SimTech Cluster of Excellence

Universität Stuttgart Germany

www.simtech.uni-stuttgart.de

Numerical Error Analysis for Statistical Software on Multi-Core Systems

Wenbin Li, Sven Simon
liwn@ipvs.uni-stuttgart.de
Institute of Parallel and Distributed Systems
University of Stuttgart, Germany

Outline

1. Numerical Errors and Accuracy Control
2. A Method for Accuracy Control
   - e.g. Result : $9.676383250285714 * 10^3$
     - Accurate 12 digits: 9.67638325028
     - Contaminated digits due to rounding errors: 5714

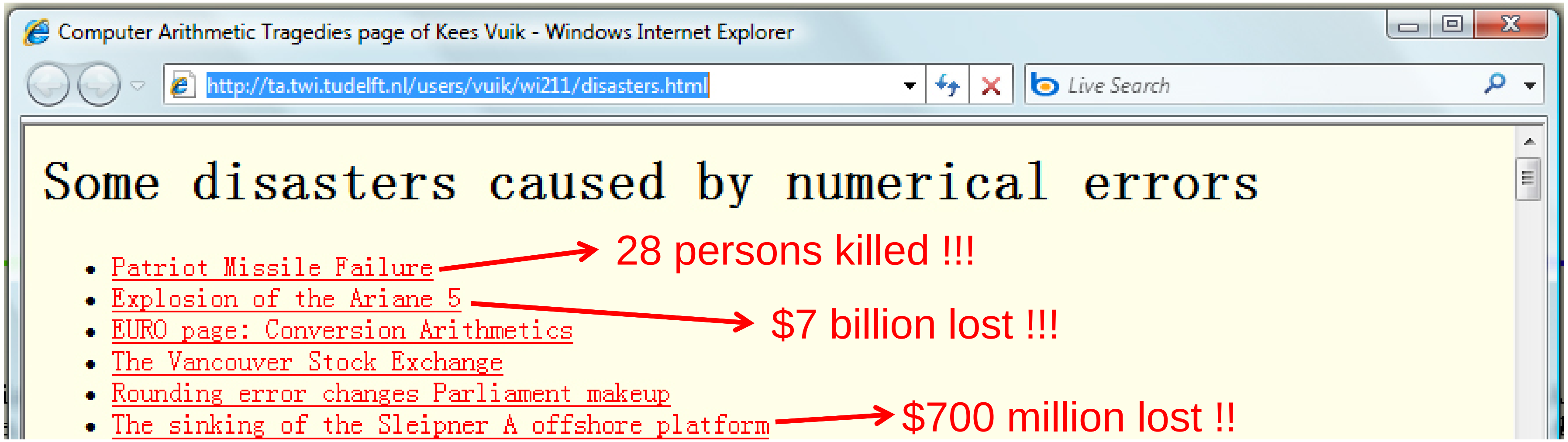
     Accuracy: 12 digits
3. Applied to a Statistical Software – R
4. Parallelization on Multi-Core Systems
5. Conclusion

Disasters Caused by Numerical Errors

"Numerical precision is the very soul of science."

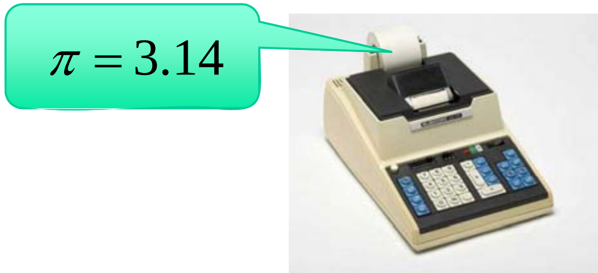
-- D'Arcy Wentworth

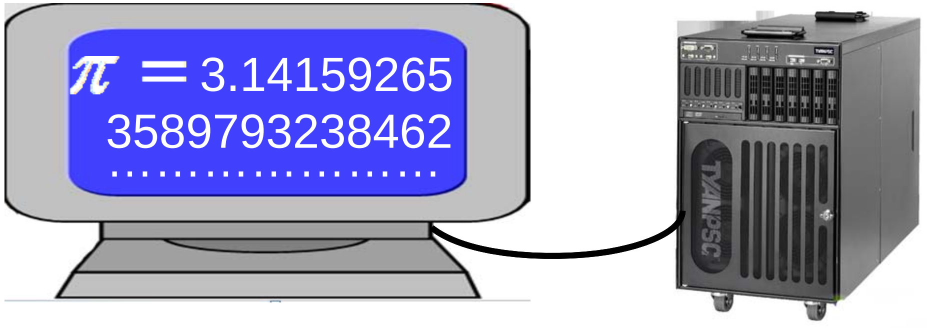
http://ta.twi.tudelft.nl/users/vuik/wi211/disasters.html



Schemes to Evaluate Numerical Quality

- Compare the result with a high precision reference
  - Reference is obtained by repeating the computation in arithmetic of increasing precision.



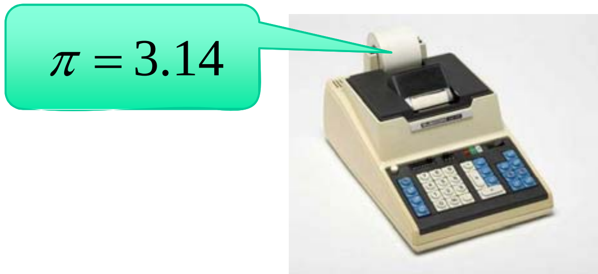


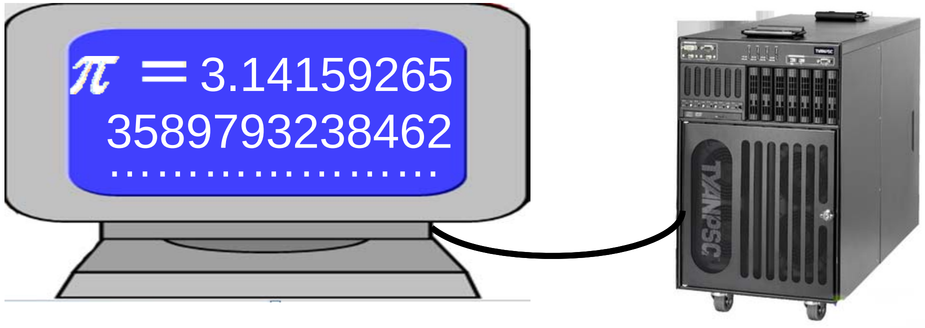
Schemes to Evaluate Numerical Quality

- Compare the result with a high precision reference
  - How many digits are sufficient?
  - Low performance:
    - Single precision performance >> Double precision (e.g. GPU, CellBE, etc.).
    - Arbitrary high precision is several orders of magnitude times slower.
  - Hard to modify source codes to generate high precision reference:
    - millions of lines of source code.
    - Unknown libraries → source code available (readable) ?
    - Mismatch in memory:
      - Fortran: common
      - C/C++ : union

Schemes to Evaluate Numerical Quality

- Compare the result with a high precision reference
  - Reference is obtained by repeating the computation in arithmetic of increasing precision.





- Interval arithmetic

  e.g. Your result is in $(-\infty, +\infty)$ guaranteed

- Probabilistic error analysis

Probability Distribution of the Rounding Error

- The calculated result in a program

$$\hat{R} = r + \varepsilon_{rounding}$$

r : exact result

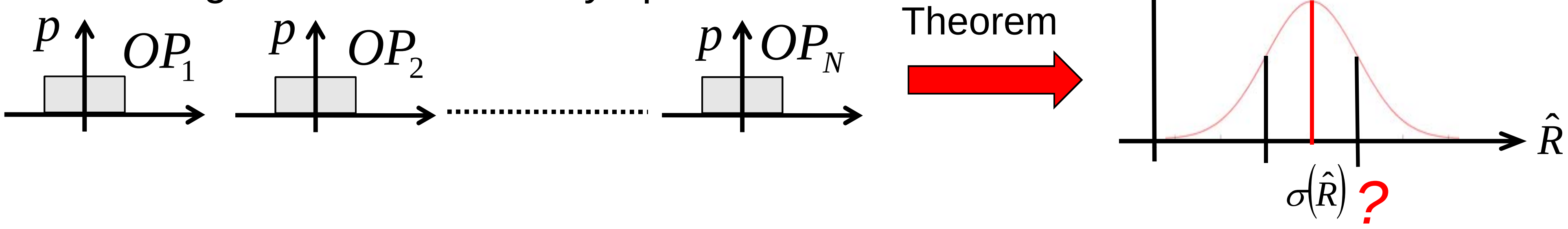
$\hat{R}$: approximation of r due to rounding error.

$\varepsilon_{rounding}$: rounding error.

- First order model ([1]) : $\varepsilon_{rounding} = \sum_{k=1}^{n} g_i(d) \cdot 2^{-p} \alpha_i + \mathrm{O}\left(2^{-2p}\right)$

Assume a probability distribution for rounding error of elementary operation

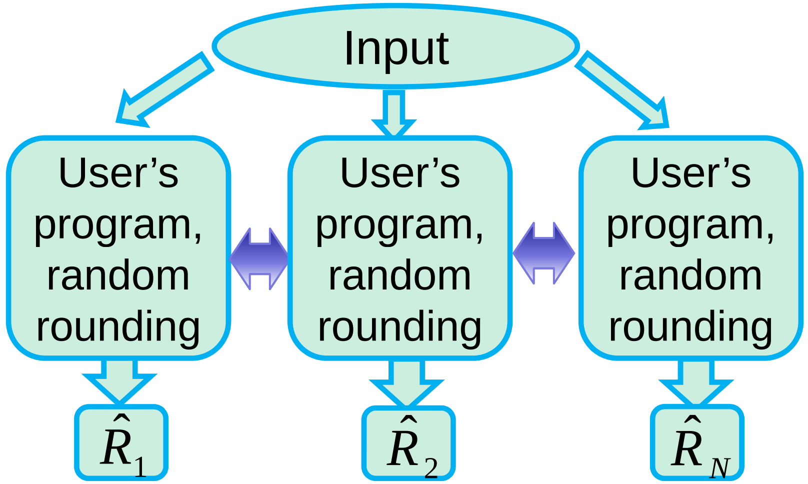
Gaussian Distribution



Probabilistic Rounding Error Analysis (PREA)

- Consider a sequence of computations providing an exact result r .
- Perform computations *N* times with random rounding (randomly choosing rounding to $+\infty$ or $-\infty$), *N* results $\hat{R}_i, i = 1,\ldots N$ are obtained.
- The computed result is

$$\overline{R} = \frac{1}{N}\sum_{i=1}^{N}\hat{R}_i$$



- Instead of $\sigma\left(\overline{R}\right)$, we have sample variance:

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(\hat{R}_i - \overline{R}\right)^2$$

- Define

$$T = \frac{\overline{R} - r}{s/\sqrt{N}}$$ ➡ sample from Student's t-distribution

- Probability density function

$$f\left(T = t\right) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\cdot\Gamma\left(\nu/2\right)}\left(1+\frac{t^2}{\nu}\right)^{-\left(\frac{\nu+1}{2}\right)}$$

Number of Significant Digits

- Confidence Interval

$$\Pr\left(-\tau_\beta < \frac{\overline{R}-r}{s/\sqrt{N}} < \tau_\beta\right) = \beta$$

- With probability β , the number of the significant digits (i.e. accurate digits) of $\overline{R}$ is

$$N_{\text{Significan t Digit of } \overline{R}} \geq \log_{10}\left(\frac{\sqrt{N}\cdot\left|\overline{R}\right|}{\tau_\beta \cdot s}\right)$$

- Two Hypothesis should hold ([1, 2]) :
  - Random rounding error is Gaussian distributed.

    -- not importance because of robustness of Student's t-test.
  - The first order approximation is legitimate.
    - The operands of multiplication must be significant.
    - The divisor of division must be significant.

} Self validation

Applied to Complex Statistical Software

R (v2.10.1): a software for statistical computing and graphics.

Benchmarks: NIST StRD (a collection of data sets and certified values)

Accuracy Estimation using PREA :

$$\log_{10}\left(\frac{\sqrt{N}\cdot\left|\overline{R}\right|}{\tau_{\beta}\cdot s}\right)$$

Reference (true accuracy):

$$LRE = -\log_{10}\left(\frac{\overline{R}-r}{r}\right)$$

Certified values from StRD

		PREA	True	PREA	True	PREA	True	PREA	True
UNIV	benchmark	Mean		Standard Deviation		Autocorrelation		-	
mean sd acf	Mavro	15	15	13	13	13	14	-	-
	Numacc3	15	15	10	10	11	10	-	-

Applied to Complex Statistical Software

R (v2.10.1): a software for statistical computing and graphics.

Benchmarks: NIST StRD (a collection of data sets and certified values)

Accuracy Estimation using PREA :
$$\log_{10}\left(\frac{\sqrt{N}\cdot\left|\overline{R}\right|}{\tau_{\beta}\cdot s}\right)$$

Reference (true accuracy):
$$LRE = -\log_{10}\left(\frac{\overline{R}-r}{r}\right)$$

Certified values from StRD

		PREA	True	PREA	True	PREA	True	PREA	True
UNIV	benchmark	Mean		Standard Deviation		Autocorrelation		-	
mean sd acf	Mavro	15	15	13	13	13	14	-	-
	Numacc3	15	15	10	10	11	10	-	-

Number of significant Digits

Applied to Complex Statistical Software

		PREA	True	PREA	True	PREA	True	PREA	True
ANOVA	benchmark	SST		SSE		MSE		F-statistic	
	SiRstv	12	12	12	13	12	13	12	12
aov	SmLs08	4	4	2	2	2	2	2	2
LINR		Coefficient		RSD		R^2		F-statistic	
	Norris	12	13	14	14	15	15	13	14
lm	Wampler5	5	6	15	15	14	14	14	15
NLINR		$Coefficient_i$		$Coefficient_j$		RSS		RSD	
	Lanczos2	5	5	6	6	7	8	8	8
Nls	Bennet5	4	4	4	5	9	9	9	9

	Exact estimation	**Underestimation by 1 digit**	**Overestimation by 1 digit**
Relative Frequency	67%	29%	4%

If underestimation by 1 digit is tolerable:

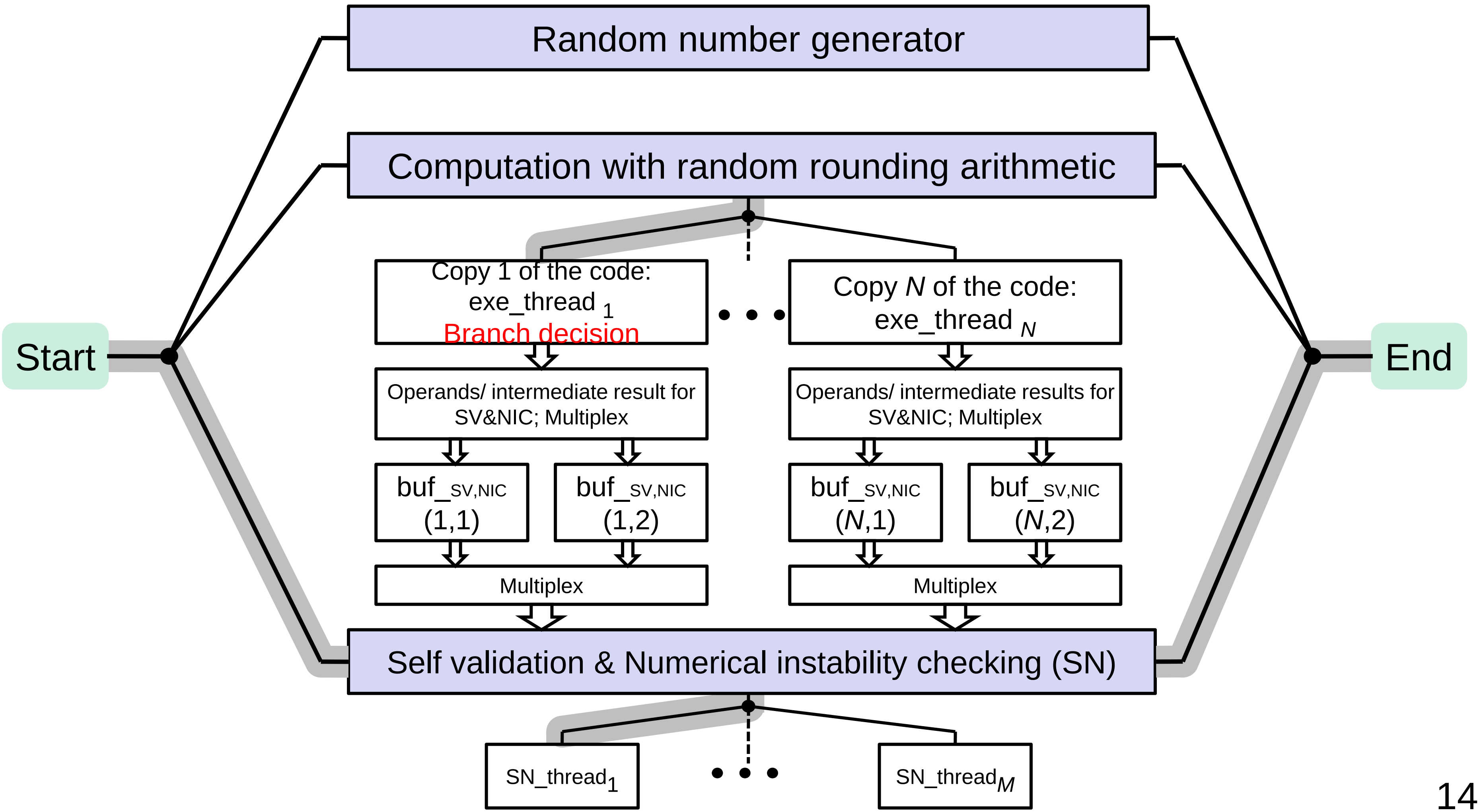
➡ **reliable estimation in 96% of the cases**

Parallelization for Acceleration

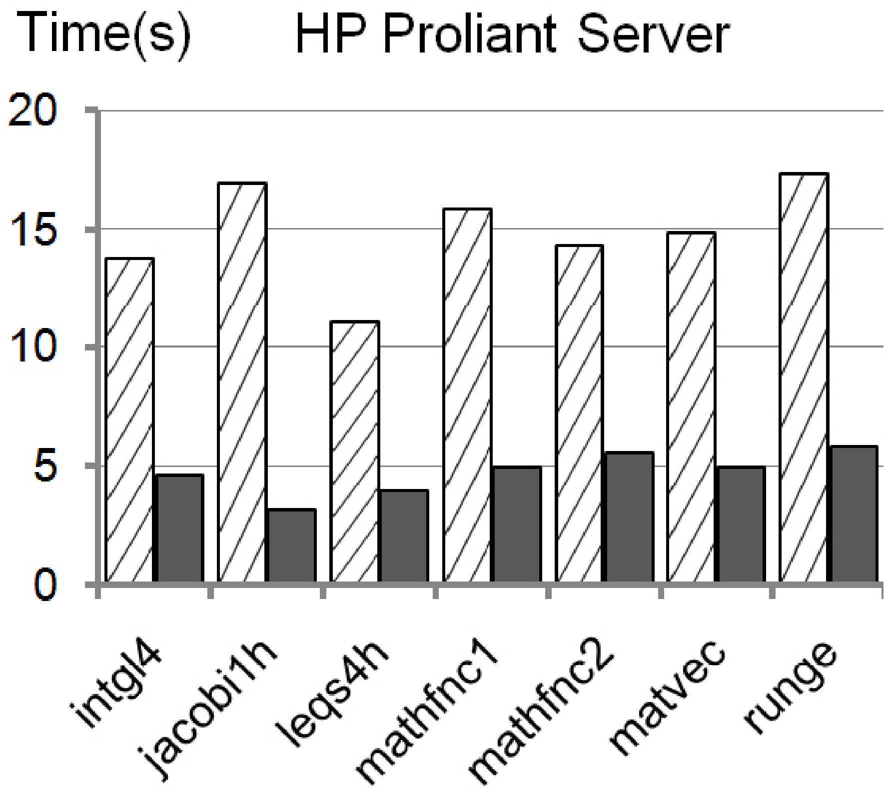
- Why do we need parallel computing?
  - The multiple runs of the code are intensive in computation time.
  - Multi-cores are the mainstream hardware architecture.
  - To harvest the performance potential of multi-core processors.
- Parallel execution
  - To get *N* random rounding result, the user's code can be executed concurrently on different CPU cores.
  - Communication between threads is necessary to:
    - make a unitive decision for instructions such as " IF() THEN".
    - perform Self Validation (SV)
    - perform Numerical Instabilities Checking (NIC).

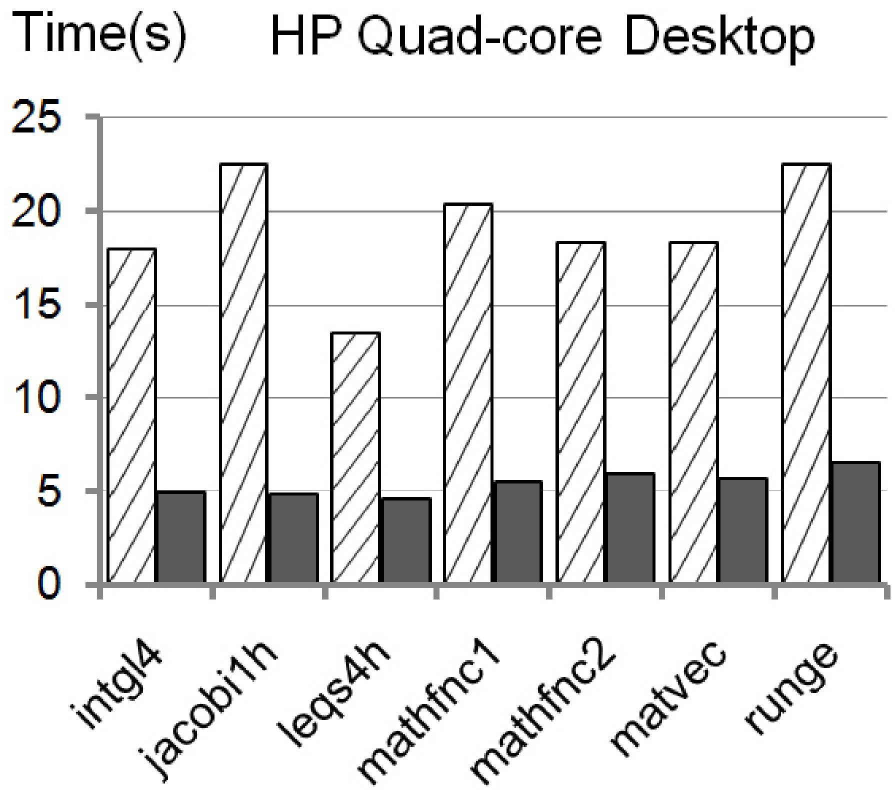
      i.e. sudden accuracy loss due to cancellation in '+' or '-').
  - How to the minimize communication & synchronization overhead?

Parallelization with Asynchronous SV & NIC



Performance of Parallelization with Asynchronous SV & NIC



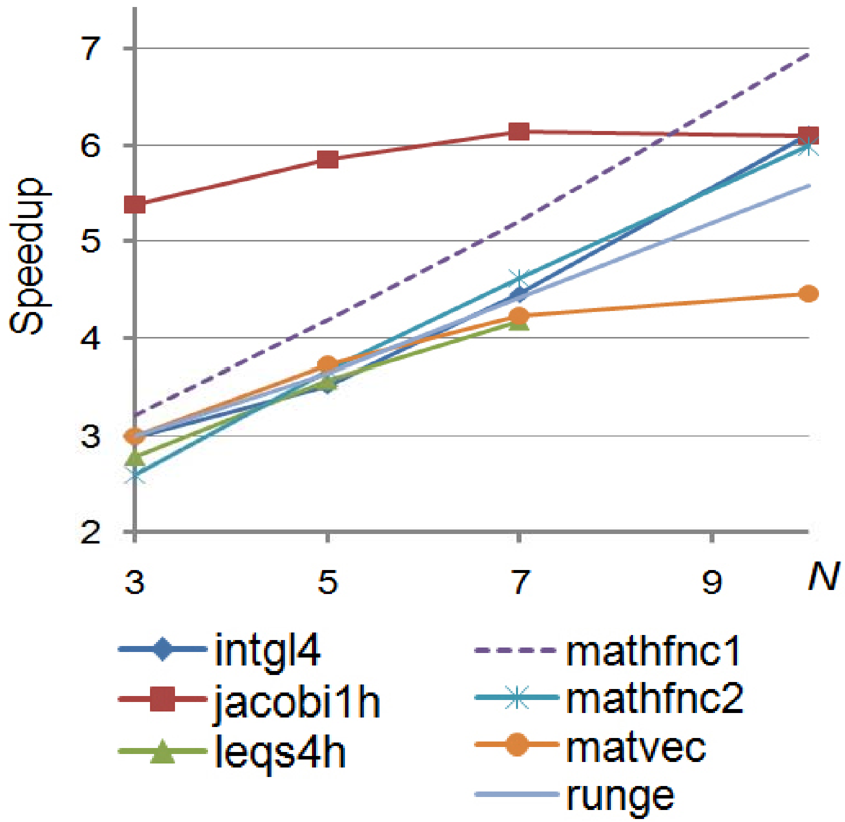


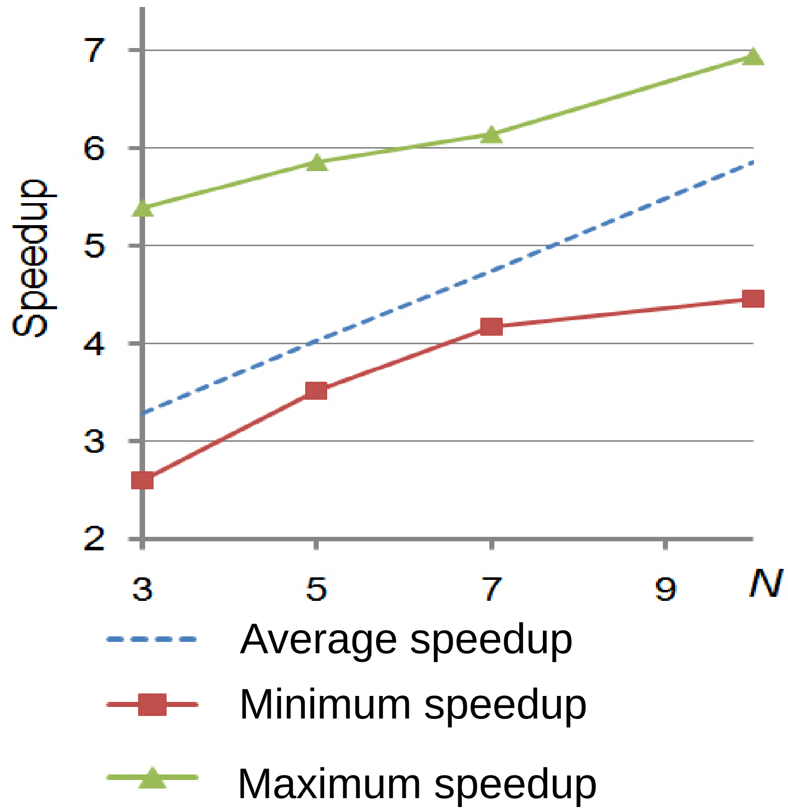
Sequential CADNA

Parallelization with asynchronous SV & NIC

Performance of Parallelization with Asynchronous SV & NIC

This parallelization approach achieves an almost linear scalability.





Conclusion

1. Probabilistic error analysis method is very robust and provide tight rounding error estimation.
2. Numerical instability can be localized.

Line 10034 → $9.6763832502857714 * 10^3$

Line 10035 → $2.7867862139879 28 * 10^3$

Sudden increase in numerical error

3. Parallelization approaches significantly accelerate the proposed method, and shows an almost linear scalability.

Reference

1. J. –M. Chesneaux, J.-M , *Study of the computing accuracy by using probabilistic approach*; Contribution to Computer Arithmetic and Self-Validating Numerical Methods, ed. C. Ulrich (J.C. Baltzer) 1990, 19-30.
2. J. Vignes, *Discrete stochastic arithmetic for validating results of numerical software*. *Numerical Algorithms,* Vol. 37, p. 377-390
3. Douglas C. Montgomery, *Applied Statistics and Probability for Engineers*. page 257.
4. N. Tajima, FORTRAN benchmark tests. *http : //www:apphy:fukui-u.ac.jp/ tajima/bench/index_e.html*

Thanks!

Questions?

Comment?