Exploring Numerical Accuracy and Mixed-Precision with Verificarlo and Stochastic Rounding

Pablo de Oliveira Castro <pablo.oliveira@uvsq.fr>

PASC'25 2025-06-16

LI-PaRAD, UVSQ, Université Paris-Saclay

Outline

Introduction

- Major ecological crisis: French roadmap targets carbon neutrality in 2050 (Stratégie Nationale Bas Carbone).
- Requires a 40% energy consumption reduction.

- HPC part of the solution: modeling and improving complex systems

- HPC part of the problem: Frontier system at ORNL
  - Exaflop performance ($> 10^{18}$ FLOPS)
  - Consumes 21MW: energy of a small town (16 000 french houses)

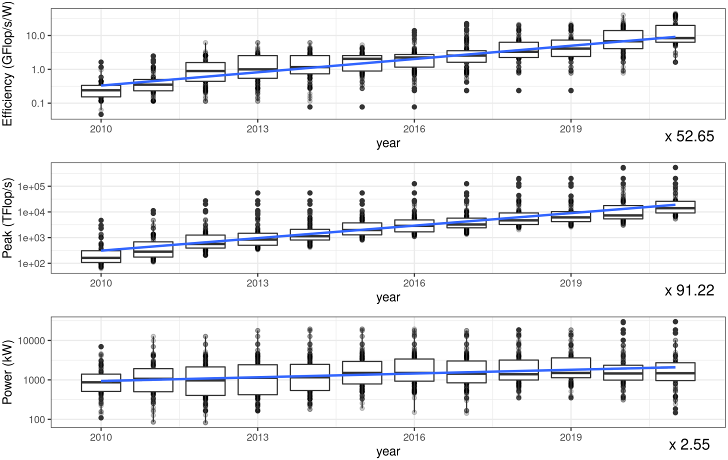
Rebound Effects: Efficiency is Not Enough

- In 1865, Jevons observed that steam engine improvements *increased* coal consumption.

- In HPC, efficiency gains fuel rising computation demand, leading to a net increase in total power consumption.
- Rebound effects for data-centers [Masanet, 2020]
  $\rightarrow$ 6% increase in energy consumption from 2010 to 2018 (255 % increase in nodes).

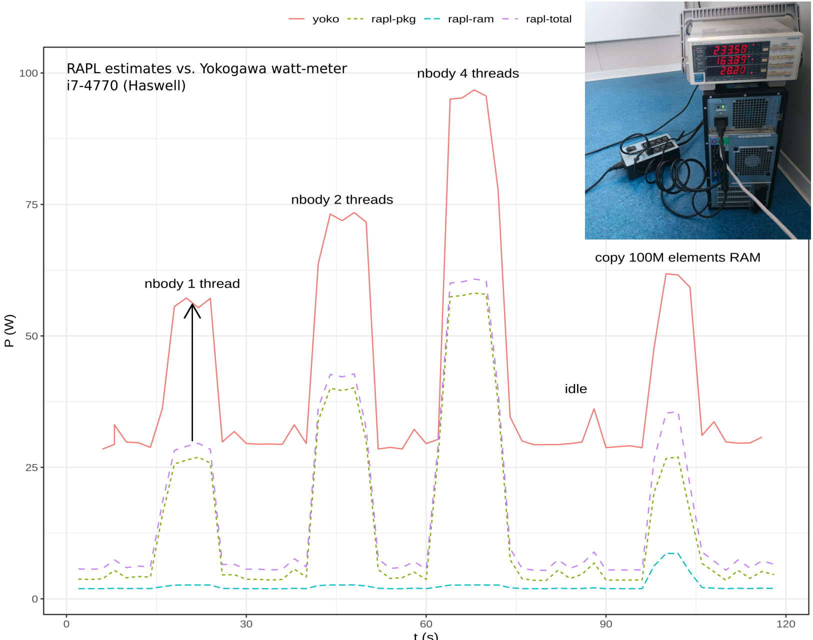
- Indirect rebound effects: advances in computation accelerate energy use in other fields.

Analysis of TOP-100 HPC systems

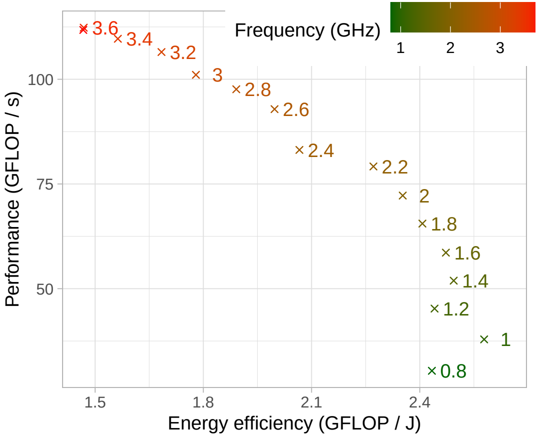


Efficiency (GFlops/W) and Peak performance are increasing. But so is overall power consumption.

RAPL vs. Yokogawa watt-meter



DVFS study of LU decomposition

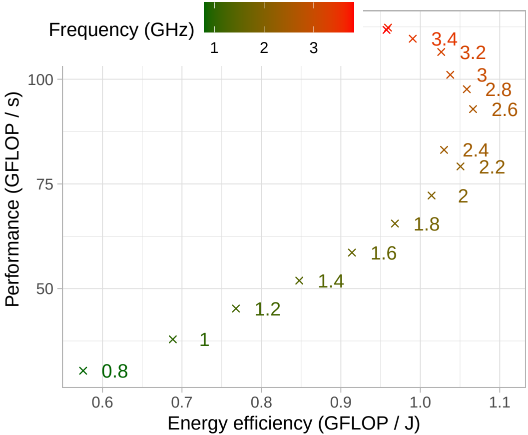


- Knights Mill 72 cores
- Intel MKL dgetrf
- $n \in [1000, 3000]$
- RAPL estimation

Save energy by computing slower: 1GHz

Thomas Roglin, M1 UVSQ/INTEL internship 2023

The "Race to Idle": Accounting for the Whole System



- Model: RAPL + 40W static power

- Optimal 2.6 GHz: compute faster and turn off machine
- Saves idle power (race to idle)

Thomas Roglin, M1 UVSQ/INTEL internship 2023

From Frequency Scaling to Precision Scaling

Frequency scaling: compromise between time-to-solution and energy.

Another powerful lever for speed and energy efficiency is using less data per operation.

- **Example:** FP32 / FP16 / Bfloat16 operations are faster and more energy-efficient than FP64.

Question: How can we reduce precision **safely**, without compromising the numerical integrity?

Outline

The Energy Imperative

Exploring numerical accuracy with Verificarlo

Monte Carlo Arithmetic [Stott Parker, 1999]

- Each FP operation may introduce a δ error

$$\widehat{z} = (x \circ y)(1 + \delta)$$

- Monte Carlo Arithmetic makes δ a random variable

$$\begin{aligned} \widehat{z_1} &= (a + b)(1 + \delta_1) \\ \widehat{z_2} &= (c + d)(1 + \delta_2) \\ \widehat{z} = \widehat{z_3} &= (z_1 \times z_2)(1 + \delta_3) \end{aligned}$$

- The forward error $\Psi = \frac{\hat{z} - z}{z}$ is analyzed probabilistically
  - Stochastic process function of the $\delta_1, \ldots, \delta_k$.

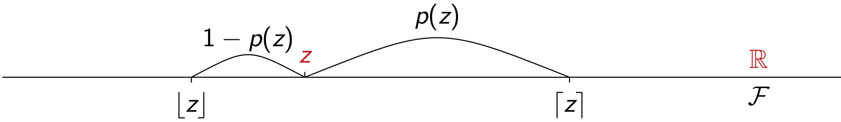
- How to choose the δ_k distribution?

Stochastic Rounding (SR) $\rightarrow$ unbiased

- Upward rounding $\lceil z \rceil$ and downward rounding $\lfloor z \rfloor$:

$$\widehat{z} = z(1+\delta) \text{ with } |\delta| \le u$$

$$\widehat{z} = \begin{cases} \lceil z \rceil & \text{with probability } p(z), \\ \lfloor z \rfloor & \text{with probability } 1 - p(z). \end{cases}$$



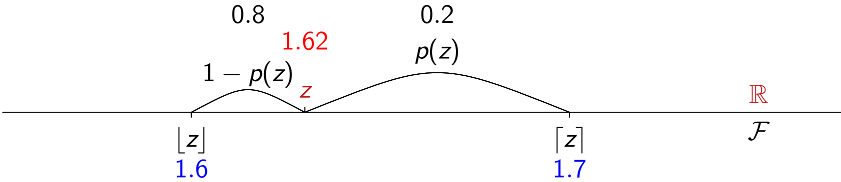
- $p(z) = \frac{z - \lfloor z \rfloor}{\lceil z \rceil - \lfloor z \rfloor}$ and $E(\hat{z}) = p(z)\lceil z \rceil + (1 - p(z))\lfloor z \rfloor = z$.

Stochastic Rounding (SR) $\rightarrow$ unbiased

- Upward rounding $\lceil z \rceil$ and downward rounding $\lfloor z \rfloor$:

$$\widehat{z} = z(1+\delta) \text{ with } |\delta| \leq u$$

$$\widehat{z} = \begin{cases} \lceil z \rceil & \text{with probability } p(z), \\ \lfloor z \rfloor & \text{with probability } 1 - p(z). \end{cases}$$



- $p(z) = \frac{z - \lfloor z \rfloor}{\lceil z \rceil - \lfloor z \rfloor}$ and $E(\hat{z}) = p(z)\lceil z \rceil + (1 - p(z))\lfloor z \rfloor = z$.
- $1.7 \times 0.2 + 1.6 \times 0.8 = 1.62$.

SR errors are mean independent

- In SR, for $x_1, x_2, x_3 \in \mathcal{F}$ and $\circ_1, \circ_2 \in \{+, -, *, /\}$,

$$z = x_1 \circ_1 x_2 \circ_2 x_3 \quad \Longrightarrow \quad \hat{z} = ((x_1 \circ_1 x_2)(1+\delta_1) \circ_2 x_3)\,(1+\delta_2),$$

- $E(\delta_1) = E(\delta_2) = 0$.

Lemma (Connolly et al.)

*The rounding errors δ_k generated by SR are **mean independent** random variables:*

$$E[\delta_k \mid \delta_1, \ldots, \delta_{k-1}] = E[\delta_k] = 0$$

- Independence $\Longrightarrow$ **Mean independence** $\Longrightarrow$ uncorrelatedness.

Bounds for sum-product DAGs

For z resulting of a multi-linear sum-product computation graph with n SR operations,

- $\Psi = \frac{\widehat{z} - z}{z}$ is a martingale (generalisation of a random walk)
- $E(\Psi) = 0$
- The error $|\Psi|$ is probabilistically bounded by $\mathcal{O}(\sqrt{n}u)$, a significant improvement over worst-case linear bounds

*For a deeper dive into the theory and optimization of SR, don't miss **EM. El-Arar's talk** coming up next!*

Error Analysis of sum-product algorithms under stochastic rounding de Oliveira Castro, El-Arar, Petit, Sohier, arXiv 2024.

Bounds for multi-linear algorithms

SR sum-product analysis gives error bounds for multi-linear algorithms:

- Dot product $\mathcal{O}(\sqrt{n}.u)$
- Horner's polynomial evaluation $\mathcal{O}(\sqrt{n}.u)$
- Pairwise summation $\mathcal{O}(\sqrt{\log_2 n}.u)$
- Karatsuba multiplication $\mathcal{O}(\sqrt{\log_2 n}.u)$

Lower than linear bounds achieved with RN $\rightarrow$ good for reduced precision

What about non-linear algorithms or complex numerical software with thousands of lines?

$\rightarrow$ Monte Carlo Simulation

MCA: Stott Parker's significant bits

$1.9999999850477848e+00$

$1.9999999957687429e+00$

$2.0000000024646973e+00$

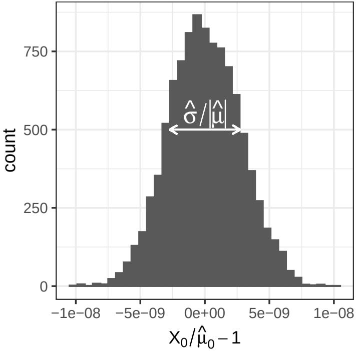


Figure: Error distribution for 10,000 samples MCA on an ill-conditioned system

- Stott Parker defines the number of significant bits as

$$s_{\text{PARKER}} = -\log_2 \frac{\hat{\sigma}}{|\hat{\mu}|} \approx 28.5.$$

$$(s_{\text{IEEE}} \approx 28.8)$$

- Magnitude of the signal to noise ratio.
- We provide confidence intervals depending on number of samples[a]

[a] **Confidence Intervals for Stochastic Arithmetic.** Sohier, de Oliveira Castro, Févotte, Lathuilière, Petit, Jamond. ACM Transactions Mathematical Software 2022.

SR to detect rounding bias in IEEE-754

- **Problem**: Round-to-nearest can be biased. In large summations, small numbers are repeatedly rounded away, causing the sum to stagnate.
- **Solution**: SR's unbiased nature avoids and detects this stagnation.

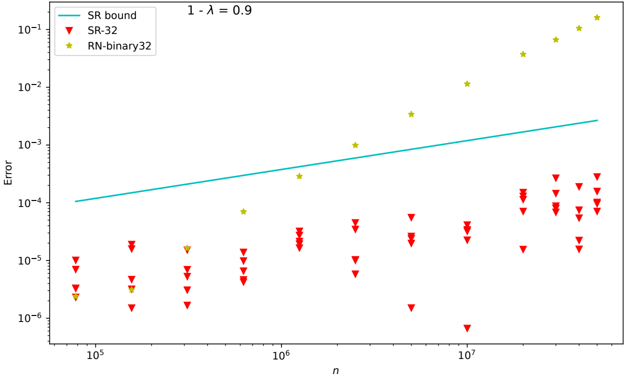
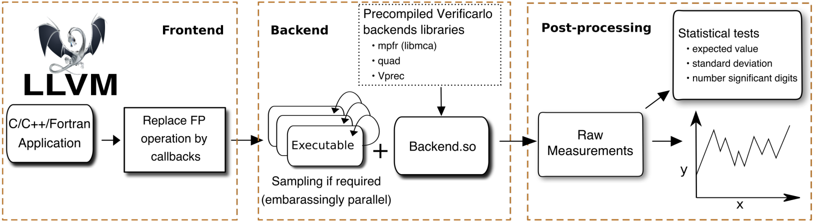


Figure: Dot product error: Round-to-Nearest (RN) stagnates, while SR error grows as expected ($\propto \sqrt{n}$).

Verificarlo

Verificarlo github.com/verificarlo/verificarlo

- An active, open-source project (15 contributors) based on the LLVM compiler.
- Backends: debugging (MCA, PRISM, ...) + mixed-precision (Vprec)
- MCA overhead from ×6 (binary32) to ×160 (binary64). Fast PRISM backend based on Google Highway (Y. Chatelain, 2025).

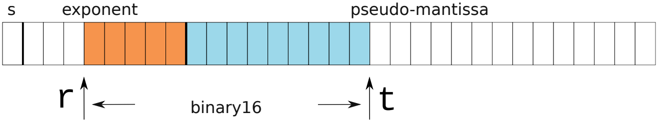


Verificarlo: Checking Floating Point Accuracy through Monte Carlo Arithmetic.
Denis, de Oliveira Castro, Petit. IEEE Symposium on Computer Arithmetic 2016

VPREC for mixed precision

- Safely explore lower precision: Emulate formats like fp32, bfloat16, tf32, on standard hardware *before* paying the high cost of porting.

- VPREC emulates any range and precision within the original type
  - Uses native types for storage and intermediate computations
  - Handle overflows, underflows, denormals, NaN, $\pm\infty$
  - Rounding to nearest (faithful)
  - Fast: $\times$ 2.6 to $\times$ 16.8 overhead



Nekbone: Mixed-Precision CG with Verificarlo

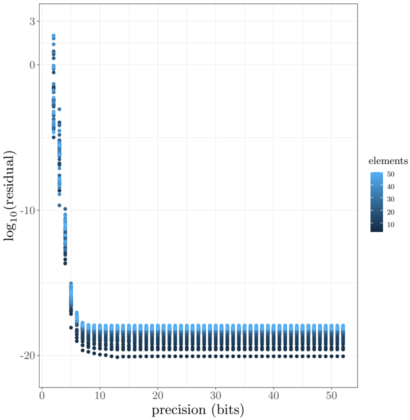


Figure: VPREC precision sweep (3-52 mantissa bits)

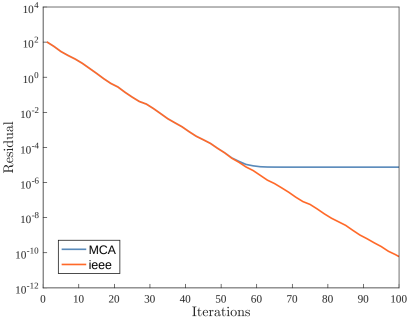


Figure: MCA robustness analysis (29 samples) - with multigrid preconditioner

- **MCA revealed a critical issue**:
  - Robust without multigrid
  - Stagnation with multigrid

Significant Energy-to-Solution Reductions

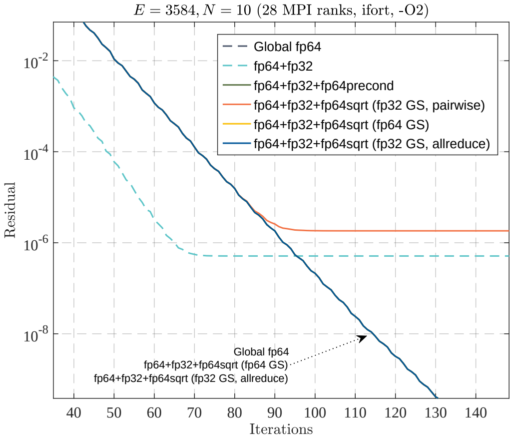


Figure: Residual history with multigrid preconditioner

LUMI-C 128 MPI ranks:

- 47.1% ↓ energy-to-solution
- 40.7% ↓ time-to-solution

Neko (full application):

- ≈30% gains in time/energy

Enabling mixed-precision with the help of tools: A nekbone case study. Chen, de Oliveira Castro, Bientinesi, Iakymchuk. PPAM 2024

Conclusion

- Verificarlo transparently instruments large codes to analyze and guide mixed-precision implementation.
  - Stochastic Rounding is a powerful technique for finding hidden numerical instabilities.
  - VPREC safely emulates the effect of lower precision on standard hardware *before* porting.

- Collaboration with Y. Chen and R. Iakymchuk on Nekbone and Neko.
- Used on many large codes: ABINIT, Dipy, EPX, Yales2, QMCkl, etc.
- Limitations: costly overhead and data-dependent analysis.
- Energy/Precision/Frequency scaling – but Jevons Paradox...

Thanks !

Acknowledgments:

- Verificarlo : Y. Chatelain (Concordia), and all contributors contributors (INTEL, UPVD, CEA, CNRS, etc.)
- CEEC Energy/Precision compromise : Y. Chen (UMEA), R. Iakymchuk (UU / UMEA)
- Stochastic Rounding : EM. El-Arar (INRIA), D. Sohier (UVSQ), E. Petit (INTEL)