UNIVERSITY of DELAWARE

The Numerical Reproducibility Fair Trade: Facing the Concurrency Challenges at the Extreme Scale

Michela Taufer

With Dylan Chapp, Travis Johnston

Based on our IEEE Cluster 2015 paper

University of Delaware

Reproducible Accuracy

- From Van Nostrand's Scientific Encyclopedia

  Reproducibility: "closeness of agreement among repeated simulation results under the same initial conditions over time"

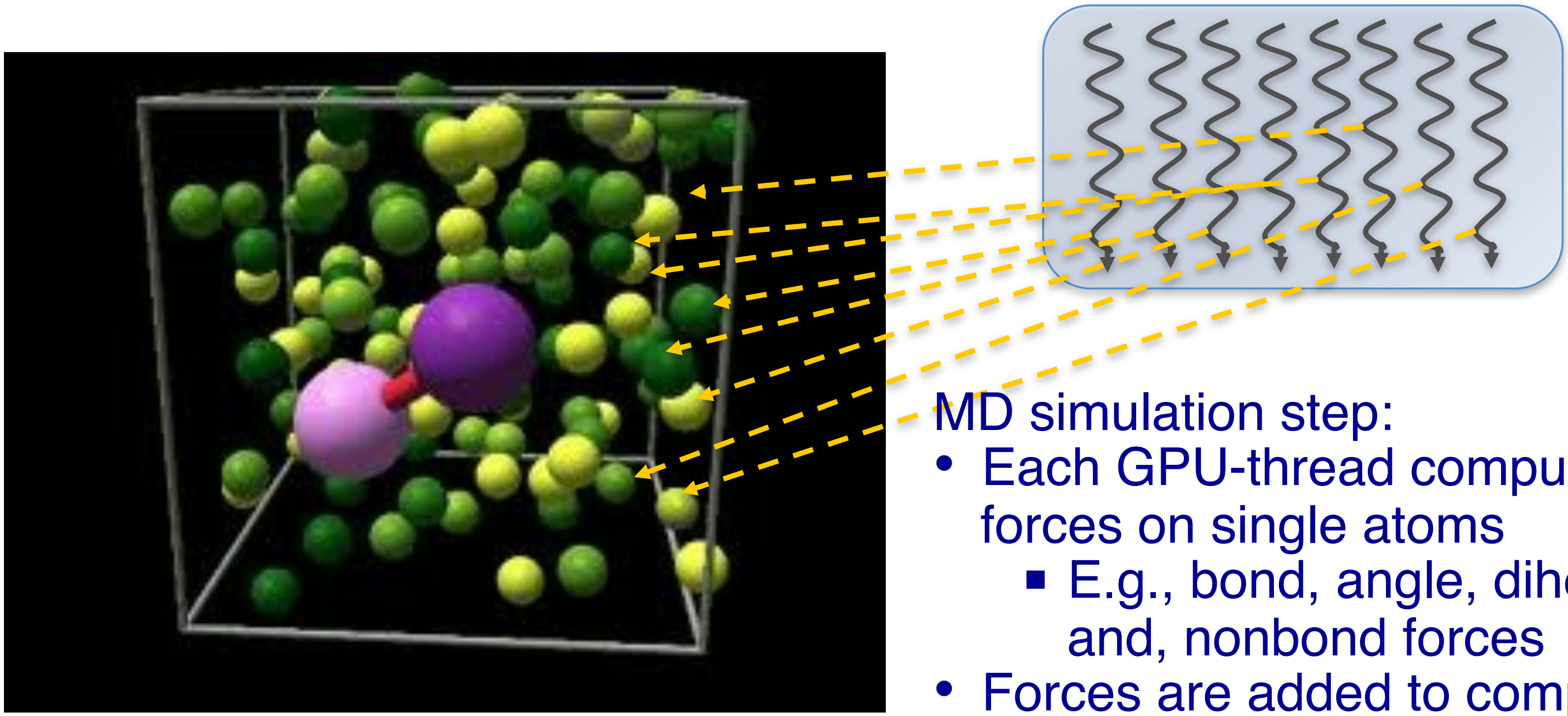
  Accuracy: "conformity of a resulted value to an accepted standard (or scientific laws)"

- Context: ensemble simulations of scientific phenomena at extreme scale with multithreading hardware consisting of multi-core processors coupled with many-core accelerators

- Repeatability (Same team, same experimental setup)
  - The measurement can be obtained with stated precision by the same team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same location on multiple trials. For computational experiments, this means that a researcher can reliably repeat her own computation.
- Replicability (Different team, same experimental setup)
  - The measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author's own artifacts.
- Reproducibility (Different team, different experimental setup)
  - The measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts which they develop completely independently.

From: https://www.acm.org/publications/policies/artifact-review-badging

Molecular Dynamics on Accelerators



Force → Acceleration → Velocity → Position

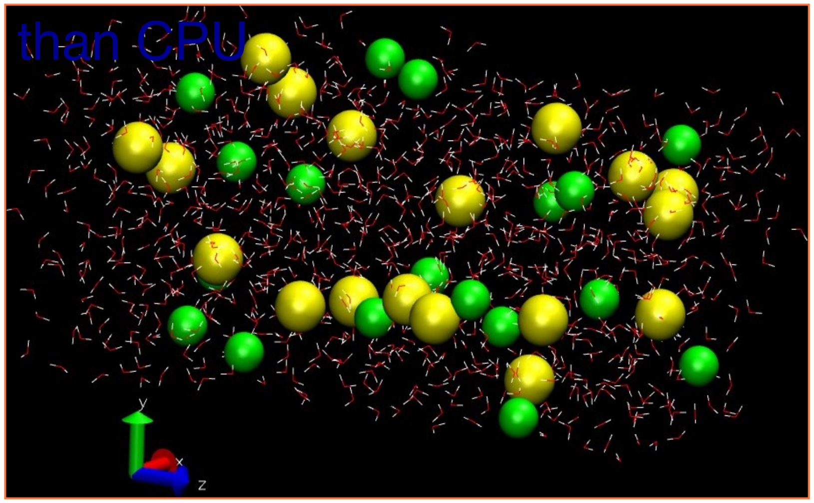
MD simulation step:

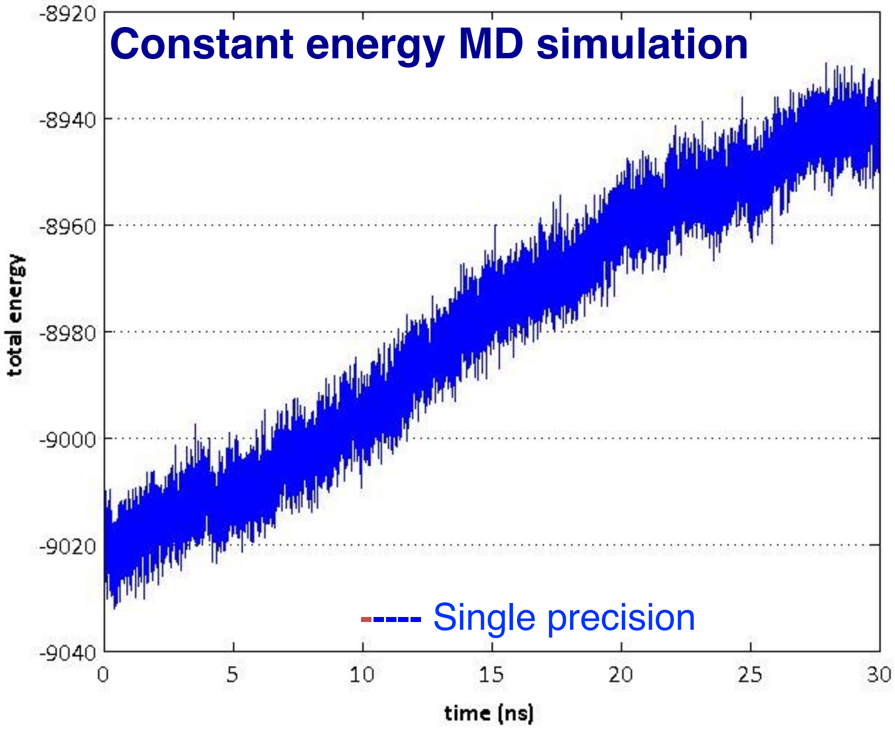
- Each GPU-thread computes forces on single atoms
  - E.g., bond, angle, dihedrals and, nonbond forces
- Forces are added to compute acceleration
- Acceleration is used to update velocities
- Velocities are used to update the positions

The Strange Case of Constant Energy MDs

- Enhancing performance of MD simulations allows simulations of larger time scales and length scales
- GPU computing enables large-scale MD simulation
  - Simulations exhibit unprecedented speed-up factors

MD simulation of NaI solution system containing 988 waters, 18 Na+, and 18 I–: GPU is X15 faster than CPU

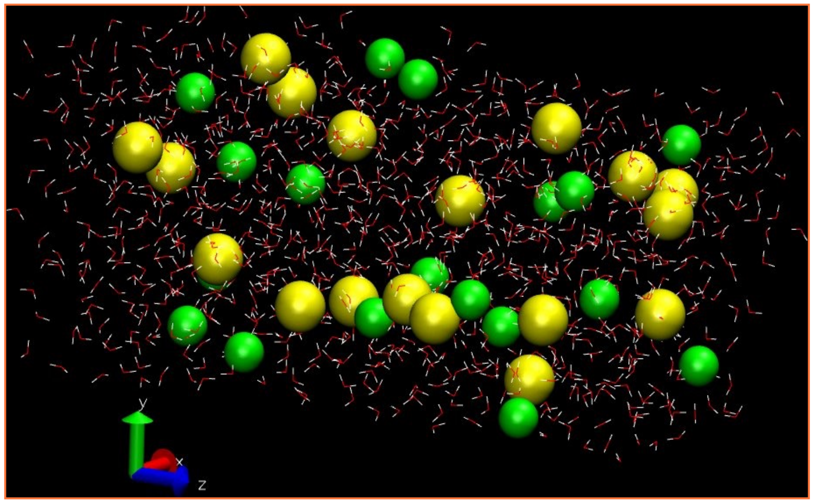


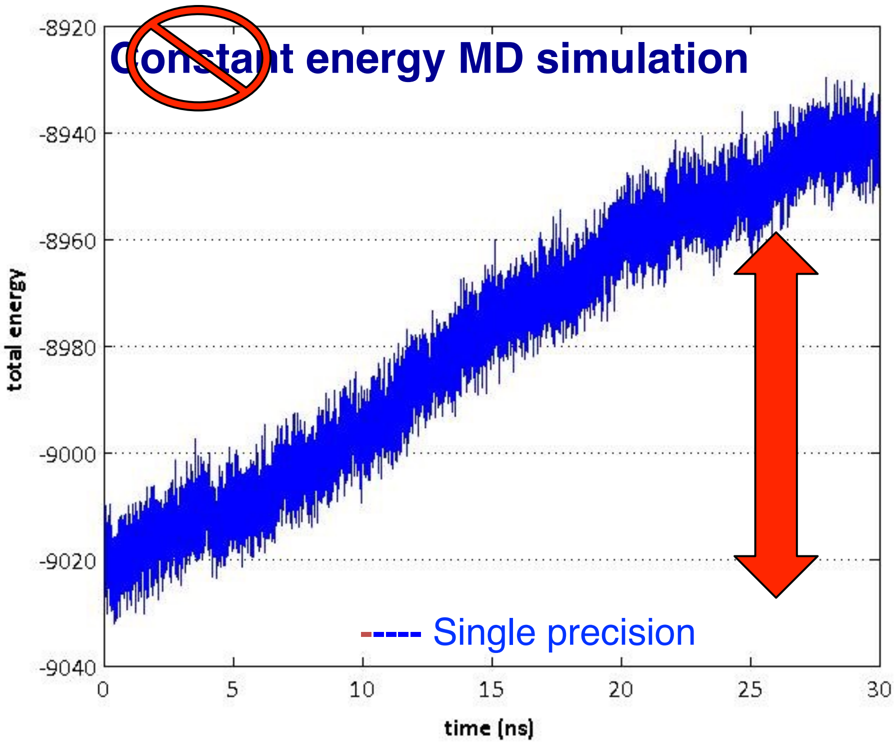


The Strange Case of Constant Energy MDs

- Enhancing performance of MD simulations allows simulations of larger time scales and length scales
- GPU computing enables large-scale MD simulation
  - Simulations exhibit speed-up factors of X10-X30

MD simulation of NaI solution system containing 988 waters, 18 Na+, and 18 I–: GPU is X15 faster

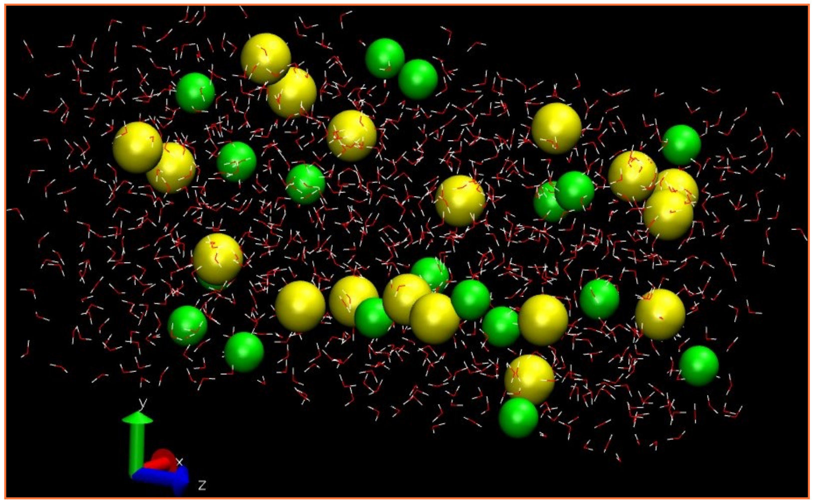


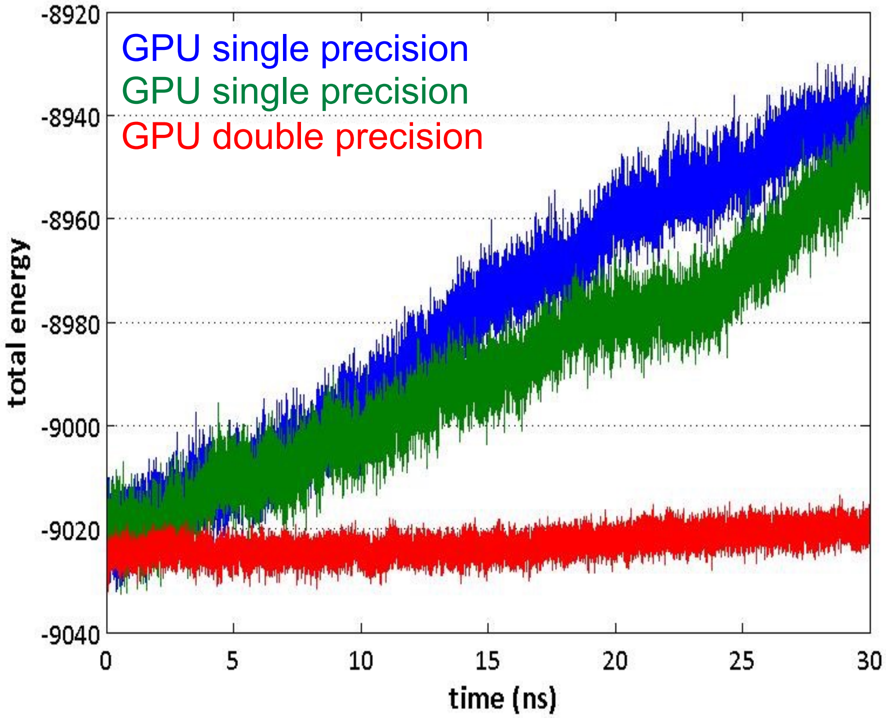


The Strange Case of Constant Energy MDs

- Enhancing performance of MD simulations allows simulations of larger time scales and length scales
- GPU computing enables large-scale MD simulation
  - Simulations exhibit unprecedented speed-up factors

MD simulation of NaI solution system containing 988 waters, 18 Na+, and 18 I–: GPU is X15 faster

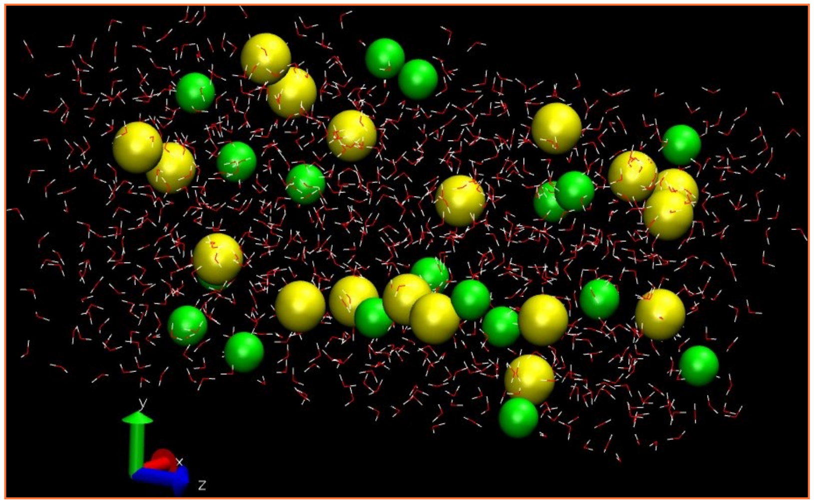


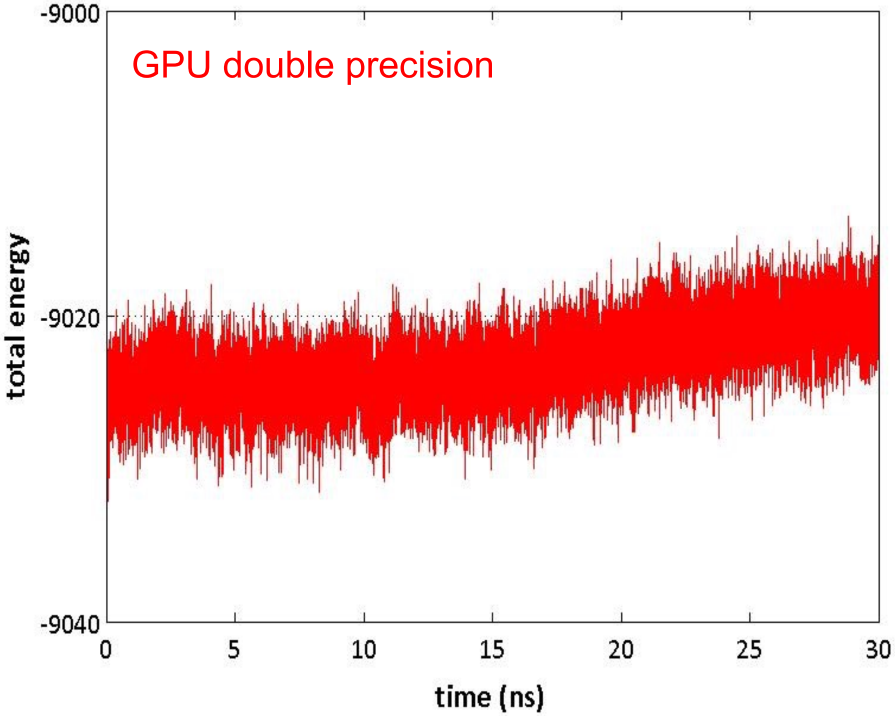


The Strange Case of Constant Energy MDs

- Enhancing performance of MD simulations allows simulations of larger time scales and length scales
- GPU computing enables large-scale MD simulation
  - Simulations exhibit unprecedented speed-up factors

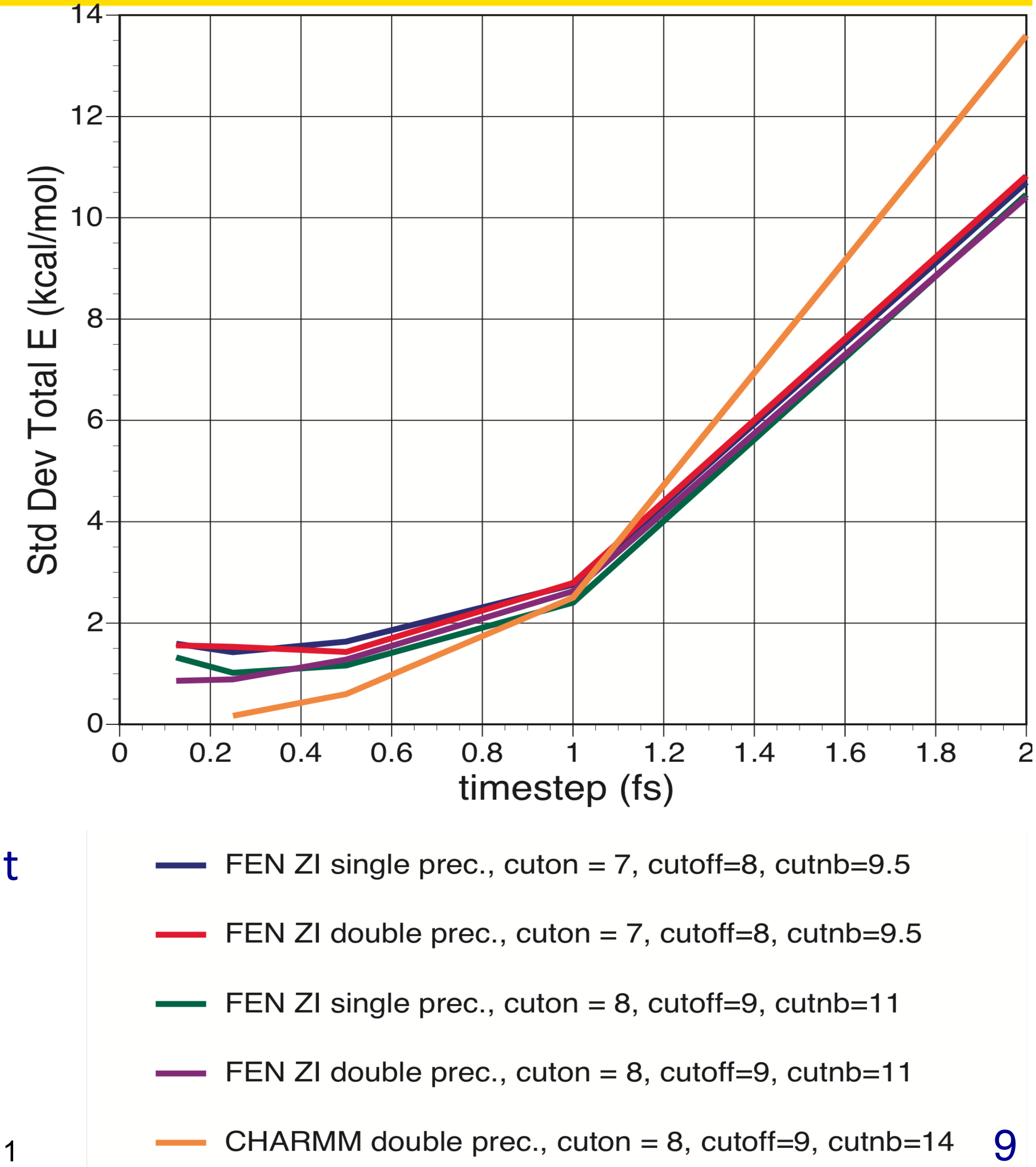
MD simulation of NaI solution system containing 988 waters, 18 Na+, and 18 I–: GPU is X15 faster





Just a Case of Code Accuracy?

- A plot of the **energy fluctuations versus time step size should follow an approximately logarithmic trend** [1]
- Energy fluctuations are proportional to time step size for large time step size
  - Larger than 0.5 fs
- A different behavior for step size less than 0.5 fs is consistent with results previously presented and discussed in other work [2]



[1] Allen and Tildesley, Oxford: Clarendon Press, (1987)
[2] Bauer et al., J. Comput. Chem. 32(3): 375 – 385, 2011

The Exascale Environment

	2010	2018	Factor Change
System peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Gf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 cpus	1,000 cpus	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4,444
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	100

DOE Exascale Initiative Roadmap, Architecture and Technology Workshop, San Diego, December, 2009.

From a recent talk of Lucy Nowell, DoE Program Director (Distinguished Speaker Lecture, University of Delaware, Oct 10, 2014)

The Exascale Environment

	2010	2018	Factor Change
System peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Gf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 cpus	1,000 cpus	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4,444
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	100

DOE Exascale Initiative Roadmap, Architecture and Technology Workshop, San Diego, December, 2009.

From a recent talk of Lucy Nowell, DoE Program Director (Distinguished Speaker Lecture, University of Delaware, Oct 10, 2014)

Discussion Outline

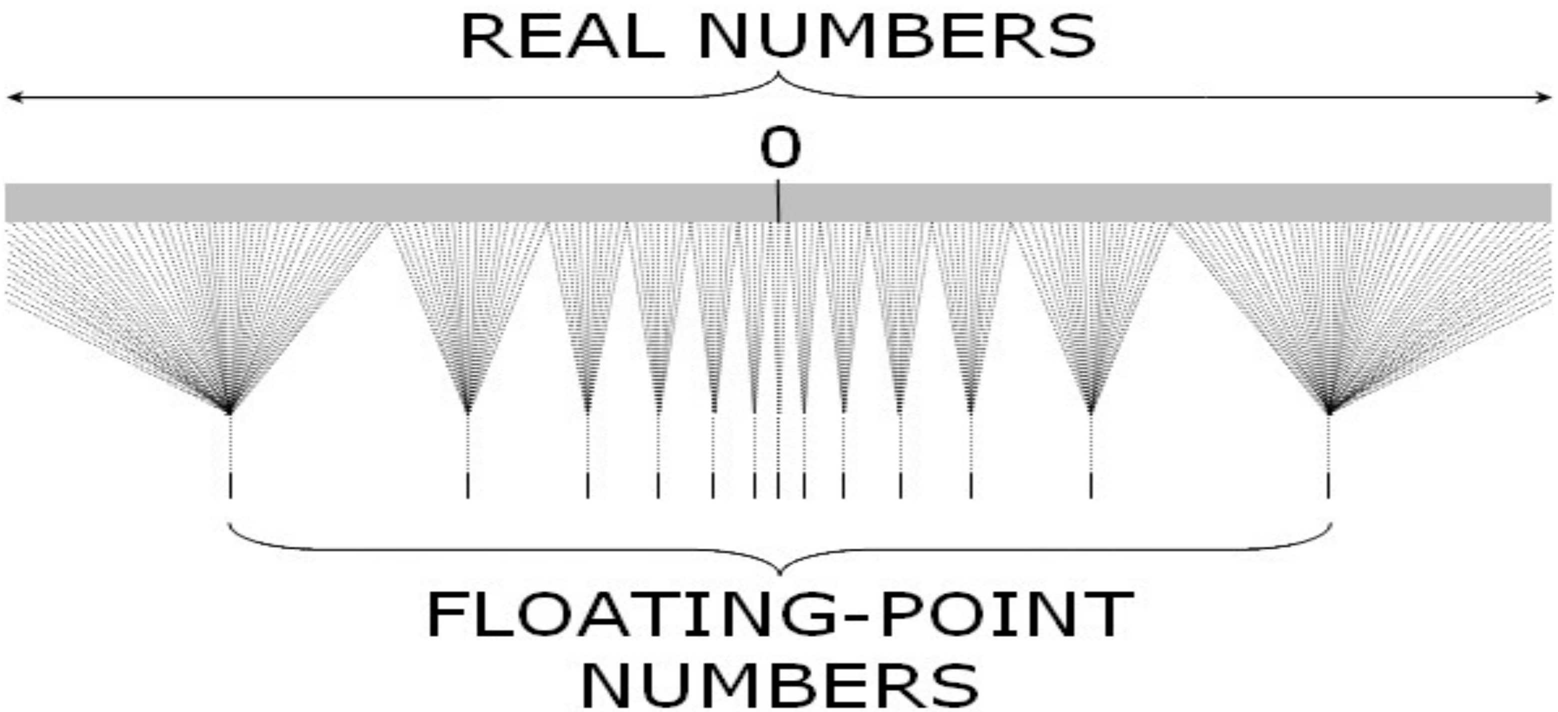
- Focus on reproducible accuracy of **global summation**
- Scientists demand increased reproducible accuracy
  - Must be **reproducible enough**
- Many approaches have been proposed
  - Must be **cost effective**
- Empirical results illustrate the need for runtime selection of reduction operators that ensure a given degree of reproducible accuracy

Discussion Outline

- Causes of loss of reproducibility
  - Well-known floating-point issues
  - Non-determinism at exascale
- Techniques for recovering reproducibility
  - Enhanced summation algorithms
- Empirical evaluation of summation algorithms' cost
- Quantifying reproducible accuracy
  - Identify key factors in variability of error accumulation
  - Study response of summation algorithms to those factors
- Lesson learned

Well-Known Problem

- The modeling of **finite-precision arithmetic** maps an **infinite set of real numbers** onto a finite set of machine numbers



http://cs.smith.edu/dftwiki/index.php/CSC231 An Introduction to Fixed- and Floating-Point Numbers

Simple Example

$$a = 10^9,\ b = -10^9,\ c = 10^{-9}$$

Summation order 1

$$(a+b)+c = (10^9 - 10^9) + 10^{-9}$$

Summation order 2

$$a+(b+c) = 10^9 + (-10^9 + 10^{-9})$$

Simple Example

$$a = 10^9,\ b = -10^9,\ c = 10^{-9}$$

Summation order 1

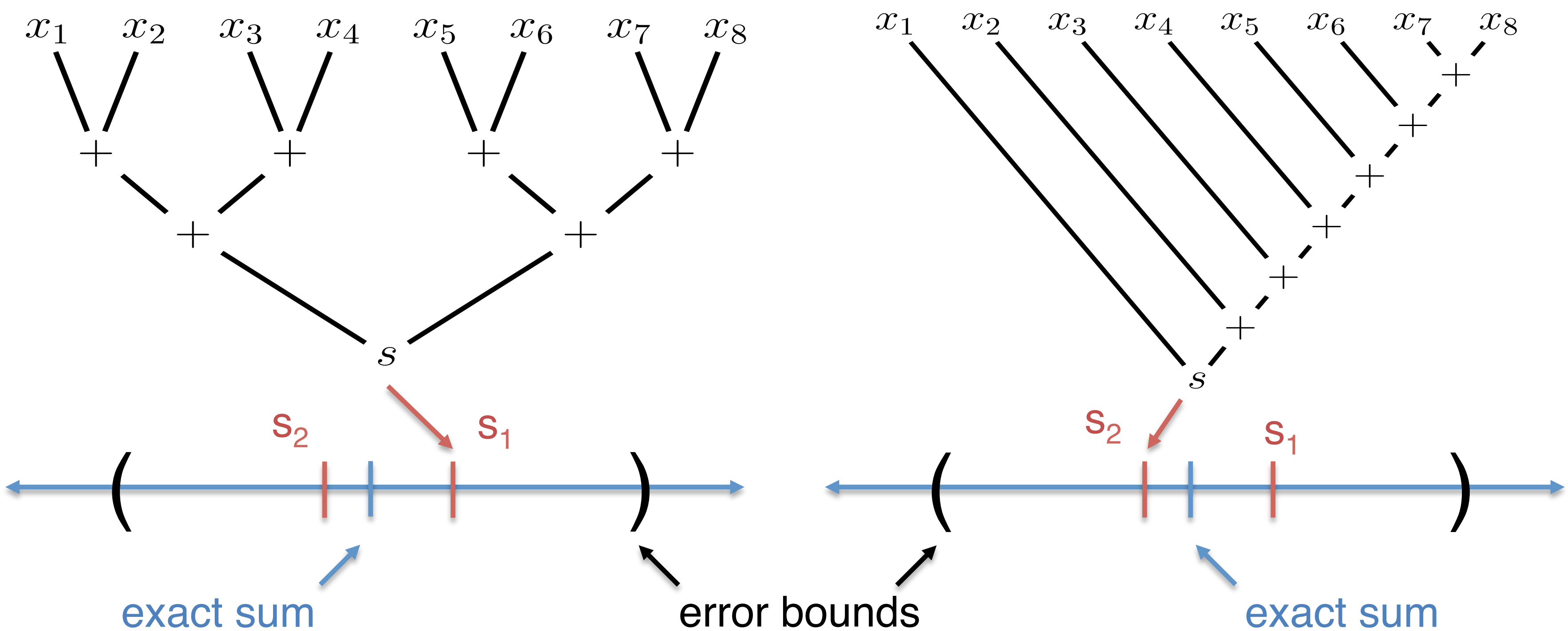
$$(a + b) + c = (10^9 - 10^9) + 10^{-9} = 10^{-9}$$

Summation order 2

$$a + (b + c) = 10^9 + (-10^9 + 10^{-9}) = 0$$

Non-Determinism at Extreme Scale

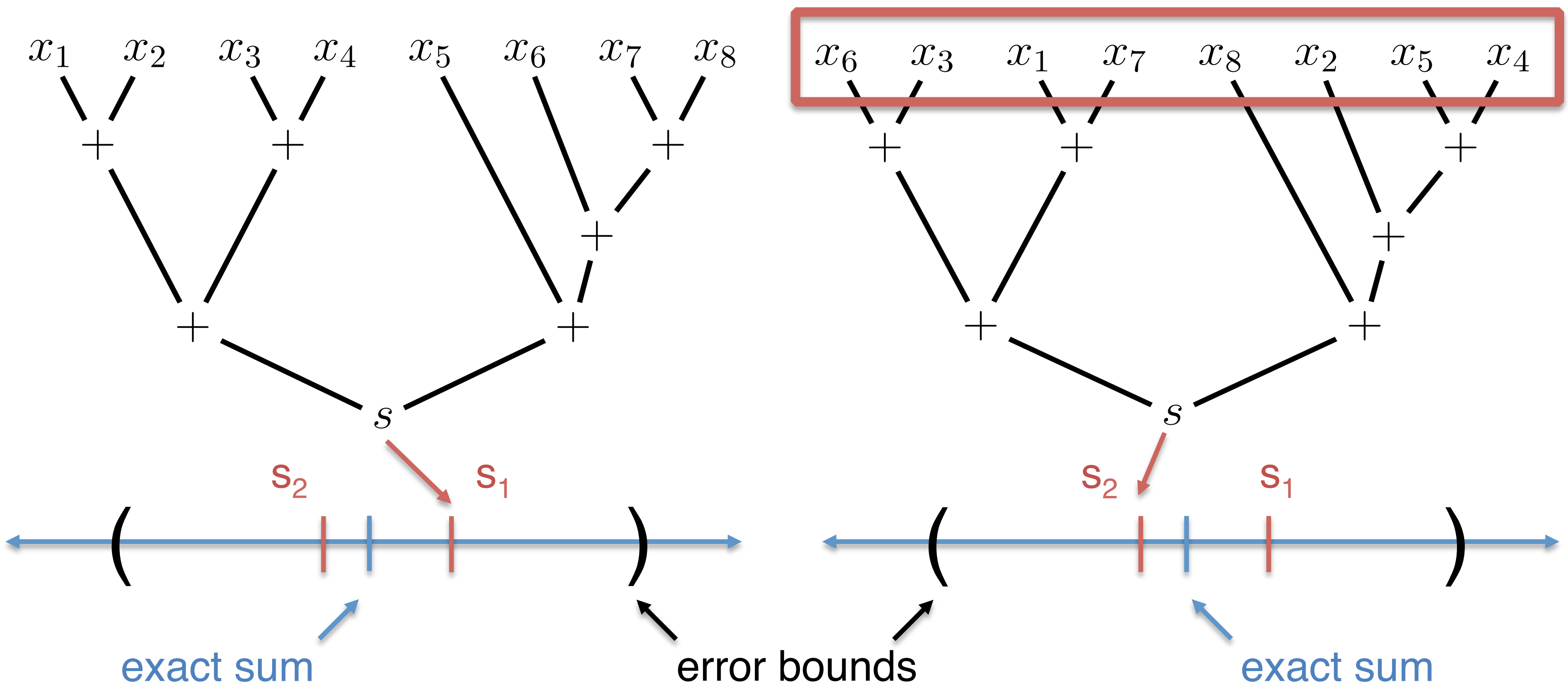
Reduction tree shape



Causes include: dynamic task scheduling and fault recovery

Non-Determinism at Extreme Scale

Arrangement of operands

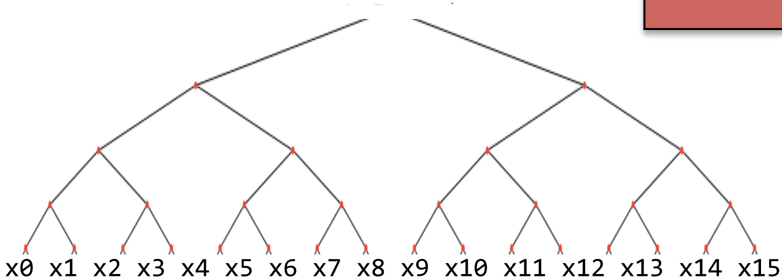


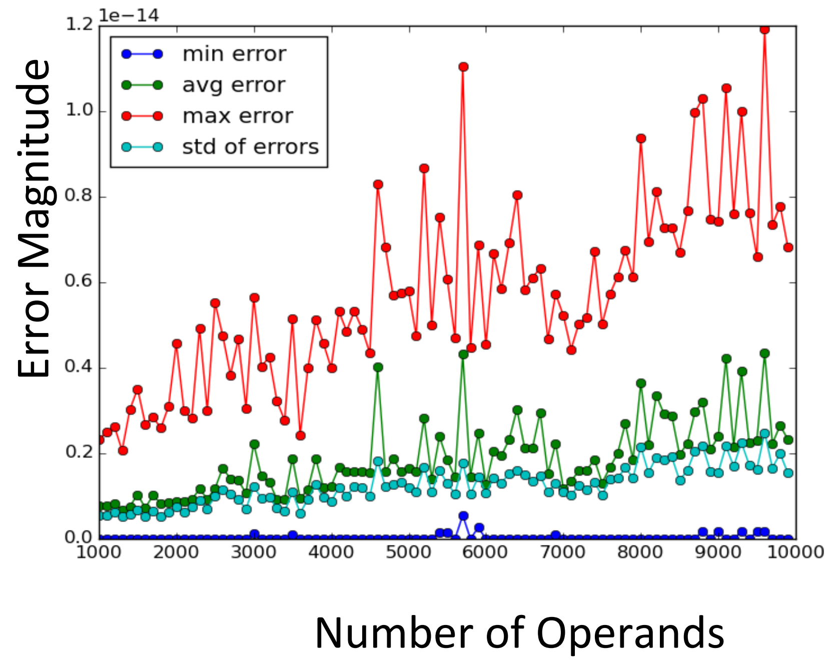
Causes include: dynamic task scheduling and fault recovery

Non-Associativity + Non-Determinism

- **No control** on the way N floating-point numbers are assigned to N threads

$$\sum_{i=1}^{n} x_i$$

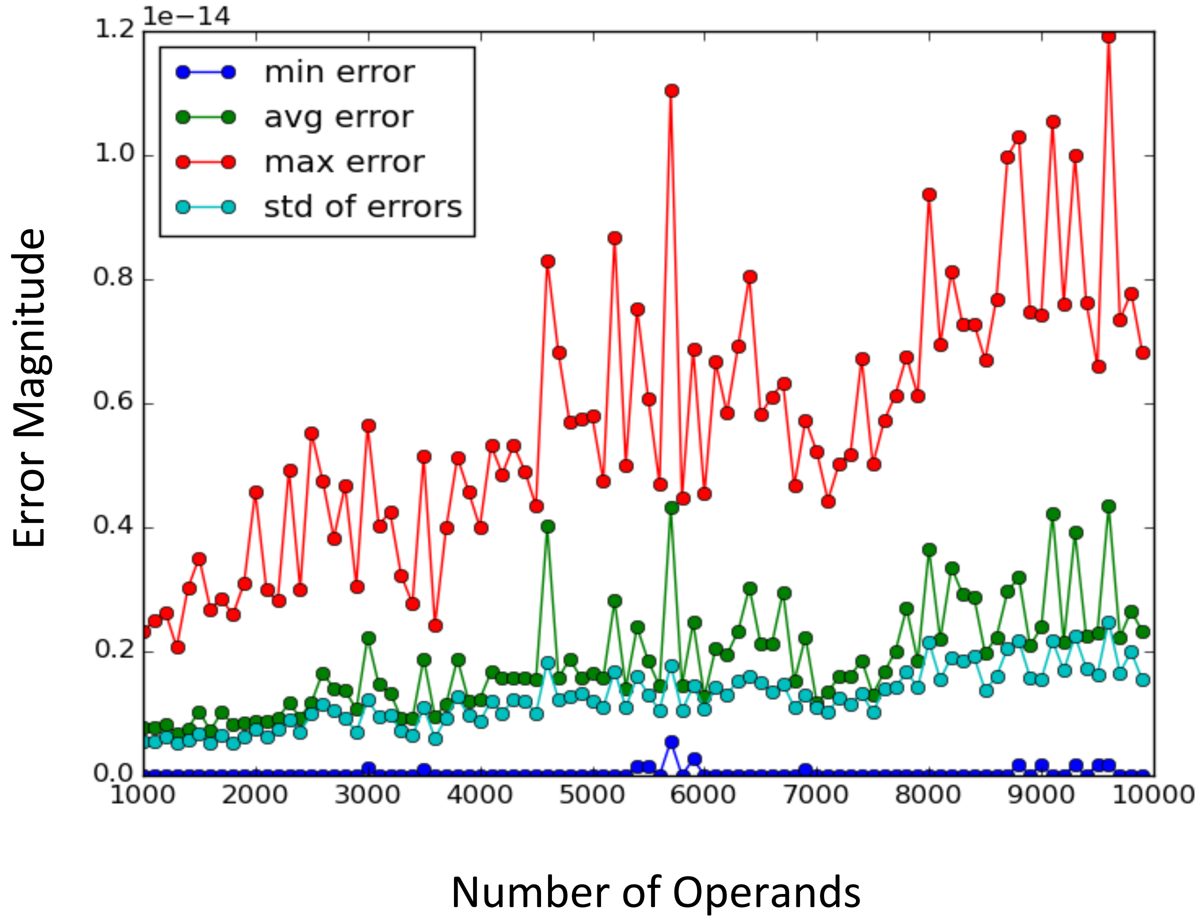




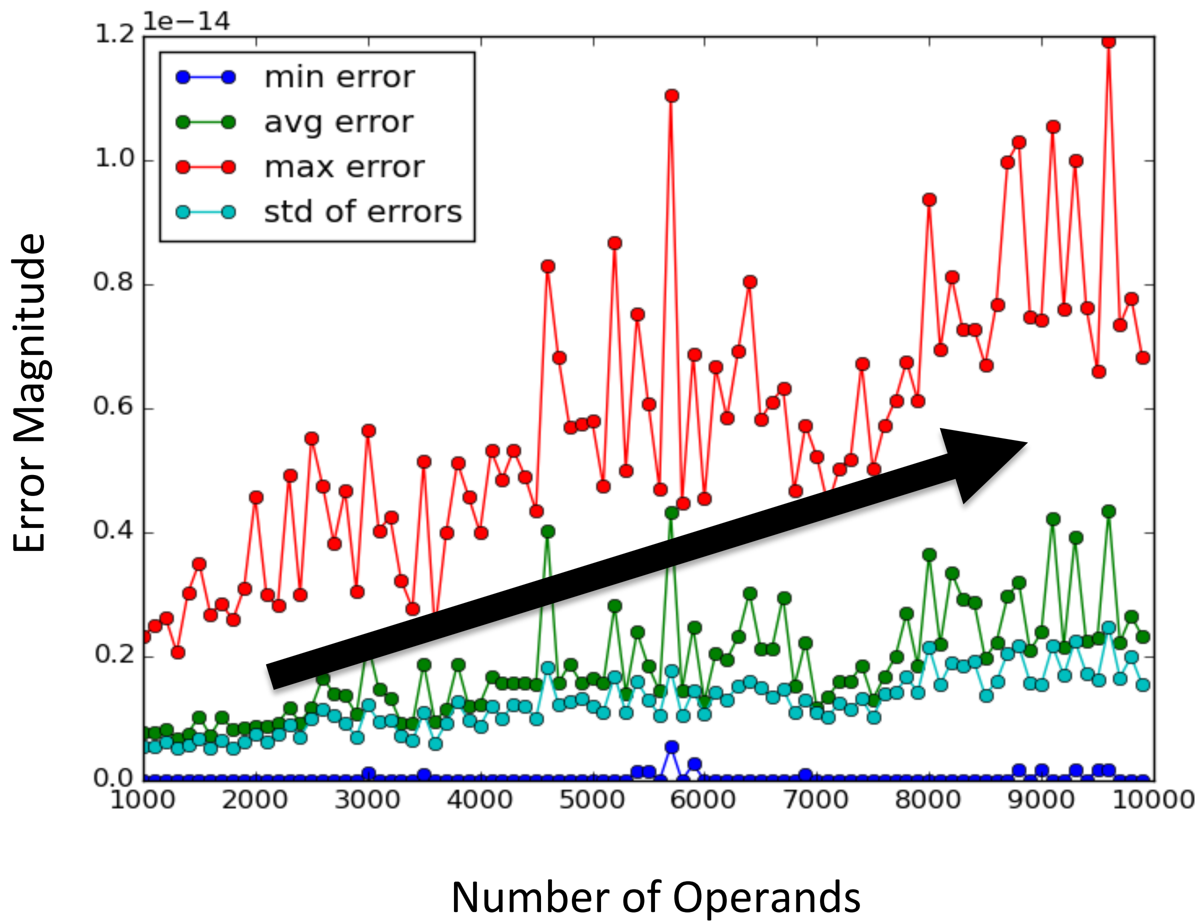
- Different thread orders cause round-off errors to accumulate in different ways, leading to different summation results

19

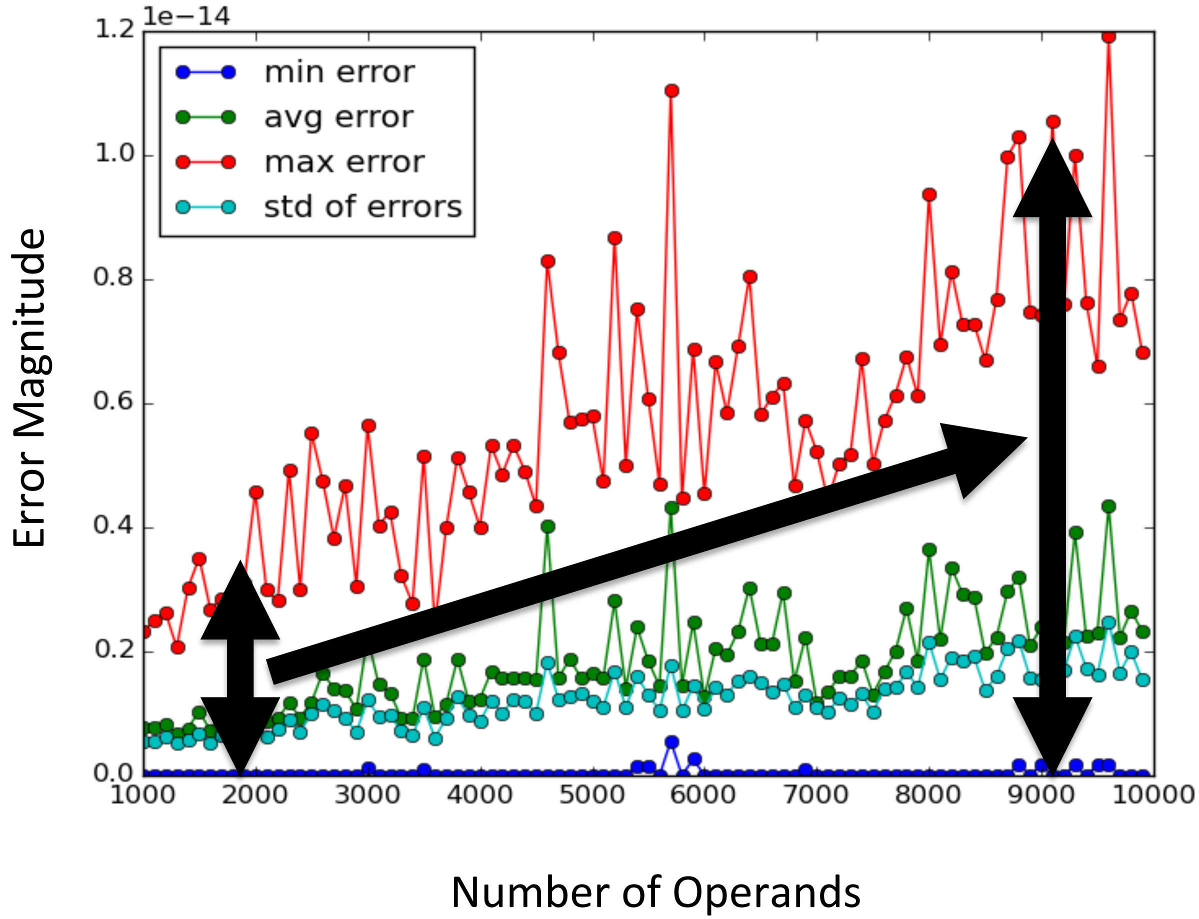
Non-Associativity + Non-Determinism



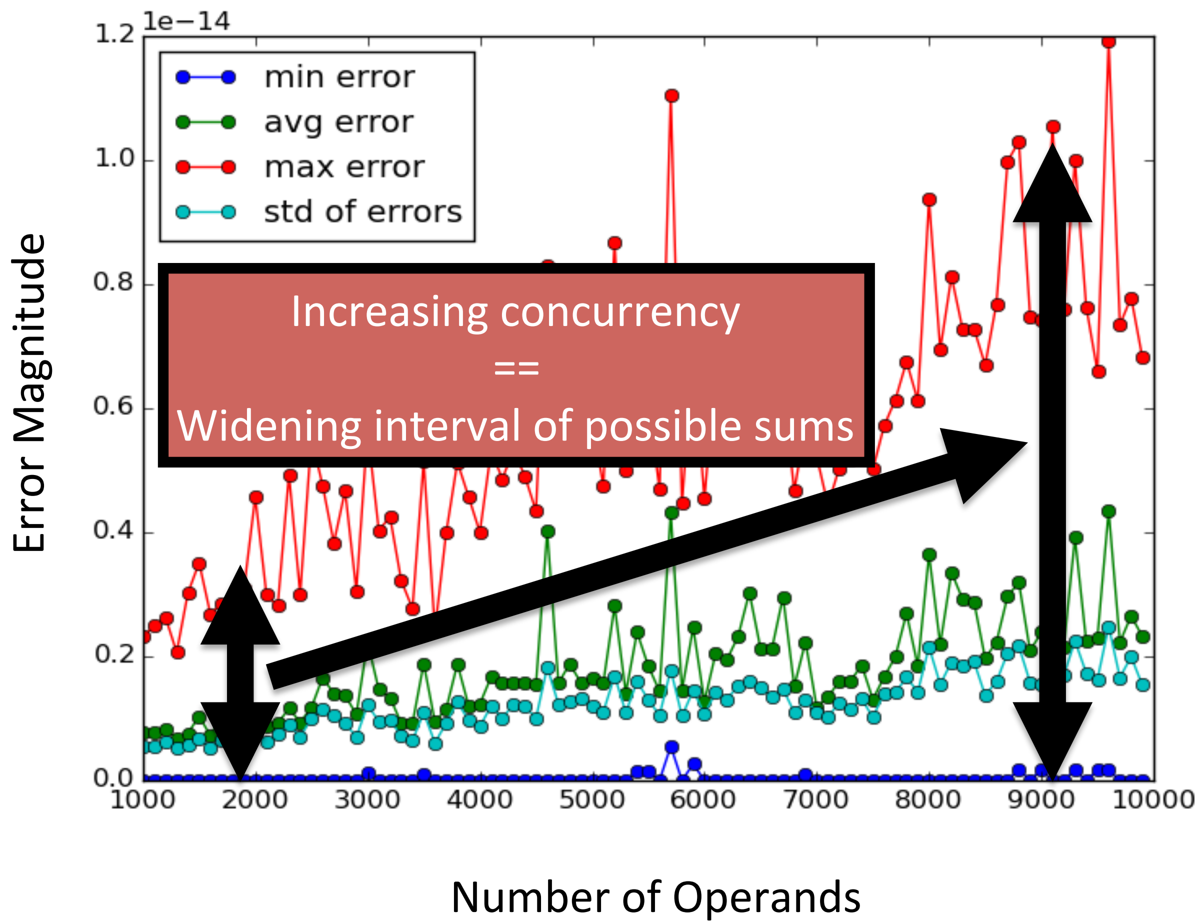
Non-Associativity + Non-Determinism



Non-Associativity + Non-Determinism

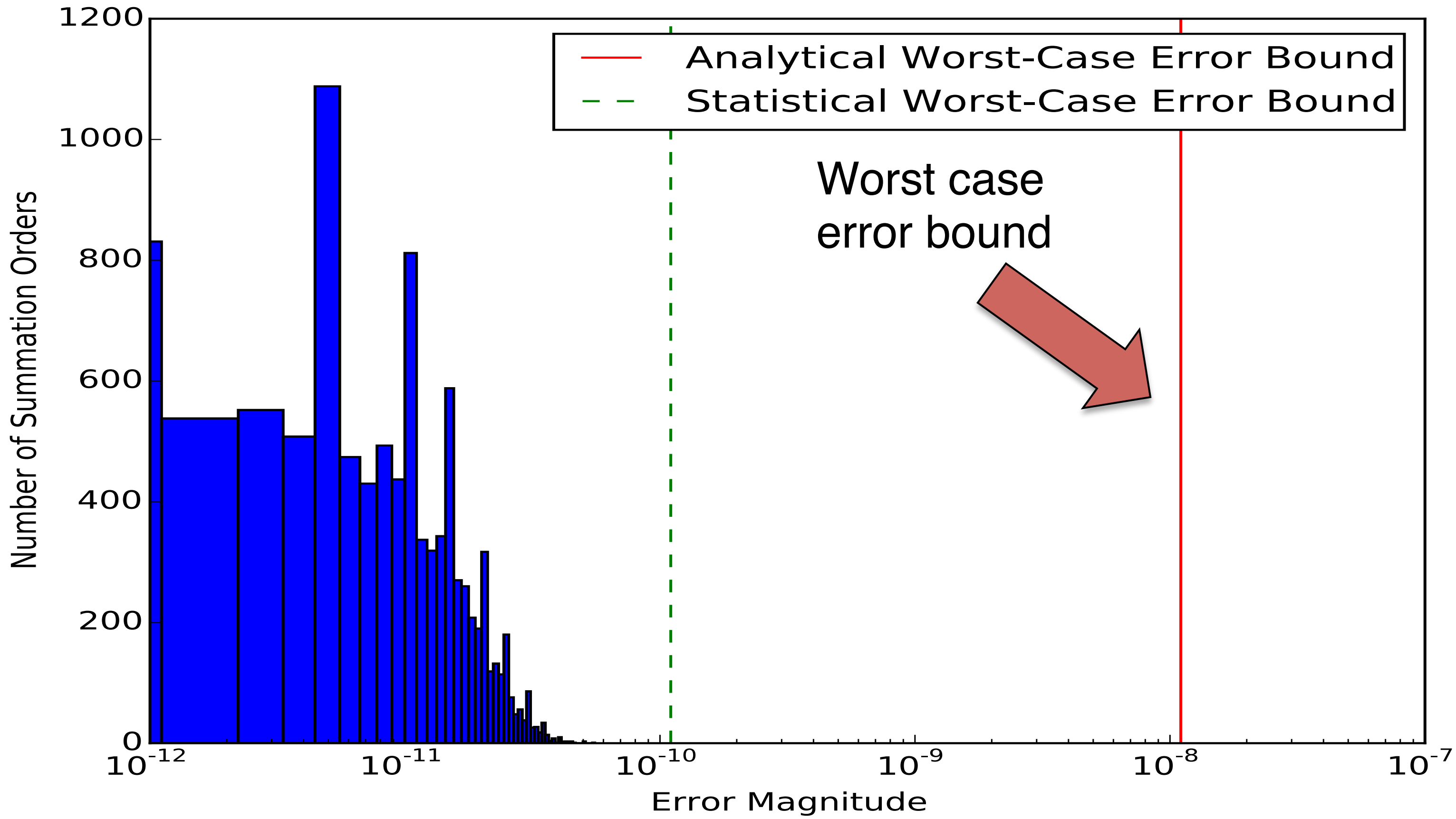


Non-Associativity + Non-Determinism



Inadequacy of Conventional Wisdom

- In practice error bounds are overly pessimistic (i.e., usually N * ε << 1) and thus unreliable predictors



Techniques for Recovering Reproducibility

- Fixed reduction order
  - Ensuring that all floating-point operations are evaluated in the same order from run to run
- Increased precision numerical types
  - Mixed precision - e.g. use higher-precision types for sensitive computations and standard types for less sensitive computations
- Interval arithmetic
  - Replace floating-point types with custom types representing finite-length intervals of real numbers
- Enhanced Summation Algorithms
  - Compensated summation e.g., Kahan and composite precision
  - Pre-rounded reproducible summation

Techniques for Recovering Reproducibility

- Fixed reduction order
  - Ensuring that all floating-point operations are evaluated in the same order from run to run
- Increased precision numerical types
  - Mixed precision - e.g. use higher-precision types for sensitive computations and standard types for less sensitive computations
- Interval arithmetic
  - Replace floating-point types with custom types representing finite-length intervals of real numbers
- Enhanced summation algorithms
  - Compensated summation e.g., Kahan and composite precision
  - Pre-rounded reproducible summation

UNIVERSITY of DELAWARE

Techniques for Recovering Reproducibility

- Fixed reduction order
  - Ensuring that all fl g-point operations are evaluated in the same order from run to
- Increased precision n cal types
  - Mixed precision - e.g. us r precision types for sensitive computations and standar for less sensitive computations
- Interval arithmetic
  - Replace floating-point types with n types representing finite-length intervals of real numbers
- Enhanced summation algorithms
  - Compensated summation e.g., Kahan and composite precision
  - Pre-rounded reproducible summation

TOO COSTLY!!!

Standard Summation: Definition

```
function StandardSummation(input)

var sum = 0.0

for i = 1 to input.length do {
    sum = sum + input[i]
}
return sum
```

Kahan Summation: Definition

```
function KahanSum(input)
    var sum = 0.0
    var c = 0.0
    for i = 1 to input.length do {
        var y = input[i] - c
        var t = sum + y
        c = (t - sum) - y
        sum = t
    }
    return sum
```

Holds error (var c = 0.0)

Capture error & add to operand on next iteration (c = (t - sum) - y → var y = input[i] - c)

Kahan "Further Remarks on Reducing Truncation Errors" (1964)

Composite Precision: Definition

```
struct var2{
    var val          Value or result
    var err          Error approximation
}

var2 sum, v
var t = 0.0

for i = 1 to input.length do {
    sum.val = sum.val + v.val
    t = sum.val − v.val
    sum.err = sum.val   −
          (sum.val − t) + (v.val − t) +
          sum.err  + v.err
}
retun sum
```

Error carried through each operation

Taufer et al." Improving Numerical Reproducibility and Stability in Large-Scale Numerical Simulations on GPUs" (2010)

Pre-rounded Summation: Definition

v1 + v2:

Until error < threshold

select extractor *M*

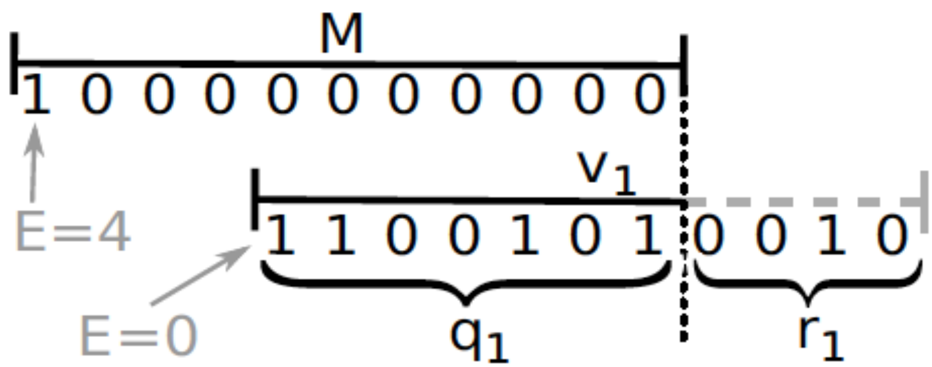
$q1 = (v1 + M) - M$

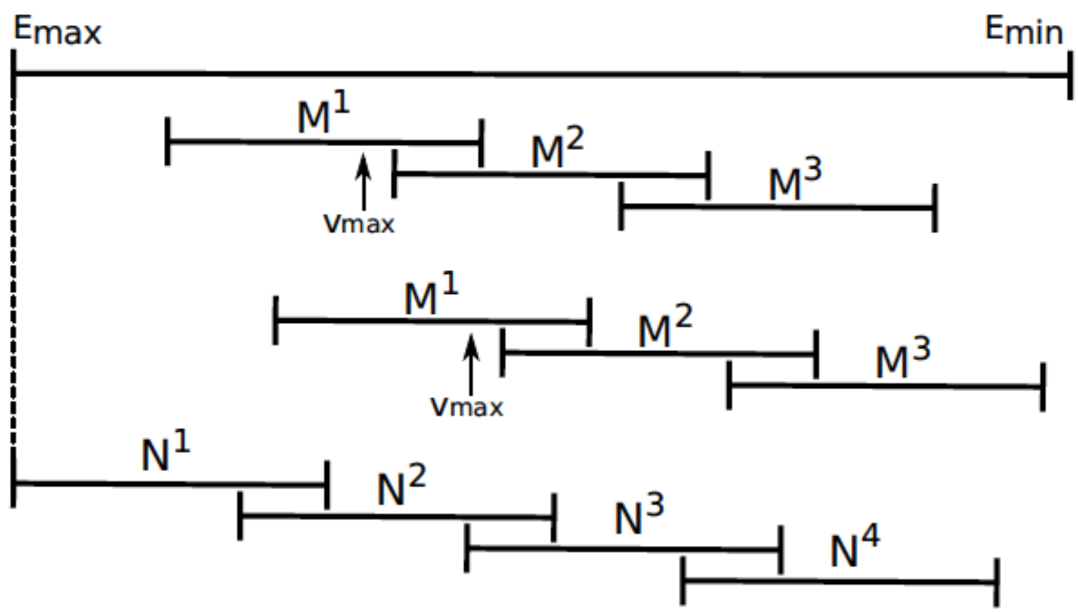
$q2 = (v2 + M) - M$

$q1 + q2$

$r1 = v1 - q1$

$r2 = v2 - q2$





Demmel and Nguyen "Parallel Reproducible Summation" (2014)

Arteaga, Hoefler et al. "*Designing Bit-*
Portable High-Performance Applicatio

Techniques for Reproducible Summation

- Fixed reduction order
  - Ensuring that all fl g-point operations are evaluated in the same order from r un
- Increased precision n ical types
  - Mixed precision - e.g. us ubles for sensitive computations and floats everywhere else
- Interval arithmetic
  - Replace floating-point types with n types representing finite-length intervals of real numbers
- Enhanced summation algorithms
  - Compensate composite precision
  - Pre-rounded summation

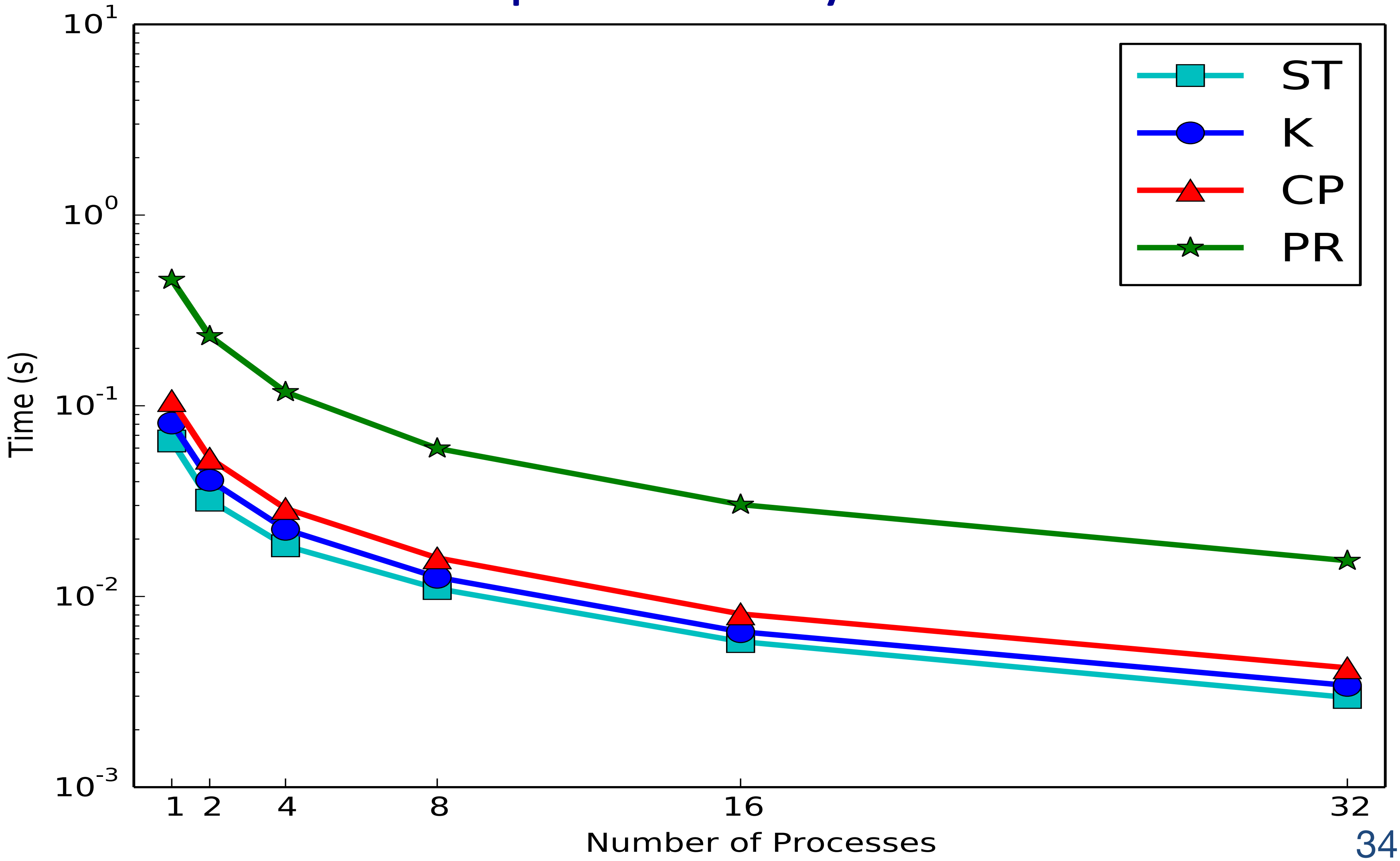
TOO COSTLY!!!

HOW COSTLY?

Empirical Study: Cost

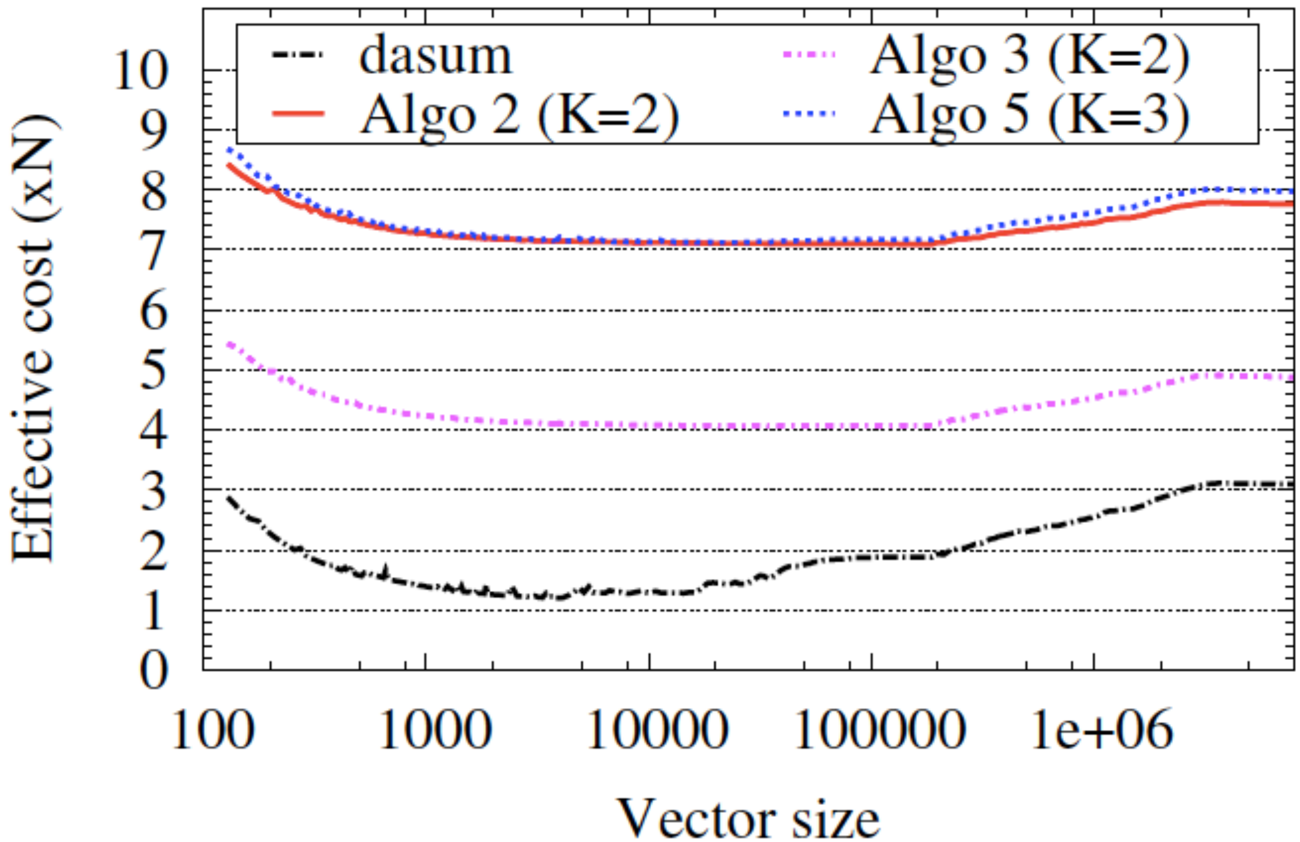
- Emulate simulation execution
  - Run parallel sum of 1M doubles using MPI
  - Perform partial sums independently
  - Reduce by global sum with MPI_REDUCE
- Summation algorithms tested
  - Standard (ST)
  - Kahan (K)
  - Composite Precision (CP)
  - Pre-rounded (PR)

Empirical Study: Cost



Error-free Transformations: Times

- 2-fold pre-rounding versions and varying vector sizes



~X7

~X4

Intel MKL library

Demmel and Nguyen “Parallel Reproducible Summation” (2013)

Techniques for Reproducible Summation

- Fixed reduction order
  - Ensuring that all fl g-point operations are evaluated in the same order from r un
- Increased precision n ical types
  - Mixed precision - e.g. us ubles for sensitive computations and floats everywhere else
- Interval arithmetic
  - Replace floating-point types with m types representing finite-length intervals of real numbers
- Enhanced su
  - Compensat posite precision
  - Pre-rounded

TOO COSTLY!!!

How reproducible?

How accurate?

Empirical Study: Reproducible Accuracy

- Emulate sums expected in exascale simulations
  - Shuffling summation order emulates nondeterministic reduction tree
- Measure sensitivity of summation algorithms to:
  - Changes in summation order
  - Mathematical properties of summands

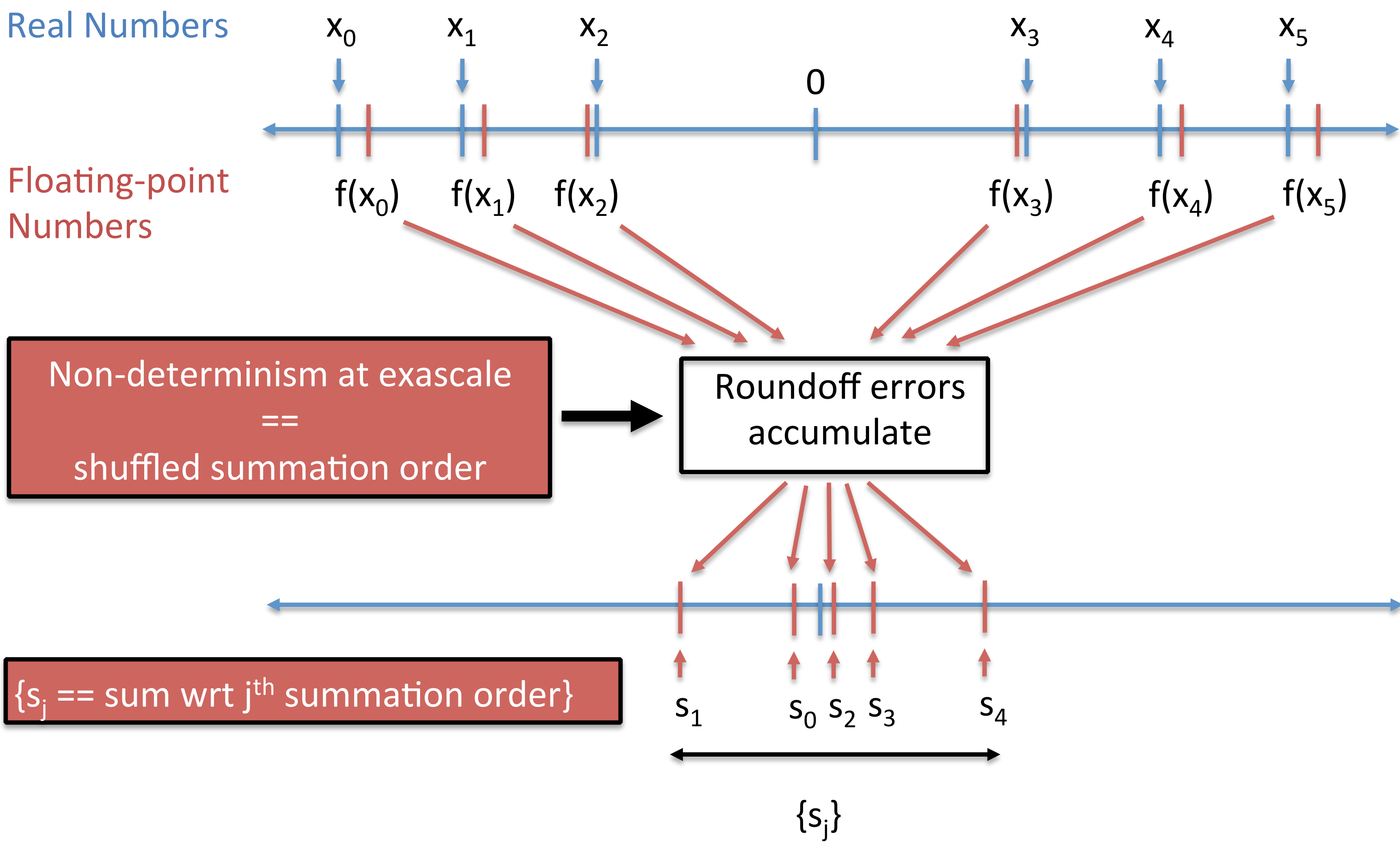
Empirical Study: Reproducible Accuracy

- Emulate sums expected in exascale simulations
  - Shuffling summation order emulates nondeterministic reduction tree
- Measure sensitivity of summation algorithms to:
  - Changes in summation order
  - Mathematical properties of summands
- Interpret width of result interval as sensitivity

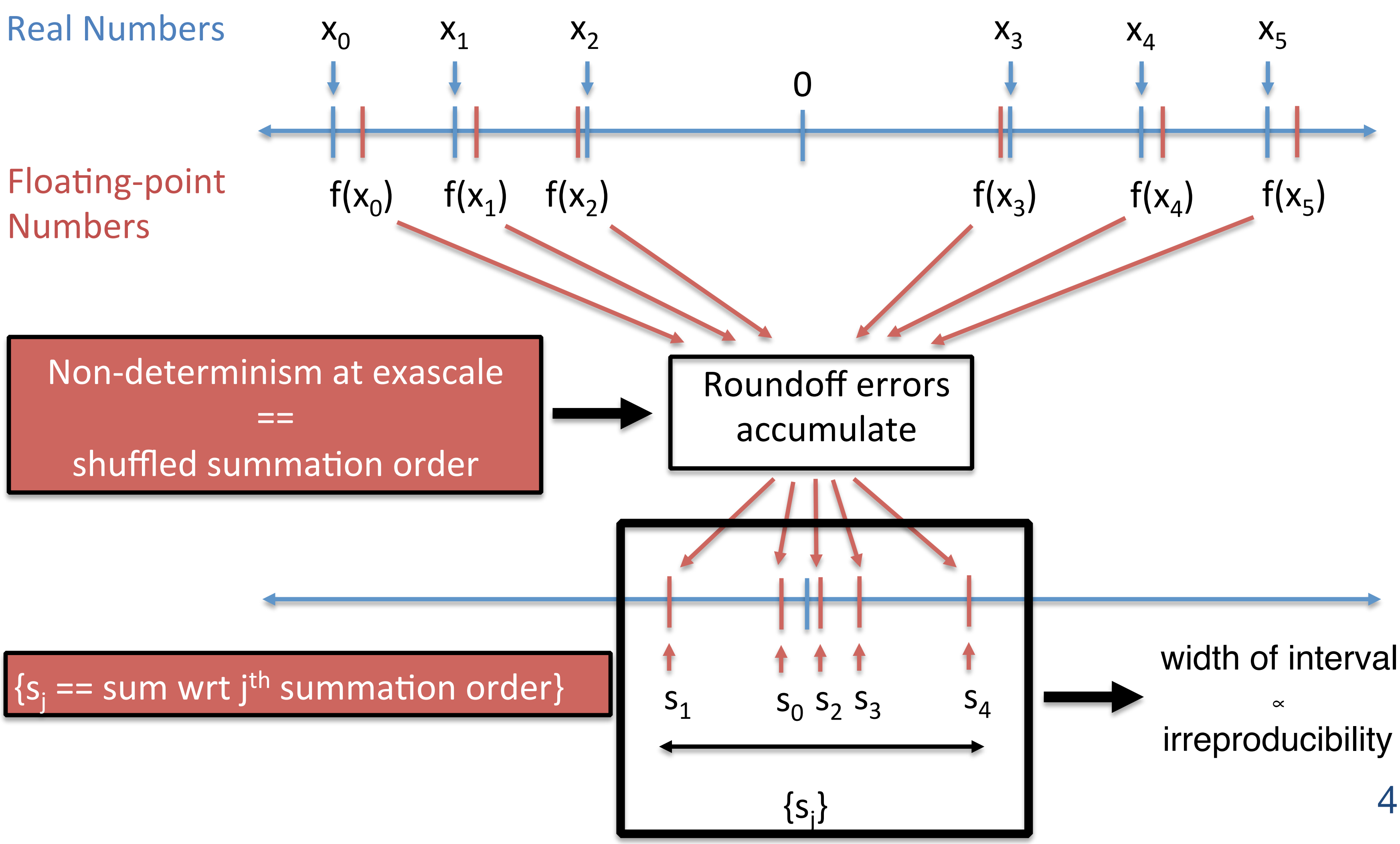
Empirical Study: Reproducible Accuracy

- Emulate sums expected in exascale simulations
  - Shuffling summation order emulates nondeterministic reduction tree
- Measure sensitivity of summation algorithms to:
  - Changes in summation order
  - Mathematical properties of summands
- Interpret width of result interval as sensitivity
- Test summation algorithms: Standard (ST), Kahan (K), Composite-Precision (CP), Pre-rounded (PR)

Emulating Exascale Scenarios



Emulating Exascale Scenarios



Characterizing Sets of Summands

$$\frac{|S_{\text{exact}} - S_{\text{j}}|}{|S_{\text{exact}}|} \leq (n-1) \cdot u \cdot \frac{\sum_{i=1}^{n} |x_i|}{|\sum_{i=1}^{n} x_i|}$$

Characterizing Sets of Summands

$$\frac{|S_{\text{exact}} - S_{\text{j}}|}{|S_{\text{exact}}|} \leq (n-1) \cdot u \cdot \frac{\sum_{i=1}^{n} |x_i|}{|\sum_{i=1}^{n} x_i|}$$

Critical Parameters

- Size: n
- Condition number: k
- Dynamic range: dr

Taxonomy of Values

Sample set of values	dr	k
{1.23e+32, 1.35e+32, 2.37e+32, 3.54e+32}	0	1
{1.23e-32, 1.35e-32, 2.37e-32, 3.54e-32}	0	1
{-1.23e+16, -1.35e+16, -2.37e+16,-3.54e+16}	0	1
{2.37e+16, 3.41e+8, 4.32e+8, 8.14e+16}	8	1
{3.14e+32, 1.59e+16, 2.65e+18, 3.58e+24}	16	1
{2.505e+2, 2.5e+2, -2.495e+2, -2.5e+2}	0	1000
{5.00e+2, 4.99999e-1, 1.0e-6, -4.995e+2}	8	1000
{5.00e+2, 4.99...99e-1, 1.0e-14, -4.995e+2}	16	1000
{3.14e+8, 1.59e+8, -3.14e+8, -1.59e+8}	0	∞
{3.14e+4, 1.59e-4, -3.14e+4, -1.59e-4}	8	∞
{3.14e+8, 1.59e-8, -3.14e+8, -1.59e-8}	16	∞

Taxonomy of Values

Sample set of values	dr	k
{1.23e+32, 1.35e+32, 2.37e+32, 3.54e+32}	0	1
{1.23e-32, 1.35e-32, 2.37e-32, 3.54e-32}	0	1
{-1.23e+16, -1.35e+16, -2.37e+16,-3.54e+16}	0	1
{2.37e+16, 3.41e+8, 4.32e+8, 8.14e+16}	8	1
{3.14e+32, 1.59e+16, 2.65e+18, 3.58e+24}	16	1
{2.505e+2, 2.5e+2, -2.495e+2, -2.5e+2}	0	1000
{5.00e+2, 4.99999e-1, 1.0e-6, -4.995e+2}	8	1000
{5.00e+2, 4.99...99e-1, 1.0e-14, -4.995e+2}	16	1000
{3.14e+8, 1.59e+8, -3.14e+8, -1.59e+8}	0	∞
{3.14e+4, 1.59e-4, -3.14e+4, -1.59e-4}	8	∞
{3.14e+8, 1.59e-8, -3.14e+8, -1.59e-8}	16	∞

Characterizing Sets of Summands

$$\frac{|S_{\text{exact}} - S_{\text{j}}|}{|S_{\text{exact}}|} \leq (n-1) \cdot u \cdot \frac{\sum_{i=1}^{n} |x_i|}{|\sum_{i=1}^{n} x_i|}$$

Critical Parameters

- Size: n
- Condition number: k
- Dynamic range: dr

→

Proxy for…

Concurrency

Subtractive cancellation

Alignment error

Empirical Study: Results

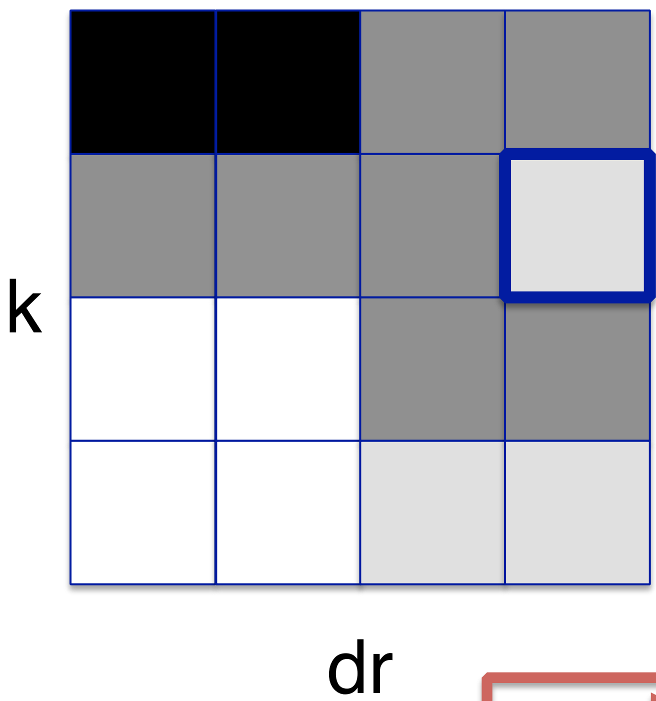
- Varying the shape of the reduction tree
  - Ill-conditioned, high dynamic range values
  - Balanced vs. unbalanced reduction trees
- Error variability within the parameter space
  - n vs. k
  - n vs. dr
  - k vs. dr
- Summation algorithm selection
  - Given a variability threshold, which algorithm is needed

UNIVERSITY of DELAWARE

Empirical Studies of Reproducible Accuracy

- Varying the shape of the reduction tree
  - Ill-conditioned, high dynamic range values
  - Balanced vs. unbalanced reduction trees
- Error variability within the parameter space
  - n vs. k
  - n vs. dr
  - k vs. dr
- Summation algorithm selection
  - Given a variability threshold, which algorithm is needed

Visualizing Degree of Reproducible Accuracy



$\{x1, x2, \ldots\ xn\}$ — Values

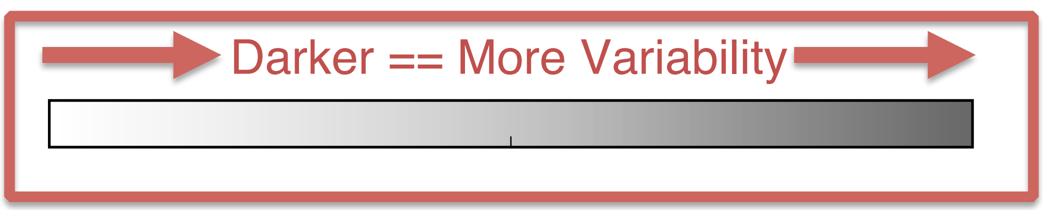
$S_\sigma = \sum_1^n x_{\sigma(i)}$ σ – perm. of [n] — Sum of shuffled values

$\{S_{\sigma\_1}, S_{\sigma\_2}, S_{\sigma\_3} \ldots\ldots S_{\sigma\_100}\}$ — Multiple sums of multiple permutations

$\{\varepsilon_{\sigma\_1}, \varepsilon_{\sigma\_2}, \varepsilon_{\sigma\_3} \ldots\ldots \varepsilon_{\sigma\_100}\}$ — Errors w/r/t GNU MPFR result

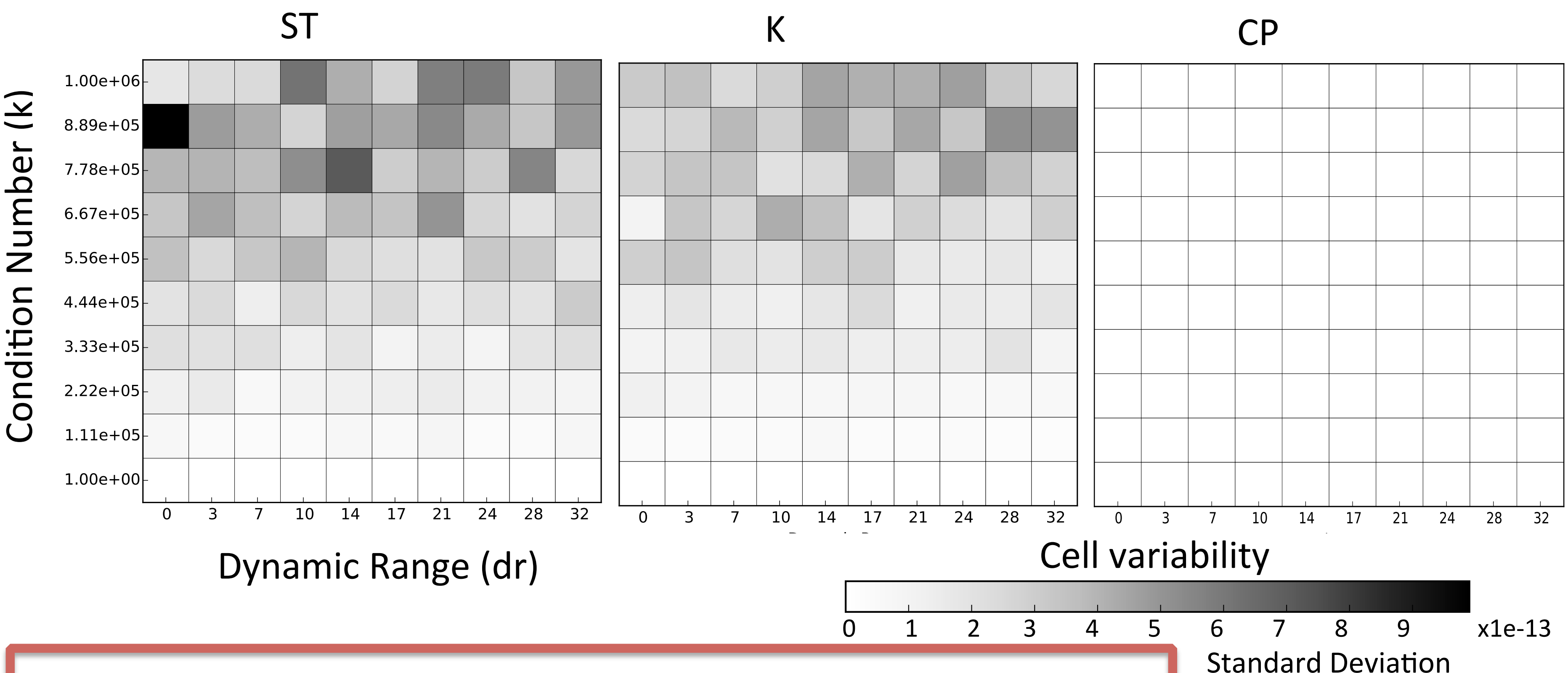
.....

error variability — Error variability



Condition Number vs. Dynamic Range

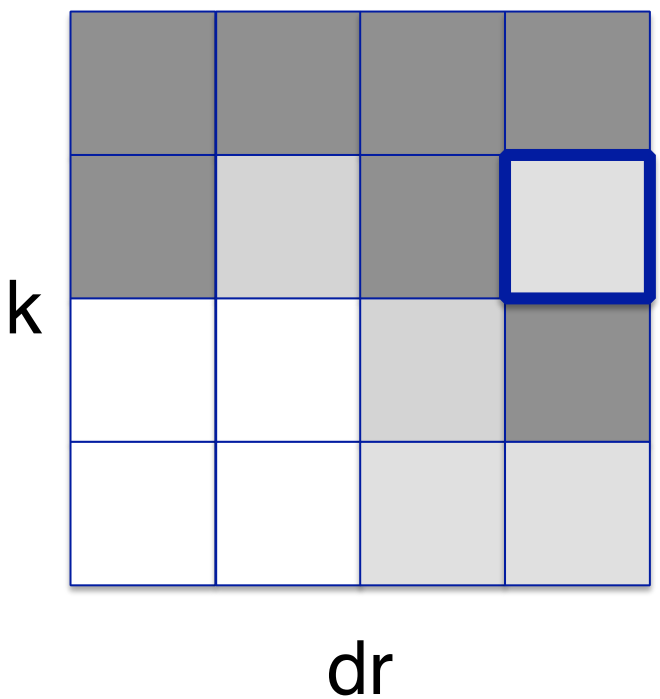
Parameter ranges: N = 10^6, k $\in$ [1, 10^6], dr $\in$ [0, 32]



Compensated summation incrementally improves reproducible accuracy

Empirical Studies of Reproducible Accuracy

- Varying the shape of the reduction tree
  - Ill-conditioned, high dynamic range values
  - Balanced vs. unbalanced reduction trees
- Error variability within the parameter space
  - n vs. k
  - n vs. dr
  - k vs. dr
- Summation algorithm selection
  - Given a variability threshold, which algorithm is needed?



$\{x1, x2, \ldots\ xn\}$ — Values

$S_\sigma = \sum_1^n x_{\sigma(i)}$ σ – perm. of [n] — Sum of shuffled values

$\{S_{\sigma\_1}, S_{\sigma\_2}, S_{\sigma\_3} \ldots\ S_{\sigma\_100}\}$ — Multiple sums of multiple permutations

$\{\varepsilon_{\sigma\_1}, \varepsilon_{\sigma\_2}, \varepsilon_{\sigma\_3} \ldots\ \varepsilon_{\sigma\_100}\}$ — Errors w/r/t GNU MPFR result

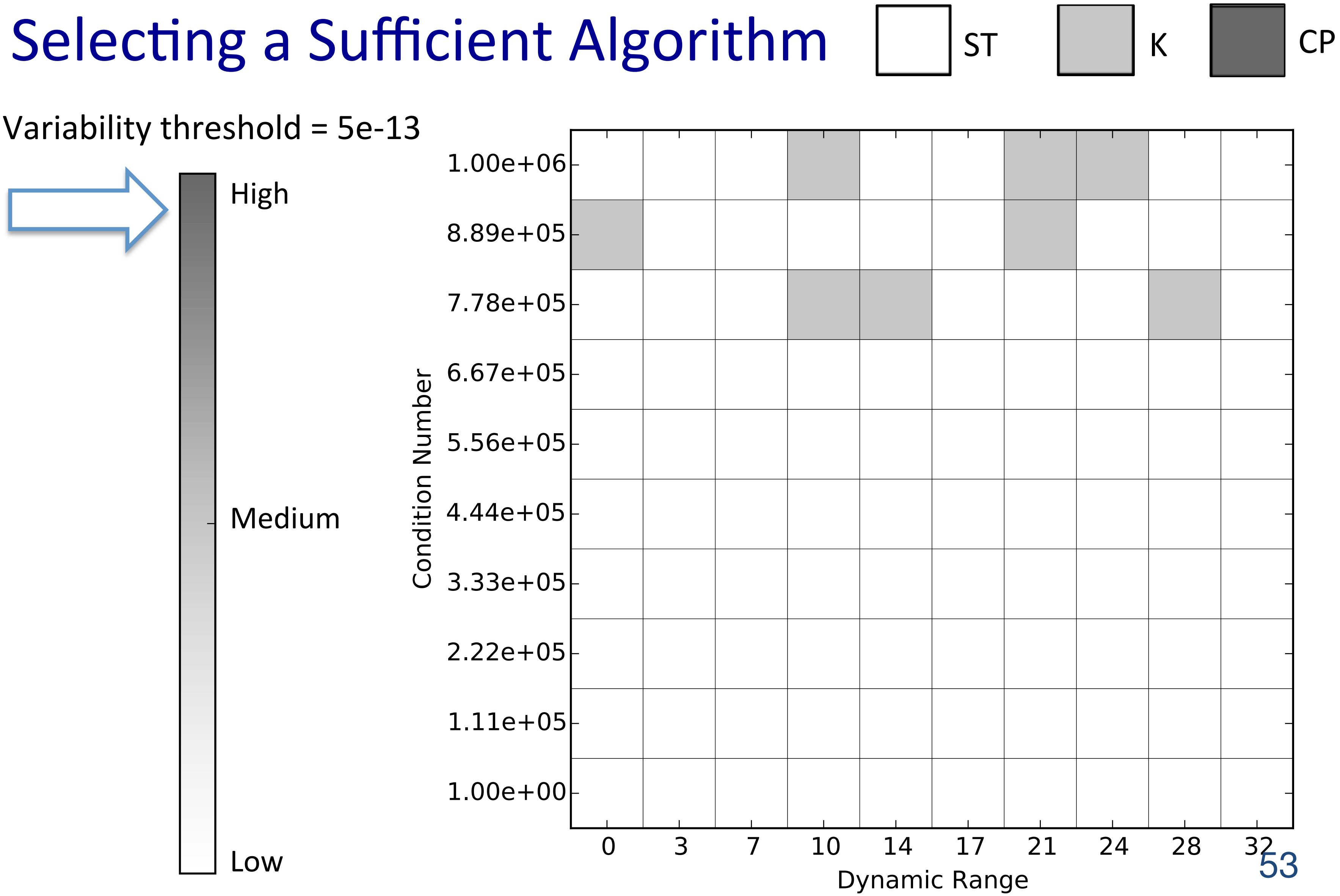
..... → error variability — Error variability

Cell shade == algorithm keeps variability below threshold

ST K CP

Selecting a Sufficient Algorithm

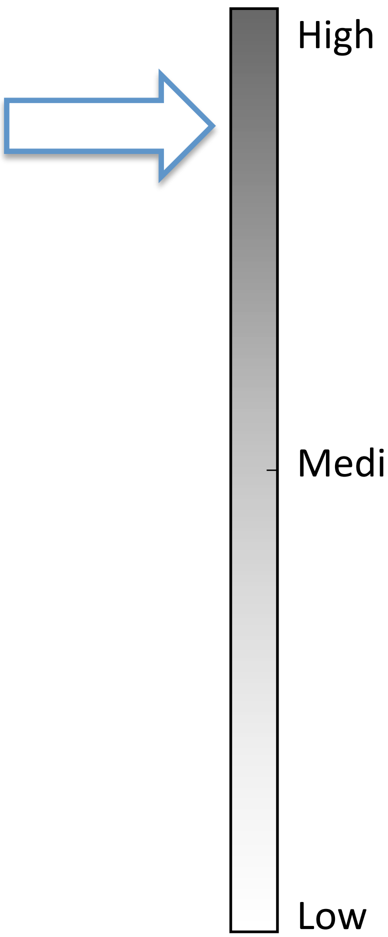
Variability threshold = 5e-13

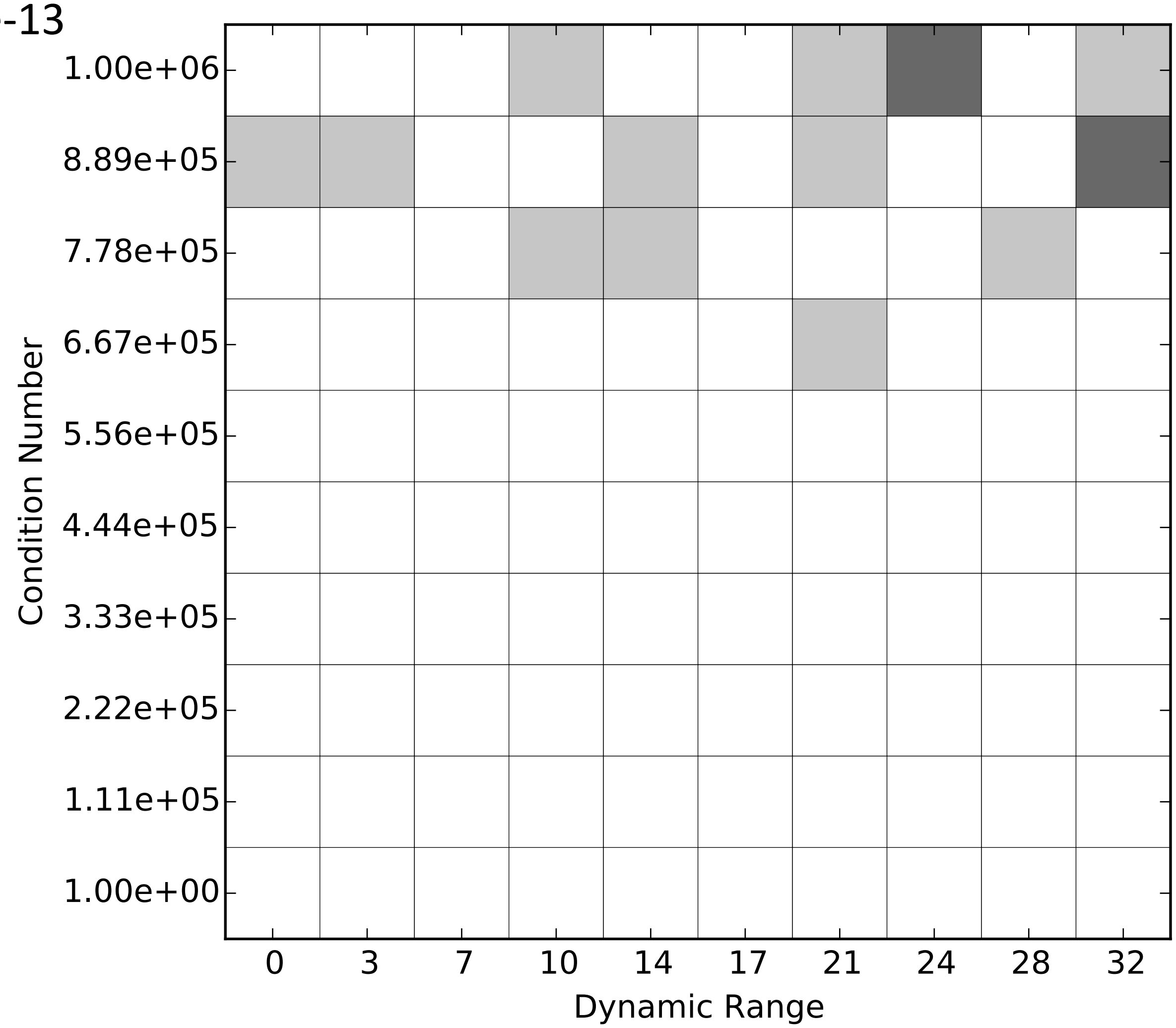


UNIVERSITY *of* DELAWARE

Selecting a Sufficient Algorithm

Variability threshold = 4.5e-13

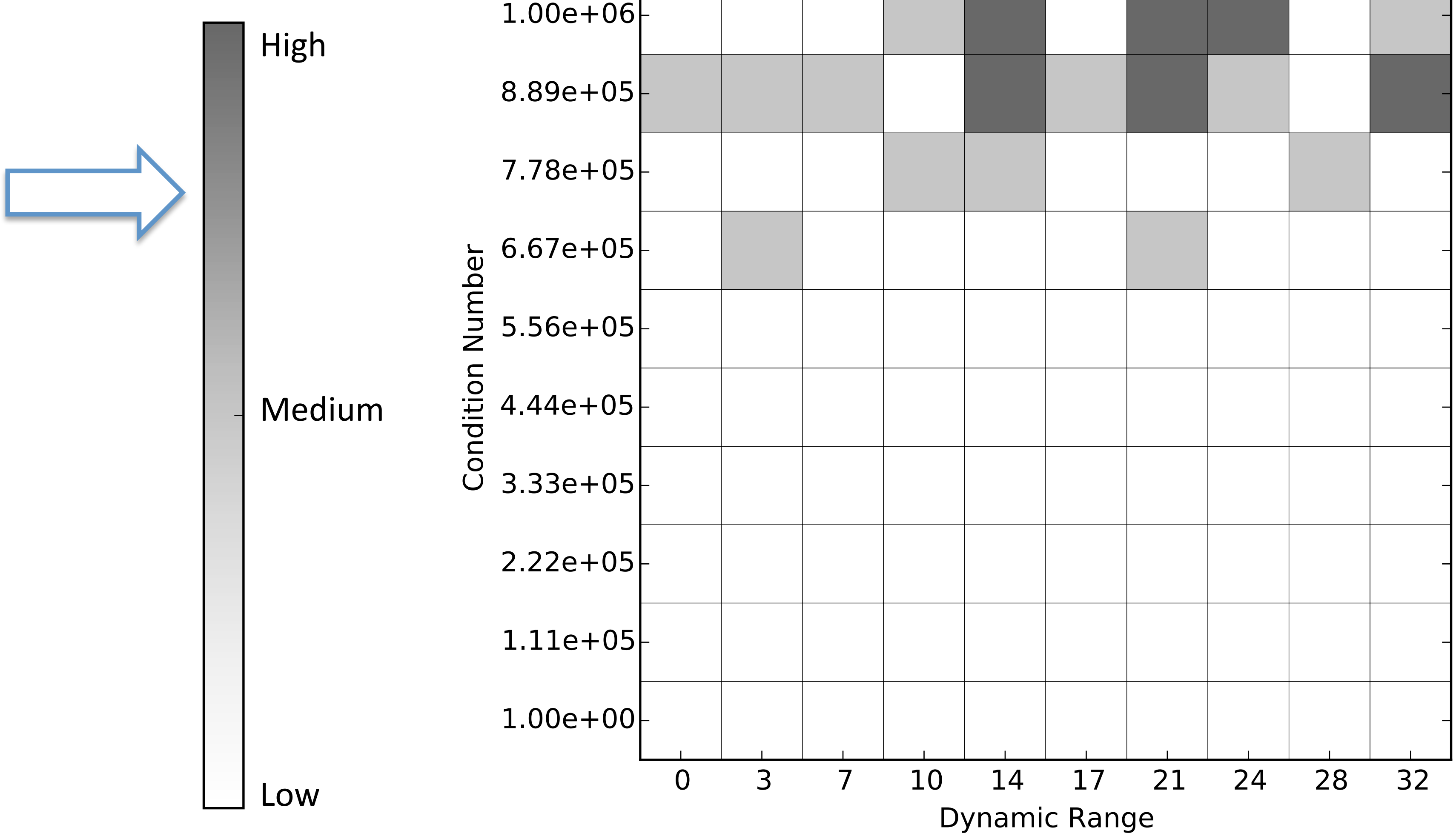




Selecting a Sufficient Algorithm

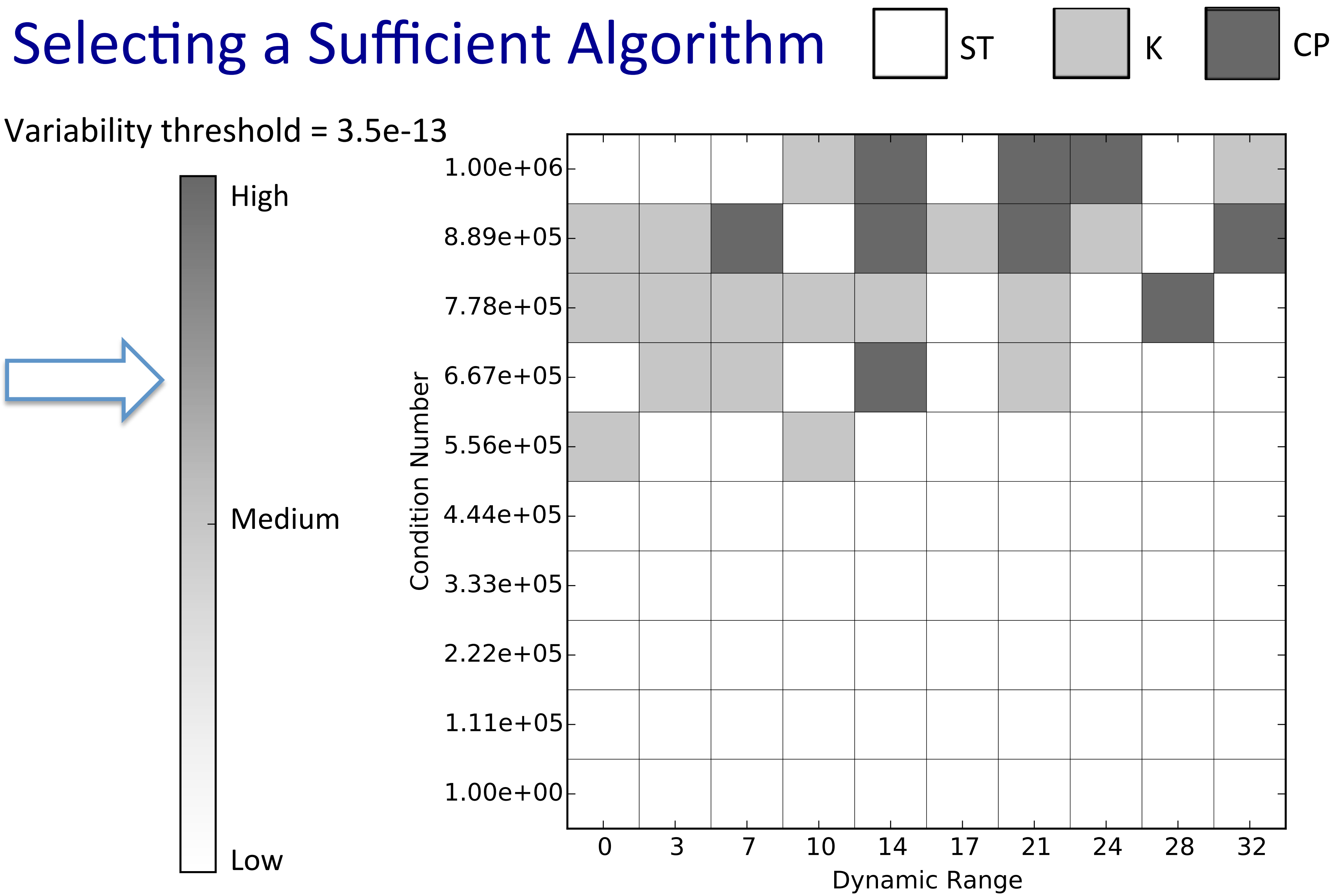
ST K CP

Variability threshold = 4e-13

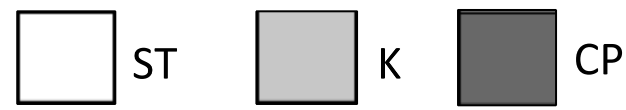


Selecting a Sufficient Algorithm

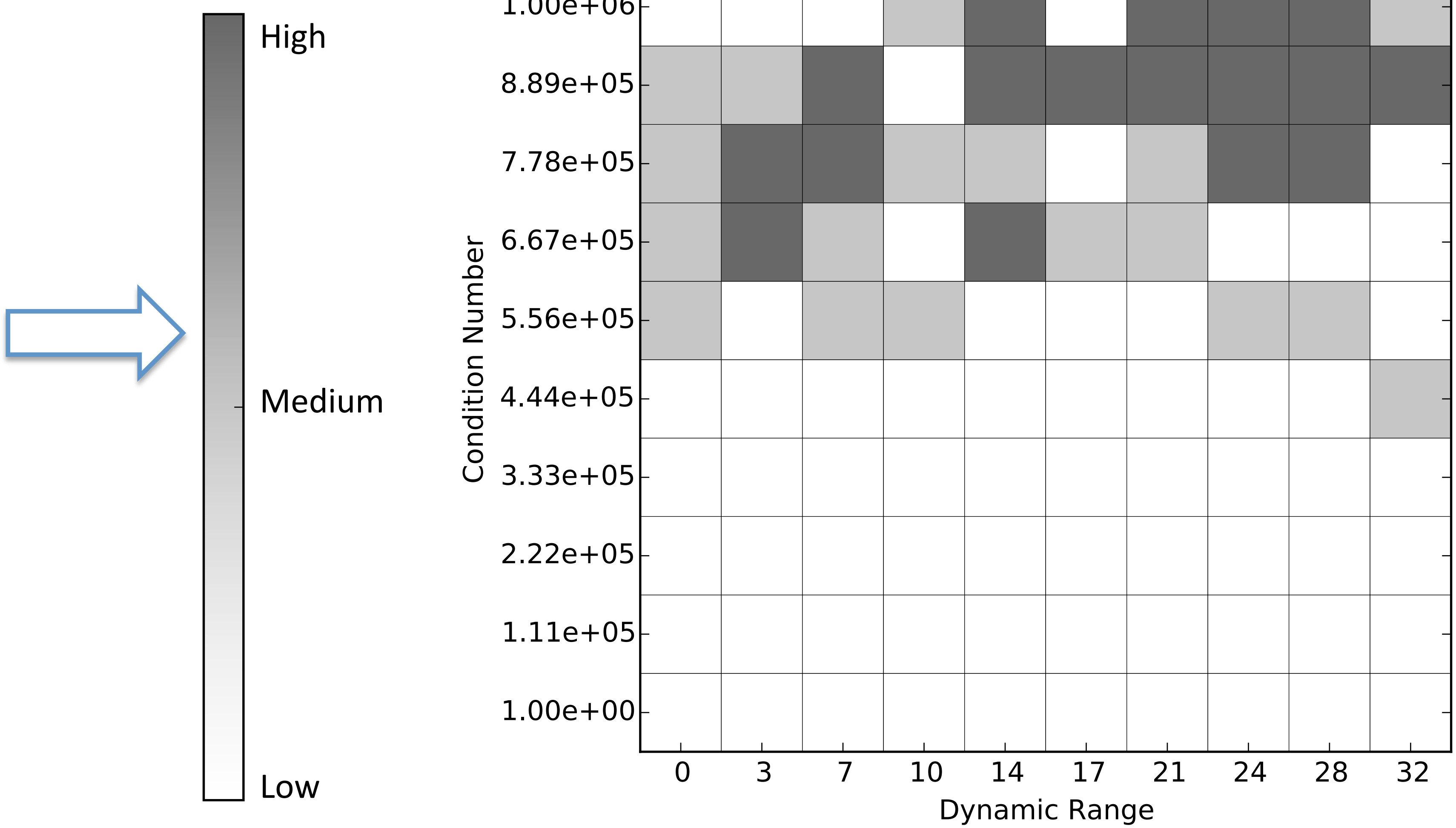
Variability threshold = 3.5e-13



Selecting a Sufficient Algorithm

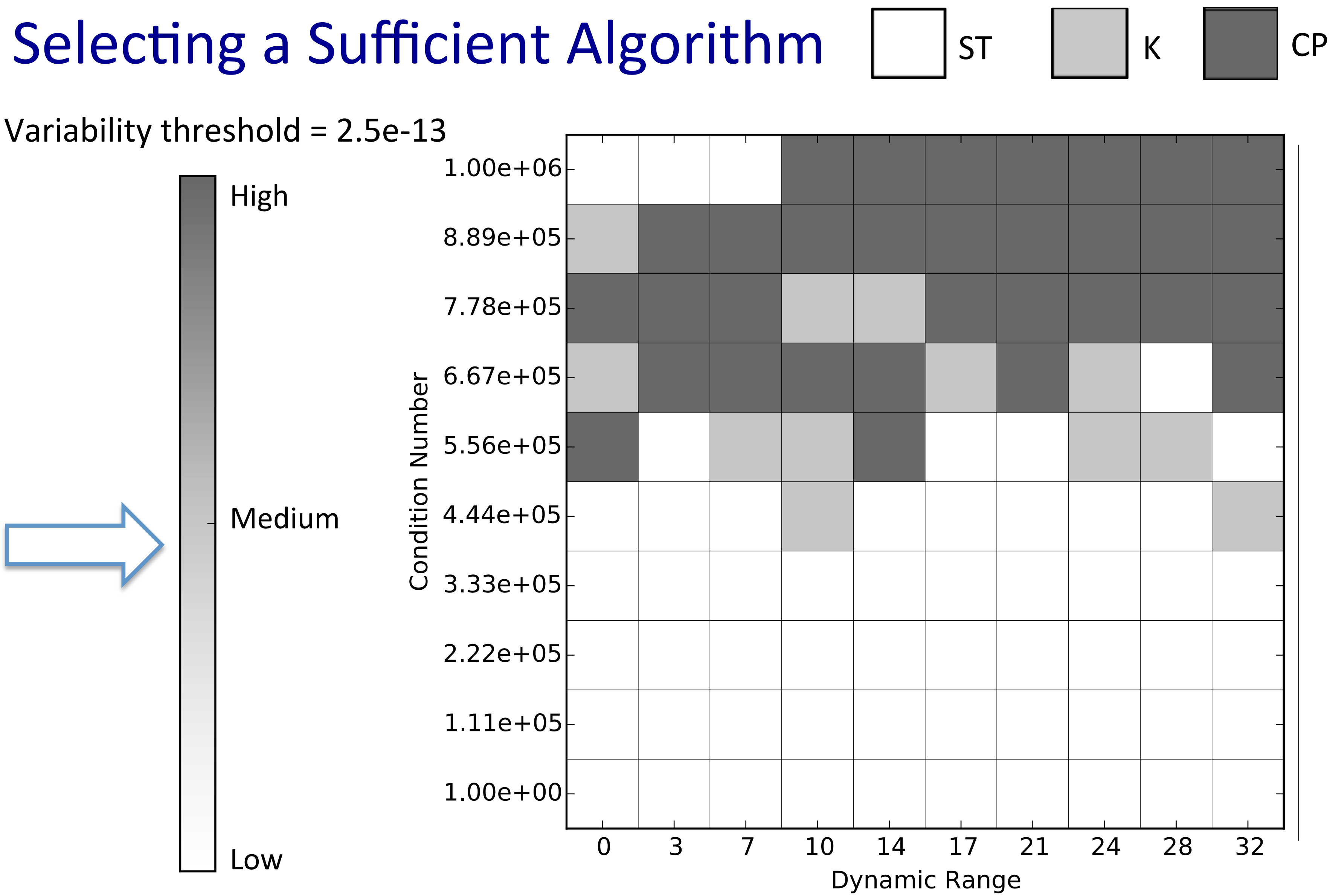


Variability threshold = 3e-13

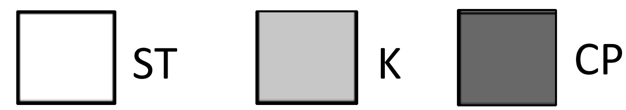


Selecting a Sufficient Algorithm

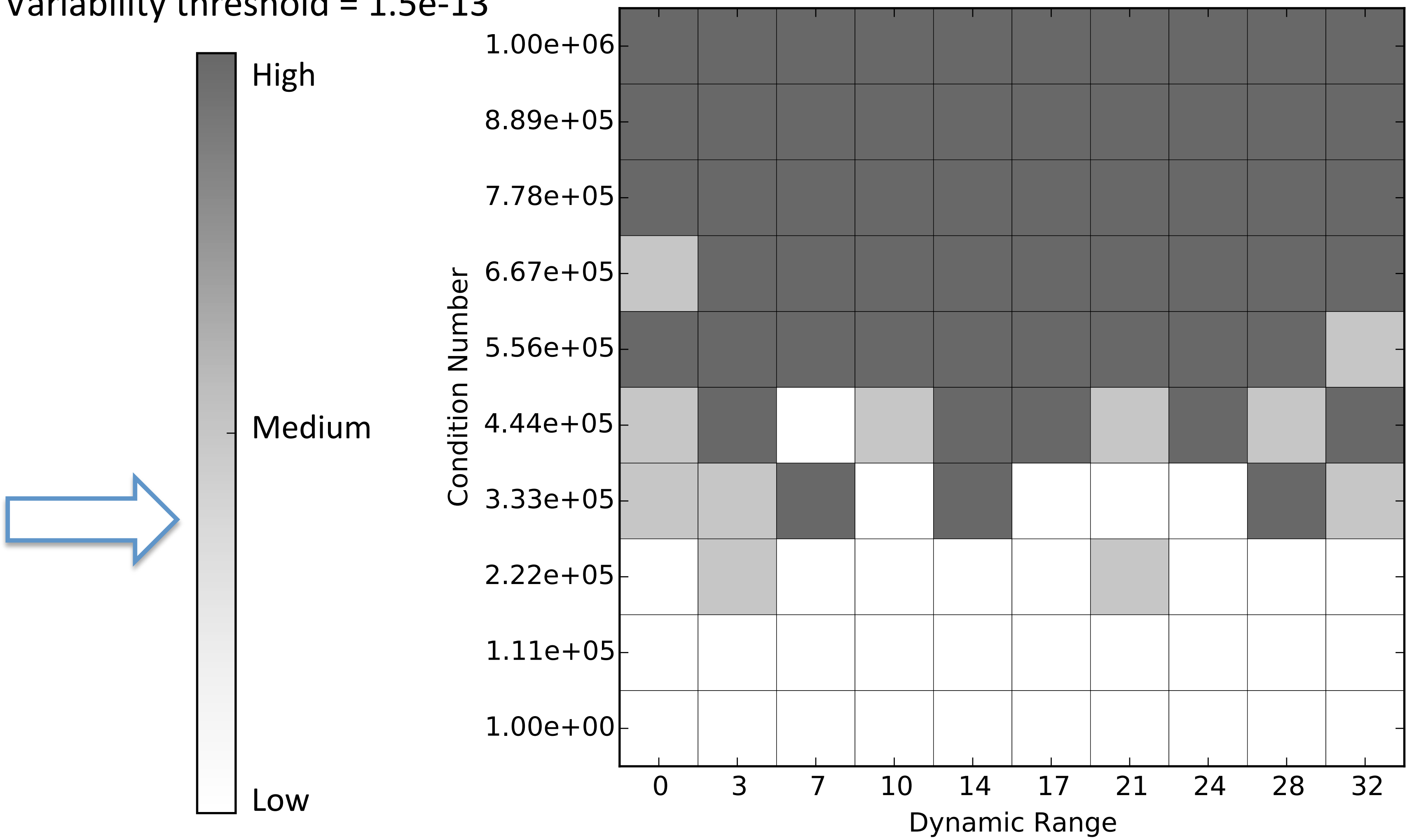
Variability threshold = 2.5e-13



Selecting a Sufficient Algorithm

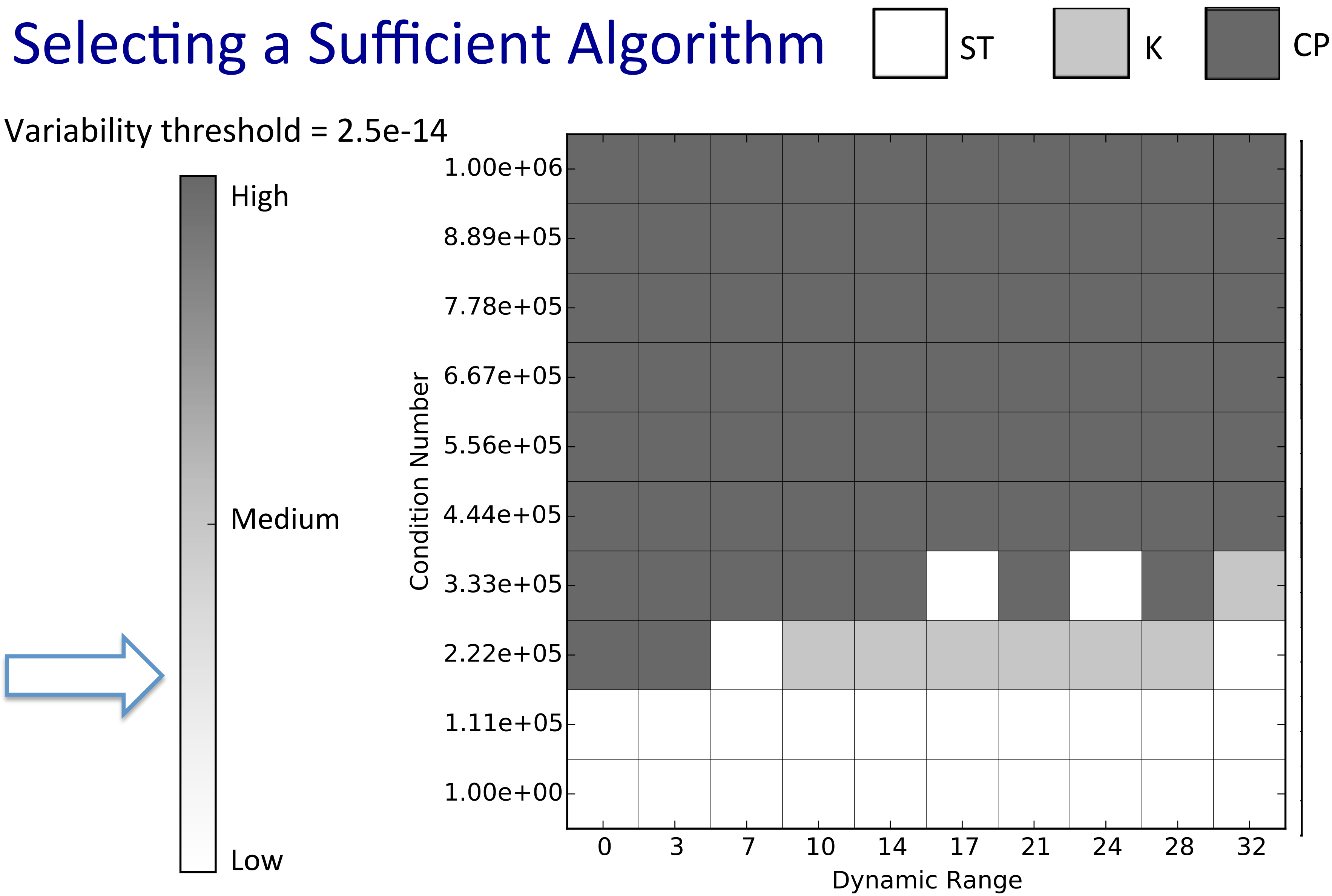


Variability threshold = 1.5e-13



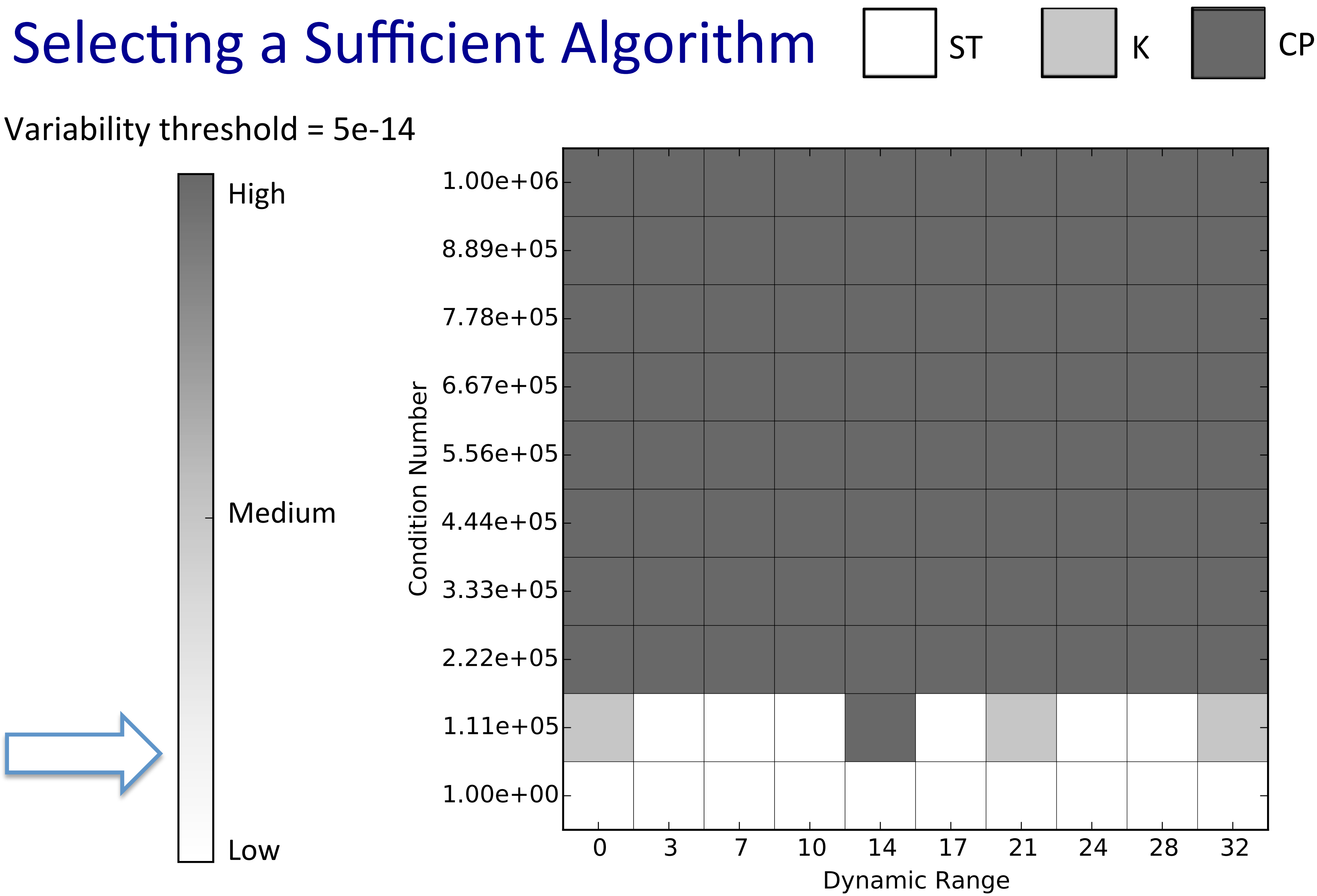
Selecting a Sufficient Algorithm

Variability threshold = 2.5e-14



Selecting a Sufficient Algorithm

Variability threshold = 5e-14



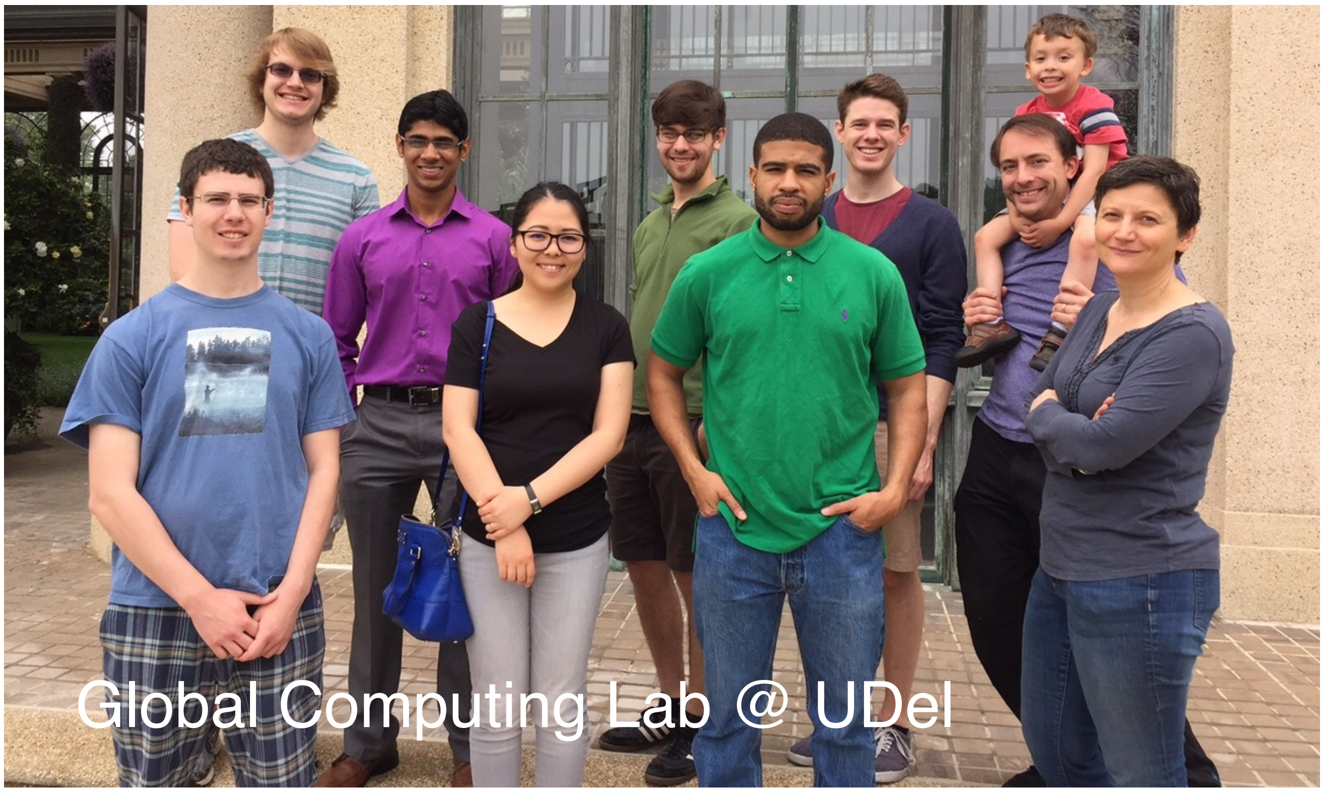
Lesson Learned

- We study an emulated scenario of global summation on exascale platforms
- Increasingly costly summation algorithms needed for reproducible accuracy in certain regions of parameter space
  - High concurrency, ill-conditioned, high dynamic range
- Exascale applications need to maintain awareness of mathematical properties of summands
  - Adjust summation algorithms used to keep variability below threshold

Future Directions

- ***Can we achieve reproducible numerical accuracy by intelligent runtime selection of reduction algorithms?***

Acknowledgments



Sponsors:

Contact:
taufer@udel.edu
gcl.cis.udel.edu