A Statistical Analysis of Error in MPI Reduction Operations

Samuel D. Pollard and Boyana Norris

UNIVERSITY OF OREGON
Computer and Information Science

11 November, 2020

Floating-Point Arithmetic Is Not Associative

HPCL

- Let $\oplus$ be floating-point addition
- $0.1 \oplus (0.2 \oplus 0.3) =$ `0x1.3333333333334p-1`
- $(0.1 \oplus 0.2) \oplus 0.3 =$ `0x1.3333333333333p-1`
- Worse error when the magnitudes are different

```
a <- 1.0
b <- 1e16
c <- -1e16
(a + b) + c = 0
a + (b + c) = 1
```

What is the effect of assuming associativity for parallel summation error?

Bound on Relative Error

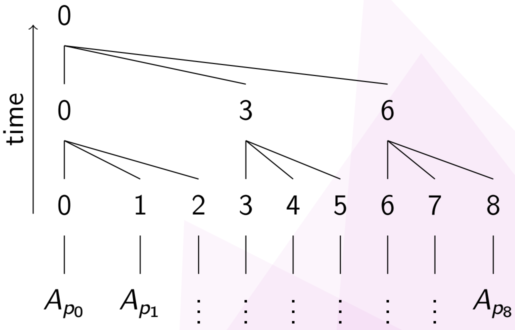
- Let op $\in \{+, -, \div, \times\}$, and $\odot$ be its corresponding floating point operation. Then

$$x \text{ op } y = (x \odot y)(1 + \delta) \text{ where } |\delta| \leq \epsilon. \tag{1}$$

- This holds only for $x \odot y \neq 0$ and normal (not subnormal)
- For double-precision $\epsilon = 2^{-53}$

MPI Reduce

- Assume an array A of size n
- Reduce A to a single value with a binary operation
  - Interesting ones are `MPI_SUM` and `MPI_PROD`
- Distribute A across MPI ranks (each p_k)
- Unspecified but typically deterministic reduction order when run on the same architecture and topology



How many ways are there to do this reduce?

How many ways are there to do this reduce?

- Depends on how we define acceptable reduction strategy
- We list four families
  1. Canonical Left-Associative (Canon)
  2. Fixed Order, Random Association (FORA)
  3. Random Order, Random Association (RORA)
  4. Random Order, Left-Associative (ROLA)

1. Canonical Left-Associative

- Left-associative
- Unambiguous: one reduction strategy
- No freedom to exploit parallelism

```
double acc = 0.0;
for (i = 0; i < N; i++) {
    acc += A[i];
}
```

Parallel Reductions

To look at parallelism, we start with the MPI Standard

The MPI Standard is Flexible

The operation op is always assumed to be associative. All predefined operations are also assumed to be commutative... However, the implementation can take advantage of associativity, or associativity and commutativity, in order to change the order of evaluation. This may change the result of the reduction for operations that are not strictly associative and commutative, such as floating-point addition. [4]

If Commutativity Is Required

The order of operands is fixed and is defined to be in ascending, process rank order, beginning with process zero. The order of evaluation can be changed, taking advantage of the associativity of the operation. [4]

2. Fixed Order, Random Association (FORA)

HPCL

- Let's start with the `commute = false` case
- Assume inorder tree traversal
- Combinatorially well-known example of a Catalan number
- Given array of size n,

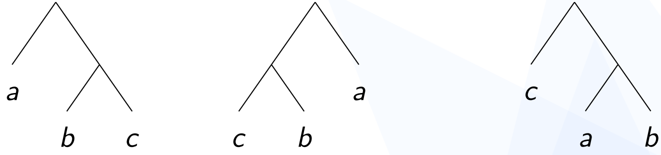
$$C_n = \frac{(2n)!}{(n+1)!n!}$$

  different combinations.
- We call these *associations*

Example Summation

- With the commutative but nonassociative operator $\oplus$, $r_1 = r_2$ but $r_2 \neq r_3$.

$r_1 = a \oplus (b \oplus c)$ $\quad$ $r_2 = (c \oplus b) \oplus a$ $\quad$ $r_3 = c \oplus (b \oplus a)$



3. Random Order, Random Association (RORA)

- This family describes the default if we call MPI_Reduce
- Greater than C_n[1]
- Less well-known, but still solved combinatorial problem [3]

$$g_n = (2n-3)!!$$

where !! is the double factorial (in this case on odd integers). That is, $(2n-3)!! = 1 \times 3 \times 5 \times \cdots \times 2n-3$.

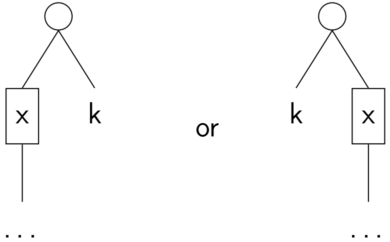
[1] Correction

4. Random Order, Left Associative (ROLA)

- Shuffle array first, then sum canonically
- Main purpose is to compare with previous work by Chapp et al. [2]

Generating Random Binary Trees

- Rémy's Procedure
- Given a tree with $n-1$ leaf nodes, pick one of the nodes randomly (x)
- Add a new node k one of two ways:



Absolute Error

Let $\sum^{\oplus}$ be floating point sum, S_A be the true sum.
Wilkinson back in '63 proved summation error is bounded by

$$\left| \sum_{k=1}^{\oplus\ n} A_k - S_A \right| \leq \epsilon(n-1) \sum_{k=1}^{n} |A_k| + O(\epsilon^2). \tag{2}$$

Estimating Error

From Robertazzi & Schwartz [5] if we assume

1. $A_k \sim U(0, 2\mu)$ or $\exp(1/\mu)$.
2. Floating point errors are independent, distributed with mean 0, variance σ^2
3. Summation ordering is random

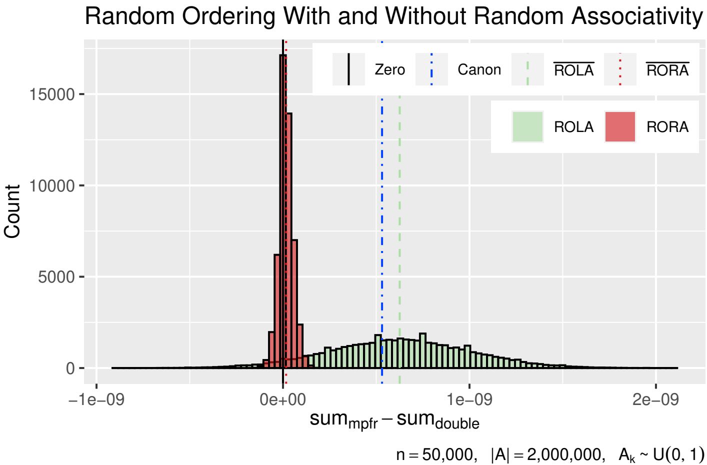
Then the relative error is approximately

$$\frac{1}{3}\mu^2 n^3 \sigma_e^2. \quad (3)$$

We'll substitute values in for (2) and (3) later

Left and Random Associativity (ROLA vs. RORA)

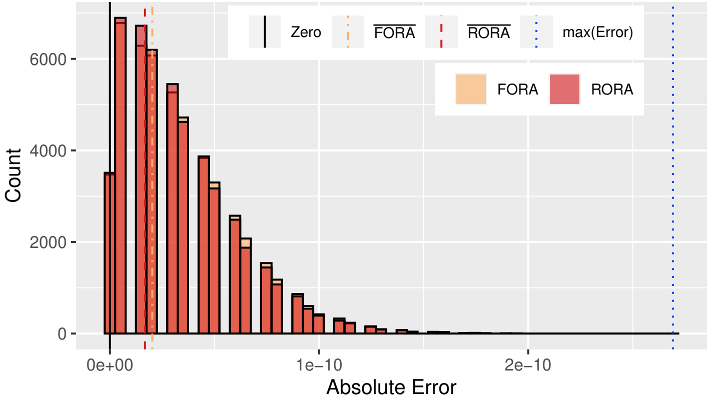
- ROLA is a biased sum
- worst RORA has smaller error than canonical



Fixed and Random Ordering (FORA vs. RORA)

- Almost identical
- Error mainly from adding small number to large partial-sum
- Notice canonical would be off this chart

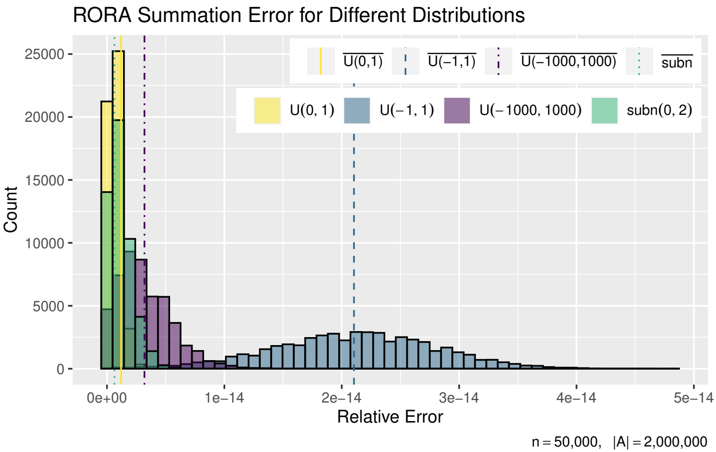
Random Associativity With and Without Random Ordering



$n = 50{,}000, \quad |A| = 2{,}000{,}000, \quad A_k \sim U(0,1)$

RORA with Different Distributions

- U$(-1, 1)$ worst because of catastrophic cancelation



Empirical Results for Uniform(0,1), Tabulated

- $\overline{}$ bar is average
- Robertazzi Estimator matches closely with observed data

Distribution	Measurement	Relative Error
$U(0,1)$	$\overline{\text{RORA}}$	6.702×10^{-16}
$U(0,1)$	max(RORA)	4.073×10^{-15}
$U(0,1)$	$\overline{\text{ROLA}}$	1.282×10^{-14}
$U(0,1)$	Canonical	1.062×10^{-14}
$U(0,1)$	Analytical	1.776×10^{-8}
$U(0,1)$	Robertazzi	6.848×10^{-16}
	machine ϵ	1.110×10^{-16}

Error Estimators for Uniform (-1,1)

HPCL

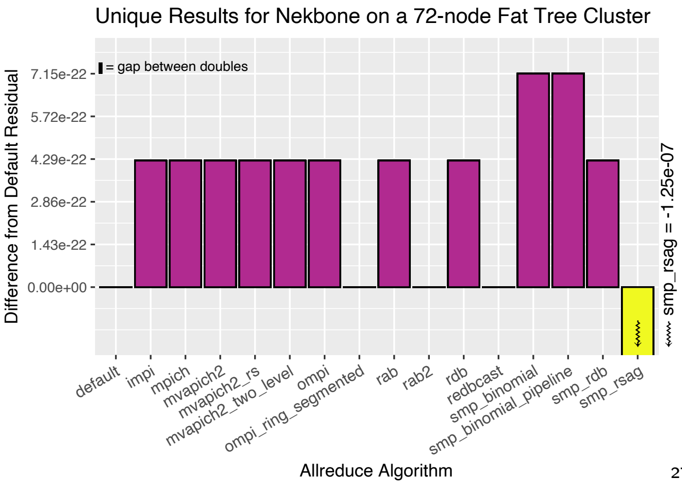
- Recap of previous figures;
- Error is greater for $U(-1,1)$

Distribution	Measurement	Relative Error
$U(-1,1)$	$\overline{\text{RORA}}$	2.104×10^{-14}
$U(-1,1)$	max(RORA)	4.824×10^{-14}
$U(-1,1)$	$\overline{\text{ROLA}}$	8.358×10^{-12}
$U(-1,1)$	Canonical	6.124×10^{-12}
$U(-1,1)$	Analytical	7.951×10^{-7}
	machine ϵ	1.110×10^{-16}

HPCL

Nekbone

- Nekbone is a computational fluid dynamics proxy app
- We look at residual of conjugate gradient
- We use SimGrid [1] to try out 16 different allreduce algorithms



Nekbone (Cont.)

- Only four results across 16 algorithms
- Most differ only by the last few bits

Allreduce Algo. Rank	Residual
Best (smp_rsag)	$1.616\,306\,278\,792\,575 \times 10^{-8}$
Default	$14.082\,603\,491\,982\,\boxed{575} \times 10^{-8}$
Worst	$14.082\,603\,491\,982\,\boxed{647} \times 10^{-8}$
Other	$14.082\,603\,491\,982\,\boxed{618} \times 10^{-8}$

Future Work & Conclusion

Future Work

- Generate more realistic reduction trees, realistic random input, expose more nondeterminism in SimGrid

In Conclusion

- Looked at error for four different families of reduction strategies
- Reduction tree shape has greater effect than how the array is ordered
- Despite large state space, realistic programs generate a tiny subset of what is permitted

Source and slides at
`github.com/sampollard/reduce-error`

Thank you!

References I

[1] Casanova, H., Giersch, A., Legrand, A., Quinson, M., and Suter, F.
Versatile, scalable, and accurate simulation of distributed applications and platforms.
Journal of Parallel and Distributed Computing 74, 10 (June 2014), 2899–2917.

[2] Chapp, D., Johnston, T., and Taufer, M.
On the need for reproducible numerical accuracy through intelligent runtime selection of reduction algorithms at the extreme scale.
In *IEEE International Conference on Cluster Computing* (Chicago, IL, USA, Sept. 2015), IEEE, pp. 166–175.

[3] Dale, M., and Moon, J.
The permuted analogues of three catalan sets.
Journal of statistical planning and inference 34, 1 (Jan. 1993), 75–87.

[4] Message Passing Interface Forum.
MPI: A message-passing interface standard: Version 3.1.
Tech. rep., MPI Forum, Knoxville, TN, United States, 2015.

References II

[5] Robertazzi, T. G., and Schwartz, S. C.
Best "ordering" for floating-point addition.
Transactions on Mathematical Software 14, 1 (Mar. 1988), 101–110.

UNIVERSITY OF OREGON