t \\ & + \mathcal{O}(\Delta x^3, \Delta t^2, \Delta x^p \Delta t^q),\end{aligned}$$

Delay k appears as prefactor, does not affect order with asynchrony

Accuracy

- Overall error in the domain

$$\langle \overline{E} \rangle = \frac{1}{N} \sum_{i=1,N} \overline{E_i^n}.$$

$$\langle \overline{E} \rangle = \frac{1}{N} \left[\underbrace{\sum_{i \in I_I} E_i^n}_{\text{Interior points } (N_I)} + \underbrace{\sum_{i \in I_B} \overline{\tilde{E}_i^n|_{\tilde{k}}}}_{\text{PE boundary points } (N_B)} \right]$$

- When error due to asynchrony dominates, it scales as:

$$\langle \overline{E} \rangle \sim \bar{\tilde{k}} \frac{P}{N} \sim \bar{\tilde{k}} P \Delta x$$

Accuracy

- Overall error in the domain

$$\langle \overline{E} \rangle = \frac{1}{N} \sum_{i=1,N} \overline{E_i^n}.$$

$$\langle \overline{E} \rangle = \frac{1}{N} \left[\sum_{i \in I_I} E_i^n + \sum_{i \in I_B} \overline{\tilde{E}_i^n|_{\tilde{k}}} \right]$$

Interior points (N_I) PE boundary points (N_B)

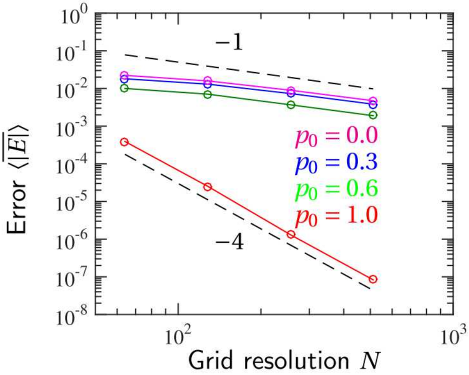
- When error due to asynchrony dominates, it scales as:

$$\langle \overline{E} \rangle \sim \bar{\tilde{k}} \frac{P}{N} \sim \bar{\tilde{k}} P \Delta x$$

- Strong scaling: $\langle \overline{E} \rangle \sim \mathcal{O}(\Delta x)$
- Weak scaling $\langle \overline{E} \rangle \sim \mathcal{O}(1)$
- Independent of accuracy of original scheme: drastic effect of asynchrony
- Validation of this theoretical framework: Donzis and Aditya (JCP 2014)

Accuracy

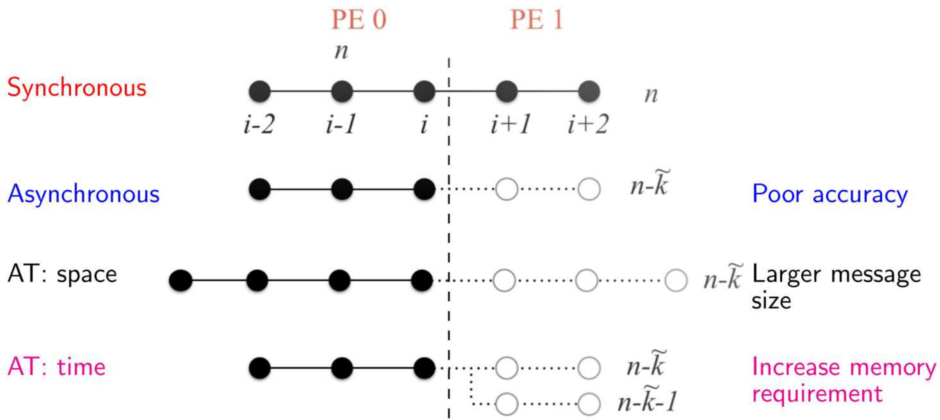
- Results from fourth order accurate simulations



$$\boxed{\langle\overline{E}\rangle \sim \bar{\tilde{k}}\frac{P}{N} \sim \bar{\tilde{k}}P\Delta x}$$

Asynchrony-tolerant schemes

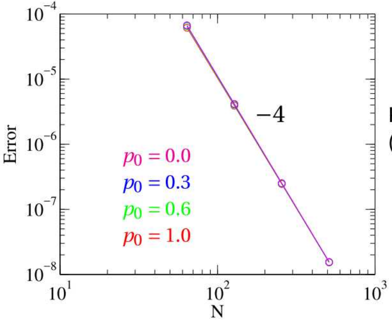
Stencil structure:



Asynchrony-tolerant schemes

Time: Runge-Kutta second order, space: fourth order AT scheme

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2} = & \frac{1}{24\Delta x^2}\Big[(k+1)(k+2)(-u_{i-2}^{n-k}+16u_{i-1}^{n-k}-30u_i^n+16u_{i+1}^n-u_{i+2}^n) \\ & -k(k+2)(-u_{i-2}^{n-k-1}+16u_{i-1}^{n-k-1}-30u_i^n+16u_{i+1}^n-u_{i+2}^n) \\ & +k(k+1)(-u_{i-2}^{n-k-2}+16u_{i-1}^{n-k-2}-30u_i^n+16u_{i+1}^n-u_{i+2}^n)\Big]\end{aligned}$$

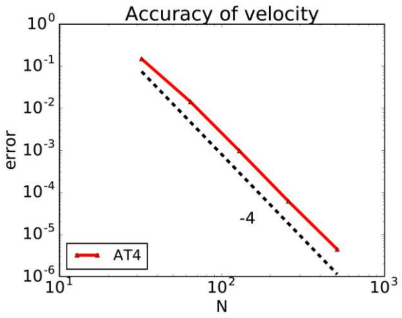


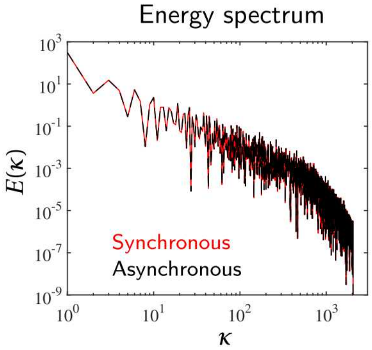
K. Aditya and D. A. Donzis (JCP 2017)

$$\boxed{\langle \overline{E} \rangle \sim P\Delta x^{a+1} \sum_{m=1}^{\mathcal{N}} \gamma_m \overline{\tilde{k}^m}}$$

Reacting flows

- Simulation of 1D premixed auto-ignition
- Fourth order central difference and asynchrony-tolerant schemes





Asynchronous algorithm

Consider for example:

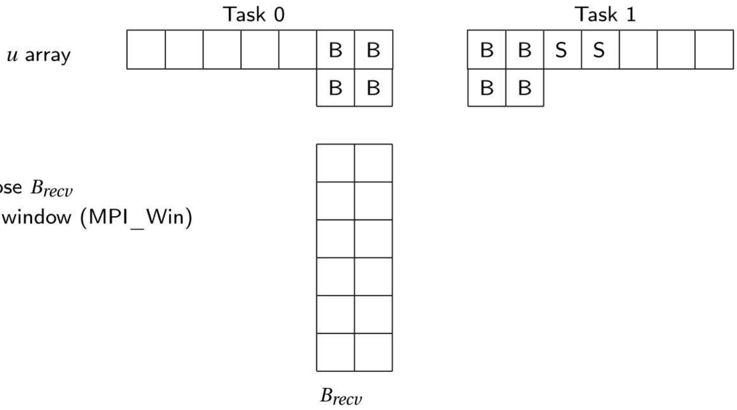
$$\begin{aligned}\frac{\partial^2 u}{\partial x^2} \approx \; & \frac{1}{2\Delta x^2}\Big[u_{i+3}^{n} - 2u_{i+2}^{n} + u_{i+1}^{n} \\ & + (\tilde{k}+1)\left(u_{i-1}^{n-\tilde{k}} - 2u_{i-2}^{n-\tilde{k}} + u_{i-3}^{n-\tilde{k}}\right) \\ & - \tilde{k}\left(u_{i-1}^{n-\tilde{k}-1} - 2u_{i-2}^{n-\tilde{k}-1} + u_{i-3}^{n-\tilde{k}-1}\right)\Big]\end{aligned}$$

What is it required to implement such a scheme efficiently?

- Communication: non-blocking, synchronization only when $\tilde{k} > L$
- Store data from multiple time levels
- Knowledge of time level ($\tilde{k}$) of each message
- Atomicity while accessing values

One-sided remote memory access (RMA) communications

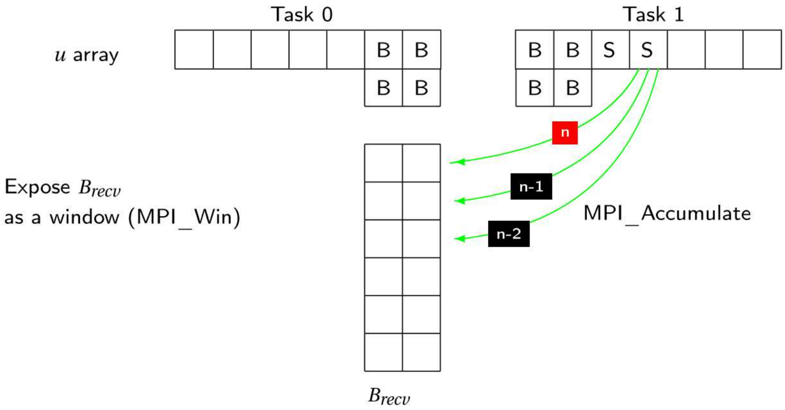
Asynchronous communications



- Atomicity of data is critical

K. Aditya, D. A. Donzis and T. Hoefler (SC 2013)

Asynchronous communications



- Atomicity of data is critical

K. Aditya, D. A. Donzis and T. Hoefler (SC 2013)

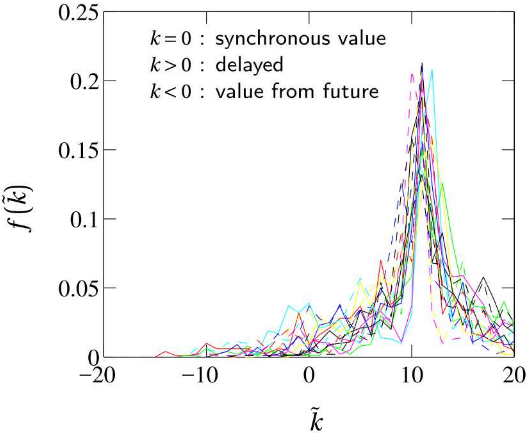
Asynchronous algorithm

1. *Initialize*
2. *Synchronous time loop to populate* B_{recv}
   a. *Compute* $\boldsymbol{U}_{\boldsymbol{i}}^{n+1} = f\left(\boldsymbol{U}^n\right), \boldsymbol{i} \in I$
   b. *Communication updated values*

Asynchronous algorithm

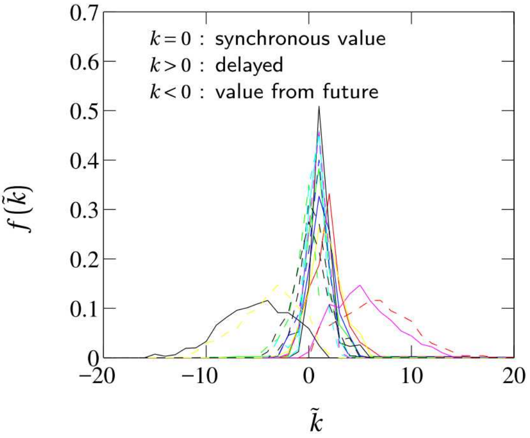
1. *Initialize*
2. *Synchronous time loop to populate B_{recv}*
   a. *Compute* $\boldsymbol{U}_{\boldsymbol{i}}^{n+1} = f\left(\boldsymbol{U}^{n}\right), \boldsymbol{i} \in I$
   b. *Communication updated values*
3. *Index values in B_{recv} and create B_{ichg}*
4. *Asynchronous time loop*
   a. *Compute* $\boldsymbol{U}_{\boldsymbol{i}}^{n+1} = f\left(\boldsymbol{U}^{n}\right), \boldsymbol{i} \in I_I$
   b. *Check communication status*
   c. *Synchronize if delays* $\tilde{k} > L$
   d. *Compute* $\boldsymbol{U}_{\boldsymbol{i}}^{n+1} = f\left(\boldsymbol{U}^{n}, \boldsymbol{U}^{n-\tilde{k}}, \boldsymbol{U}^{n-\tilde{k}-1}\right), \boldsymbol{i} \in I_B$
   e. *Initiate communication of updated values*

$\tilde{k}$ distribution: Hopper-MPI



- $N = 128$, $P = 8$, $L = 20$
- Averaged over 10 runs
- Mean: 9.6
- Standard deviation: 5.4

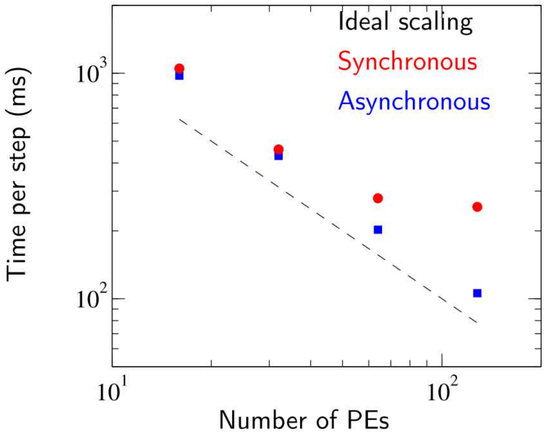
$\tilde{k}$ distribution: Hopper-foMPI



- Mean: 0.9
- Standard deviation: 1.9
- MPI implementation affects performance

foMPI implementation: R. Gerstenberg et al. (SC 2013)

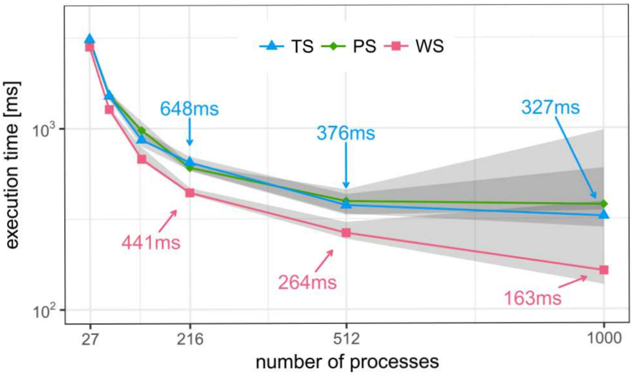
Strong scaling



- Burgers' equations, $N = 32K$, $L = 20$
- Nearly ideal scaling with asynchronous computing

K. Aditya, D. A. Donzis and T. Hoefler (SC 2013)

Strong scaling



- Traditional synchronous (TS)
- Pyramid synchronous (PS) - over decompose domain
- Weakly synchronous (WS) - asynchronous method

K. Aditya et al. (manuscript under preparation)

Conclusions

- Asynchronous computing: viable approach to solve PDEs accurately at extreme scales
- Developed theoretical framework: $\langle \overline{E} \rangle \sim \frac{P}{N} \Delta x^a \sum_{m=1}^{\mathscr{T}} \gamma_m \overline{\tilde{k}^m}$
  - Combine numerical as well as machine-specific architectural aspects
  - Understand effects of asynchrony
  - Devise new schemes which are tolerant to asynchrony
- Developed algorithm that has the potential to eliminate virtually all communication overheads at Exascale by eliminating forced synchronizations and overlapping communication and computation

Future work:

- Perform 3D reacting flow direct numerical simulations
- Couple the method with asynchronous runtime models

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.