Adaptive Mixed Precision Kernel Recursive Least Squares

JunKyu Lee, Hans Vandierendonck, Dimitrios S. Nikolopoulos

Entrans
PRECOMP
Open Transprecision Computing
QUEEN'S UNIVERSITY BELFAST
ECIT
THE INSTITUTE OF ELECTRONICS COMMUNICATIONS AND INFORMATION TECHNOLOGY

Key Message from this talk (Two Fold)

- Introduction to Transprecision Computing:
  - What? Why? How?

- Case-Study for Transprecision Computing:
  - Adaptive Mixed Precision Kernel Recursive Least Squares

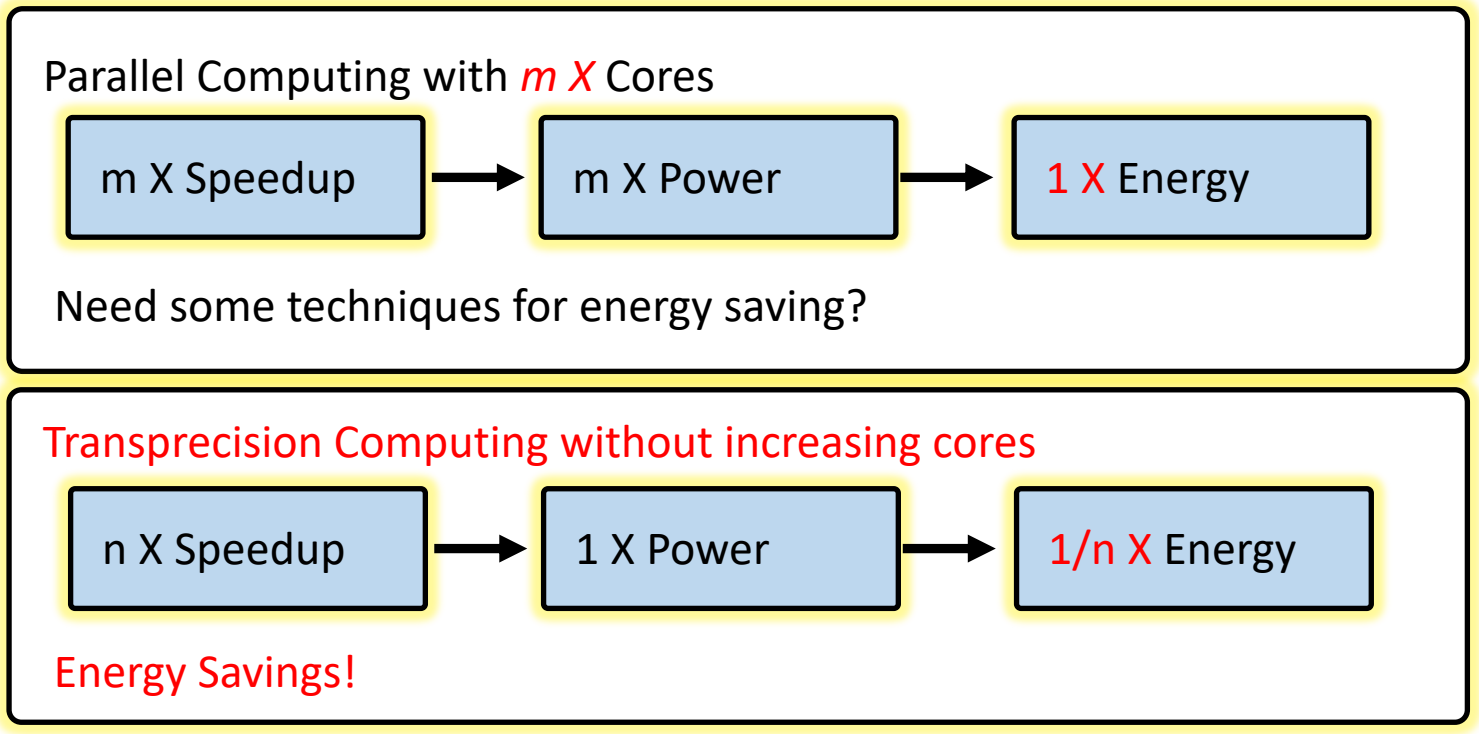
What and Why Transprecision Computing?

Transprecision Technique :

Any (Precision) Technique to Minimise Execution Time/Energy Consumption without Accuracy Loss (or Minor Accuracy Loss) and HW Resource Increment

Transprecision Computing :

Computing utilising Transprecision Techniques



Transprecision Computing on GPUs/FPGAs

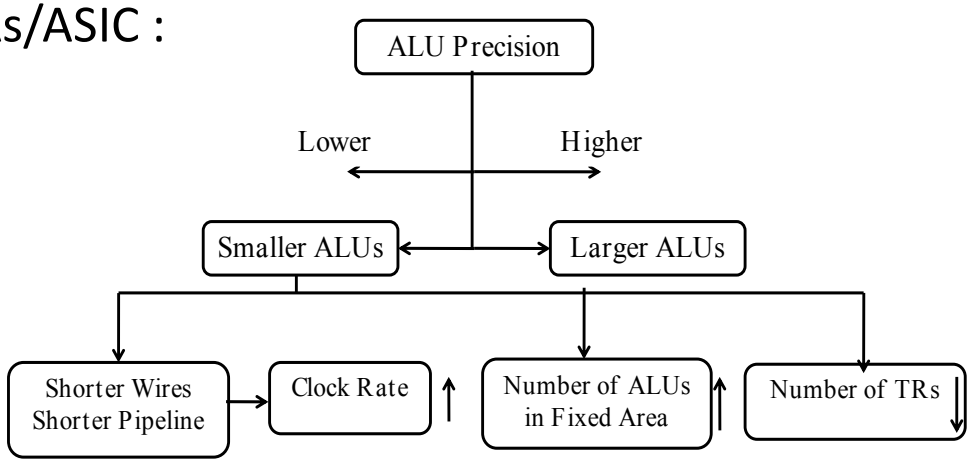
NVidia Pascal GPU (P100) with NVLink:

Half precision: 21.2 TeraFlops
Single precision: 10.6 TeraFlops
Double precision: 5.3 TeraFlops

NVidia Volta GPU (V100) with NVLink:

Half precision (Tensor Core): 125 TeraFlops
Single precision: 15.7 TeraFlops
Double precision: 7.8 TeraFlops

FPGAs/ASIC :

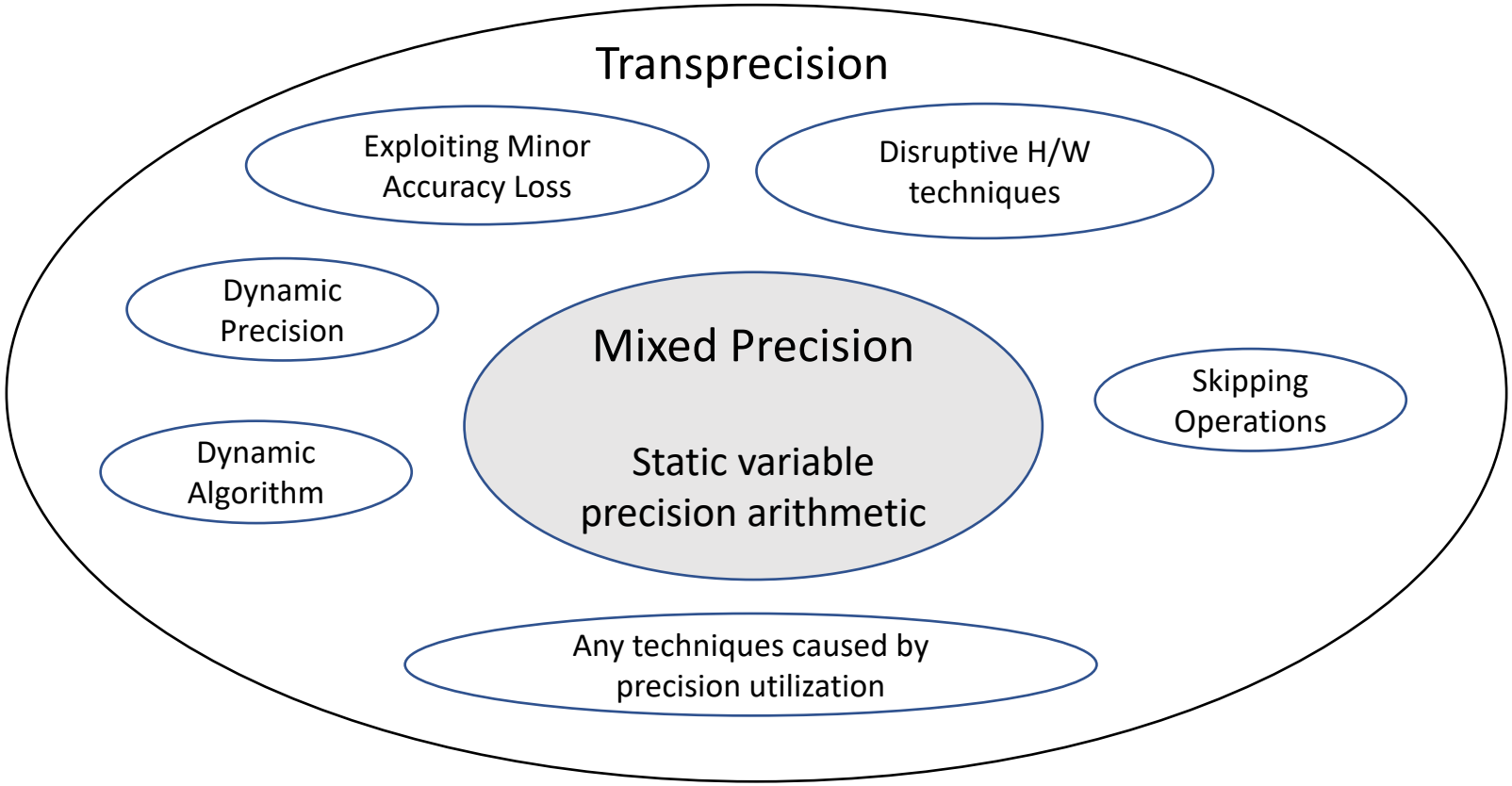


Size of Adder: Linear increase with precision

Size of Multiplier: Quadratic increase with precision

Transprecision (OPRECOMP) vs Mixed Precision

Fast Data transfer from/to Memory => Minimizing Runtime



Transprecision Computing

- Lower Precision, More Energy Savings

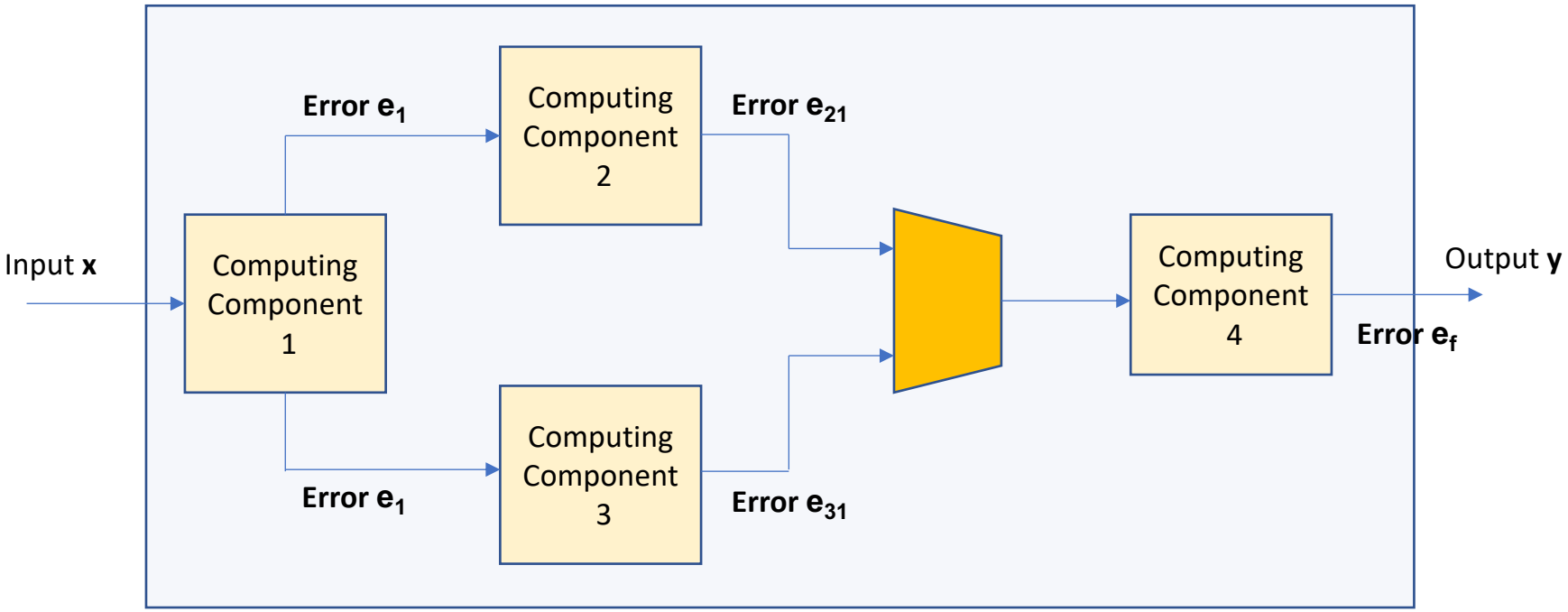
- But, Lower Precision, Lower Accuracy?

- Transprecision Computing! Energy Savings without accuracy loss!

- How?

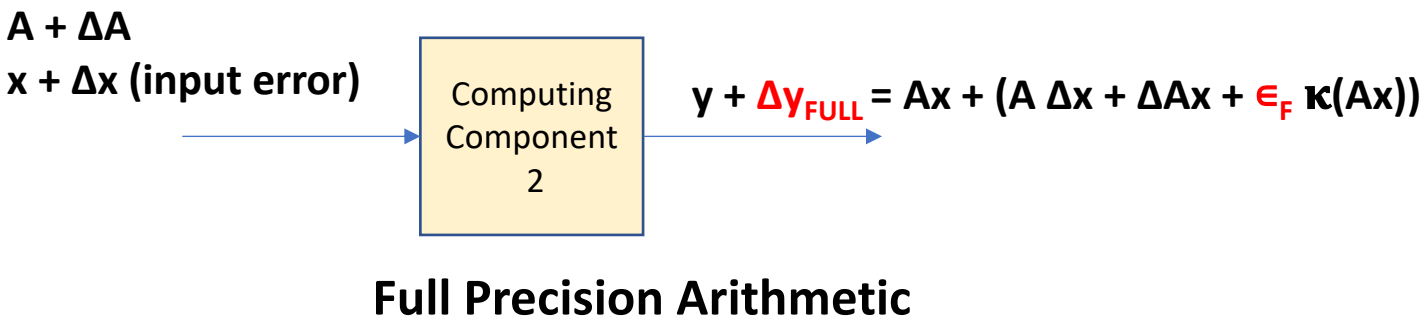
Transprecision Techniques (TTs)

TT 1: To explore error propagation for each computing component



Transprecision Techniques (TTs)

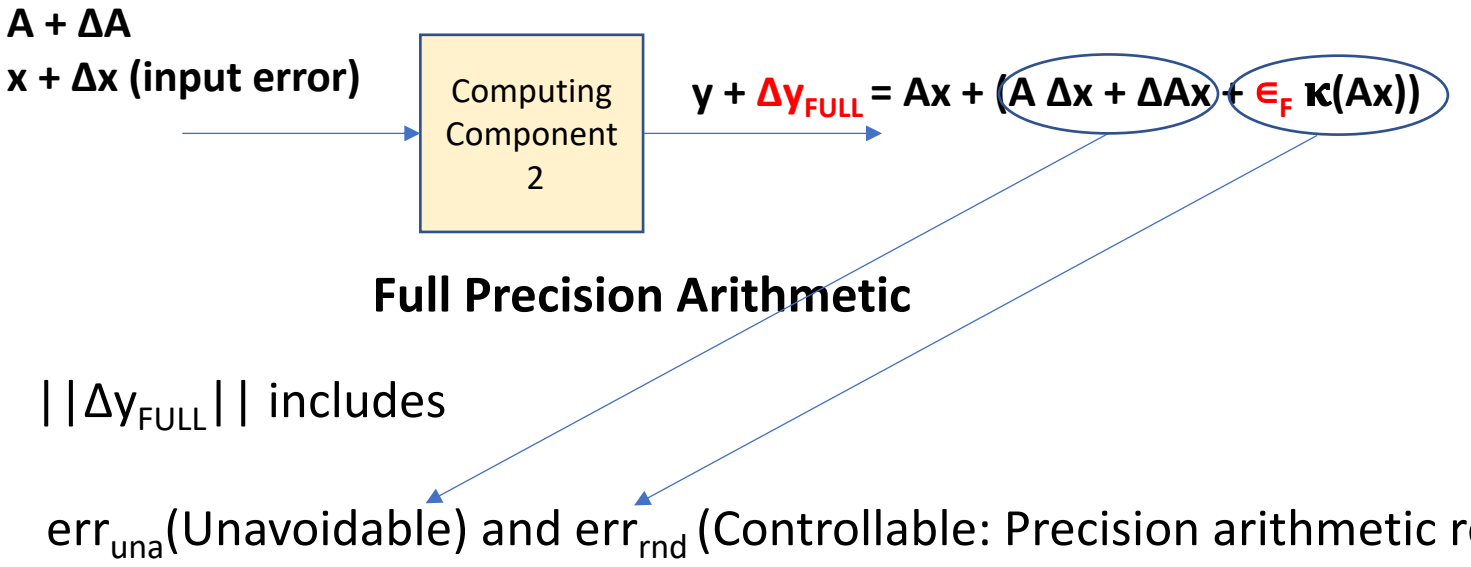
TT 1: To explore error propagation for each computing component



Full Precision Arithmetic

Transprecision Techniques (TTs)

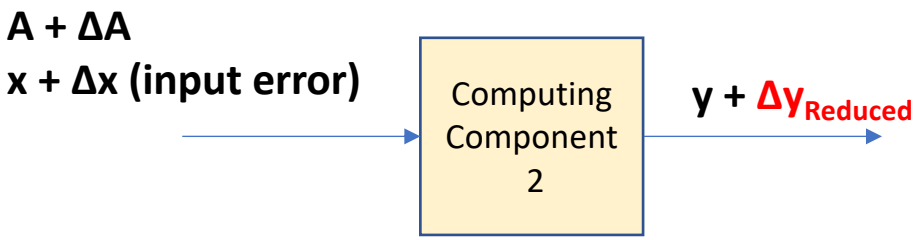
TT 1: To explore error propagation for each computing component



Key Idea: Extremely small ∈ then err_{una} dominant.
You can lower ∈ until $||\Delta y_{FULL}||$ not affected

Transprecision Techniques (TTs)

TT 1: To explore error propagation for each computing component



Reduced Precision Arithmetic

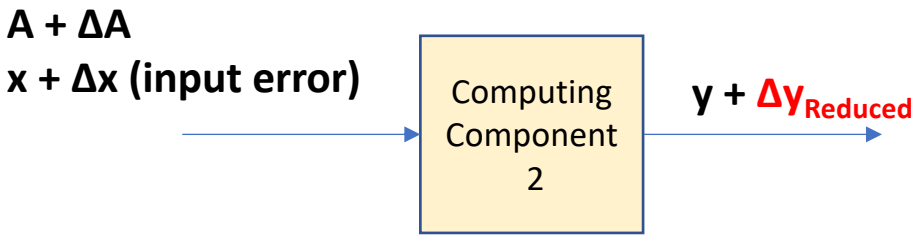
$$err_{una} \gtrsim err_{rnd}$$

When size of Unavoidable Error equivalent of size of rounding off error

=> Reduced Arithmetic can be applied to the Computing Component

Transprecision Techniques (TTs)

TT 1: To explore error propagation for each computing component



Reduced Precision Arithmetic

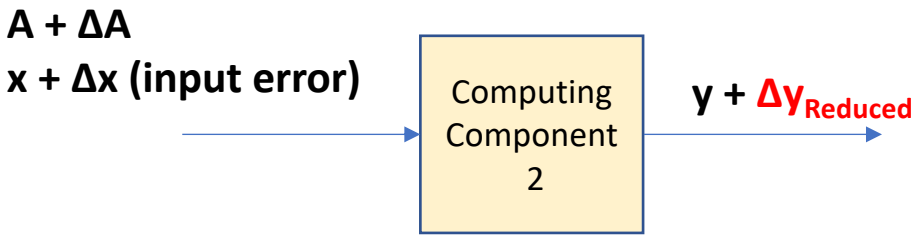
$$||A\Delta x + \Delta A x|| \gtrsim ||\epsilon_R \kappa(Ax)||$$

(unavoidable err) (controllable rounding-off error)

Reduced Arithmetic can be applied to the Computing Component

Transprecision Techniques (TTs)

TT 1: To explore error propagation for each computing component



Reduced Precision Arithmetic

Case 1: Linear Solver

$err_{rnd} \propto \kappa(A)\epsilon_R$

Case 2: Matrix-vector

$err_{rnd} \propto \kappa(A)\epsilon_R$

Case 3: Dot-product

$err_{rnd} \propto (|\mathbf{x}_1|^T|\mathbf{x}_2|)/|\mathbf{x}_1{}^T\mathbf{x}_2|\epsilon_R$

Question is:
How can we use such properties? Let us look at a Case Study with a kernel machine learning algorithm.

Kernel Recursive Least Squares (KRLS)

Applications: Non-linear regressions

- Simple RLS mechanism

- No local minima approach (Always approach to a global minimum)

- Fast Convergence / Good Prediction Accuracy

- Online Adaptive Learning (Learning weights One by One – Fit for large scale Machine Learning)

Kernel Recursive Least Squares (KRLS)

Linear Regression:

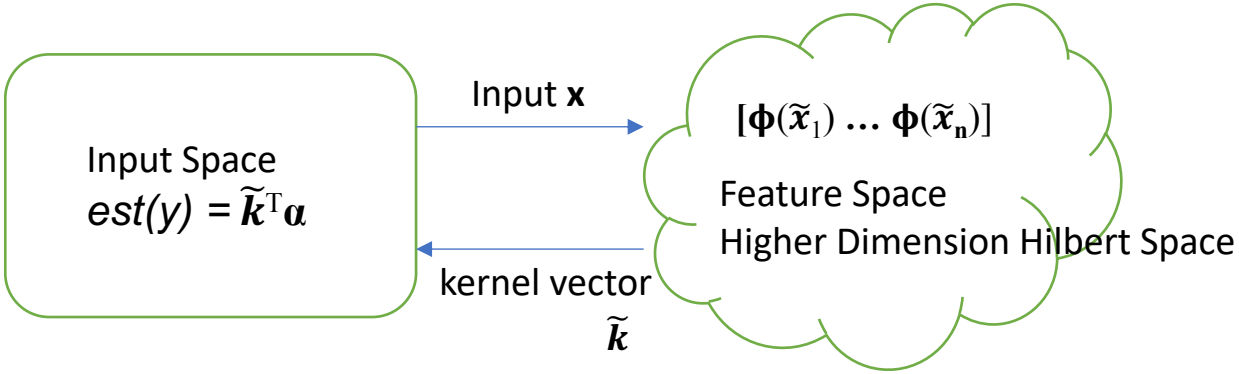
Seek $\mathbf{w} = (w_0, w_1, w_2, w_3)$ to estimate y given an input $\mathbf{x} = (1, x_1, x_2, x_3)$ with

$$est(y) = \mathbf{x}^T\mathbf{w} = w_0\ 1 + w_1\ x_1 + w_2\ x_2 + w_3\ x_3.$$

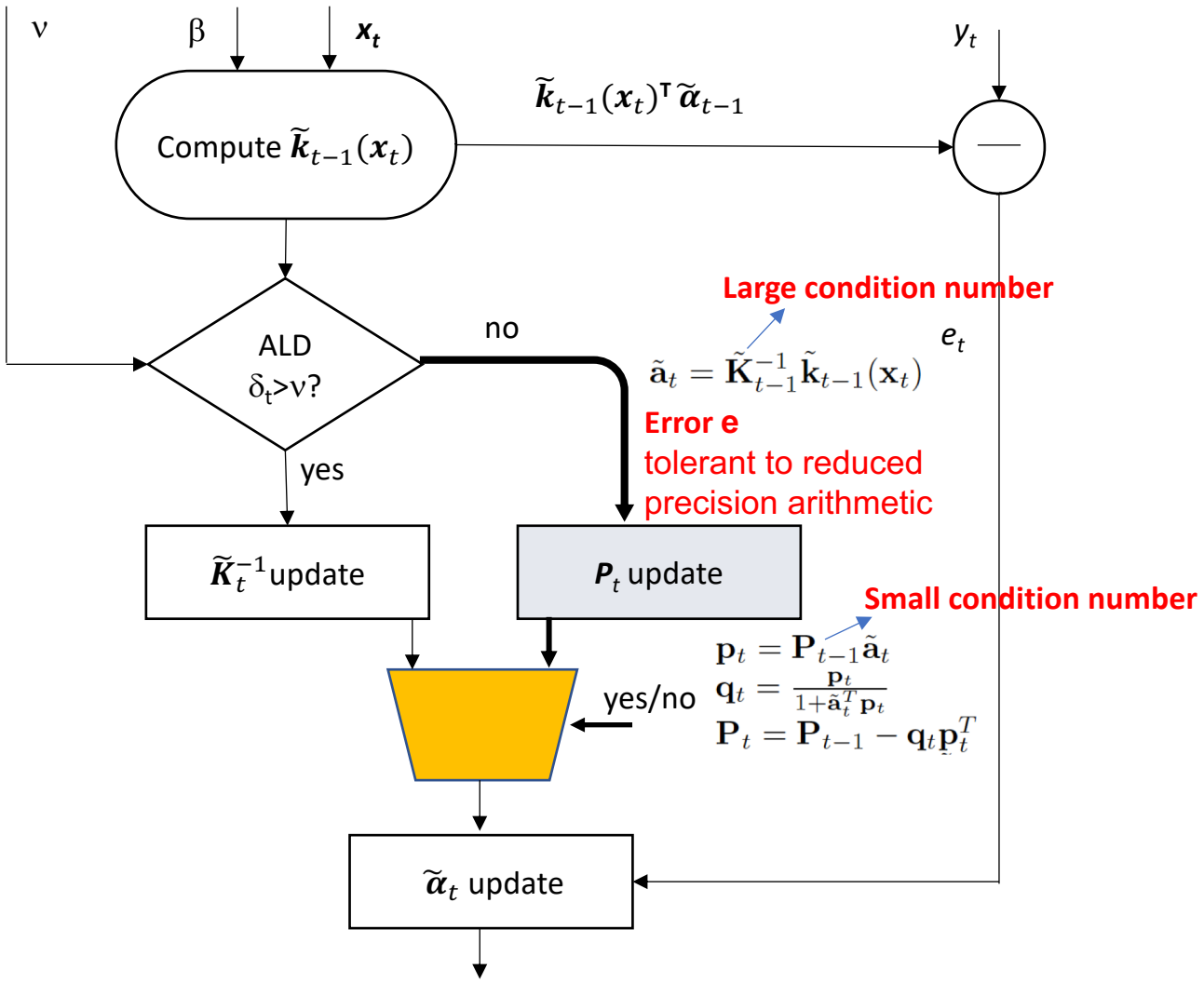
Non-linear Regression (Kernel Method):

Perform linear regression in higher dimension Hilbert space $\mathcal{H}$

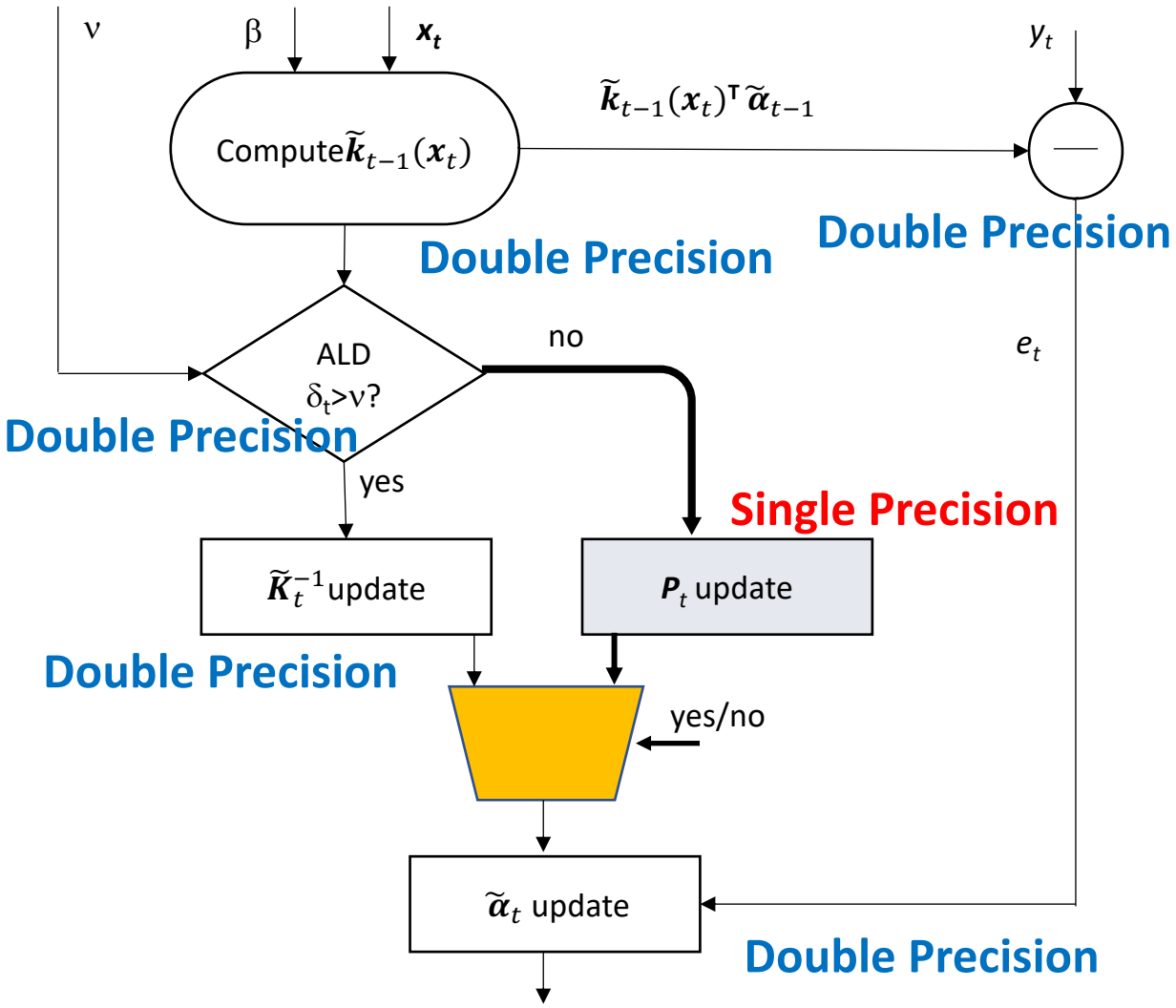
$$est(y) = \boldsymbol{\phi}(\mathbf{x})^T\ \mathbf{w} = \boldsymbol{\phi}(\mathbf{x})^T\ [\boldsymbol{\phi}(\tilde{\boldsymbol{x}}_1)\ \ldots\ \boldsymbol{\phi}(\tilde{\boldsymbol{x}}_n)]\ \boldsymbol{\alpha} = \tilde{\boldsymbol{k}}_.^T\ \boldsymbol{\alpha}$$



Computing Components for KRLS

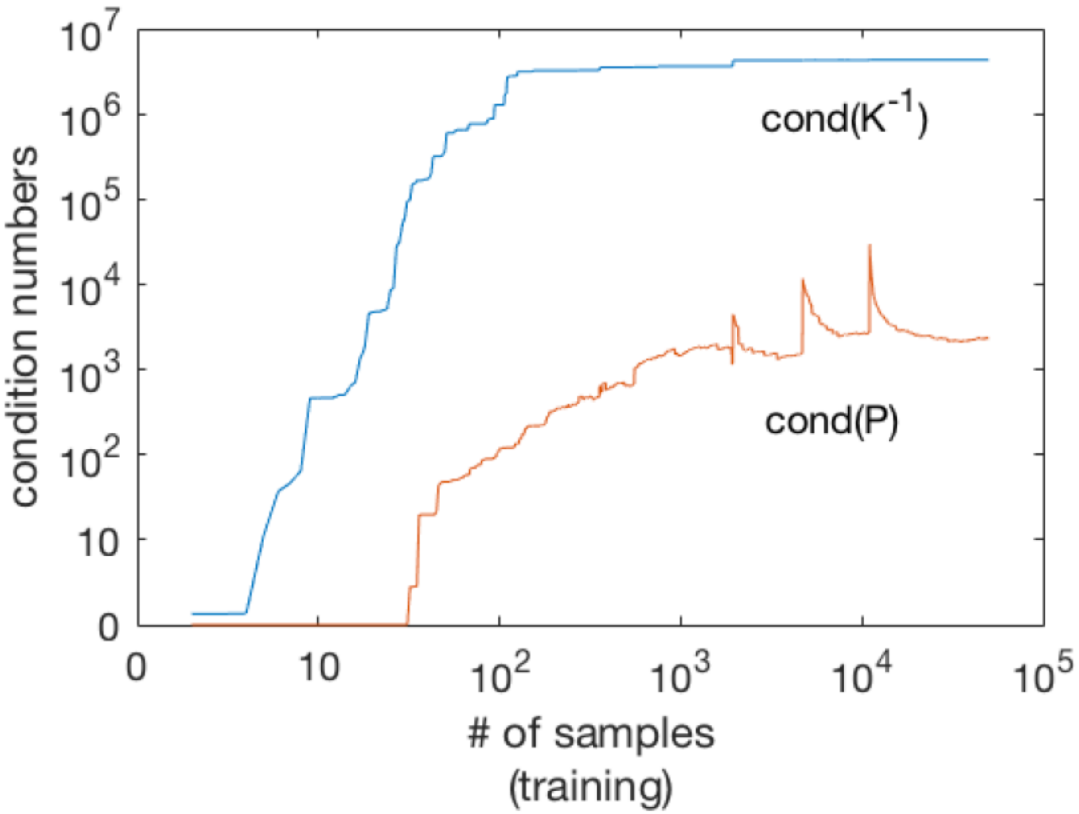


Transprecision Computing for KRLS



Condition Numbers of Matrices in KRLS

Condition number of K^{-1} and P depends on ALD ν and kernel width β
=> Cross Validation decides ν and β



Case Study: Transprecision Computing for KRLS

Non-Linear Regressions: $y = \sin(x_1)/x_1 + x_2/10.0 + \cos(x_3)$

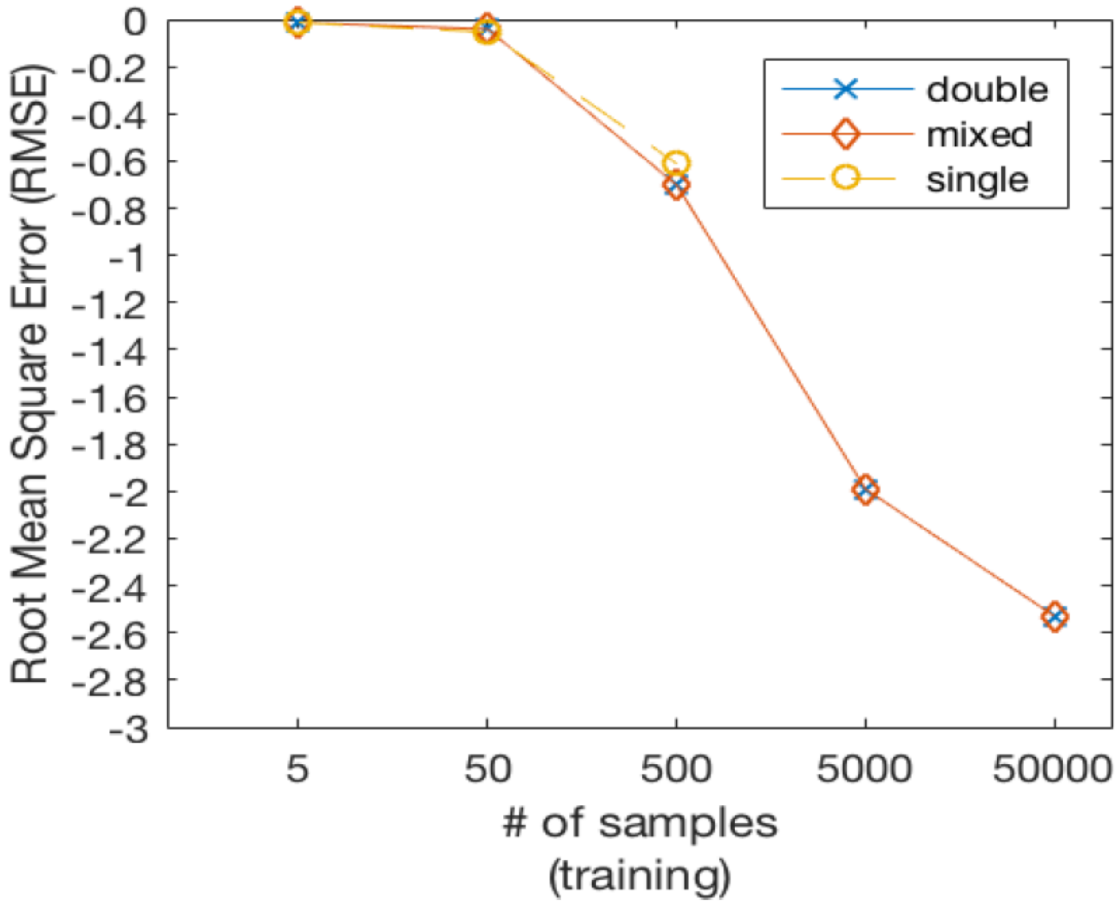
- x_1, x_2, x_3 = Uniform Random [-10, 10]
- Gaussian Kernel Width β = 2.5
- ALD ν = 0.01
- Intel(R) Xeon(R) CPU E5-2650 2GHz Single Core
- Energy Estimation: ALEA Energy Profiling Tool

Smaller ALD ν, generally larger dictionary size, larger condition number of K and better prediction

Observe Prediction Accuracies / Training Time / Energy Consumption

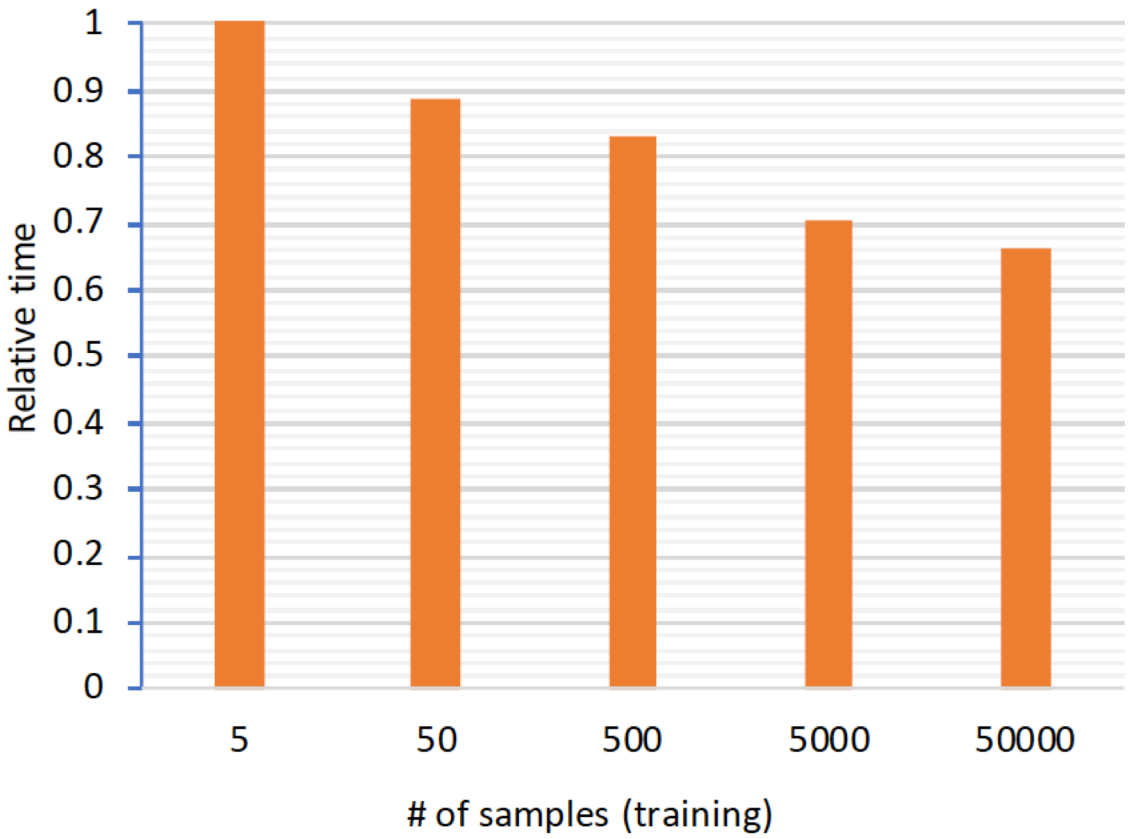
Case Study: Transprecision Computing for KRLS

Prediction Accuracy



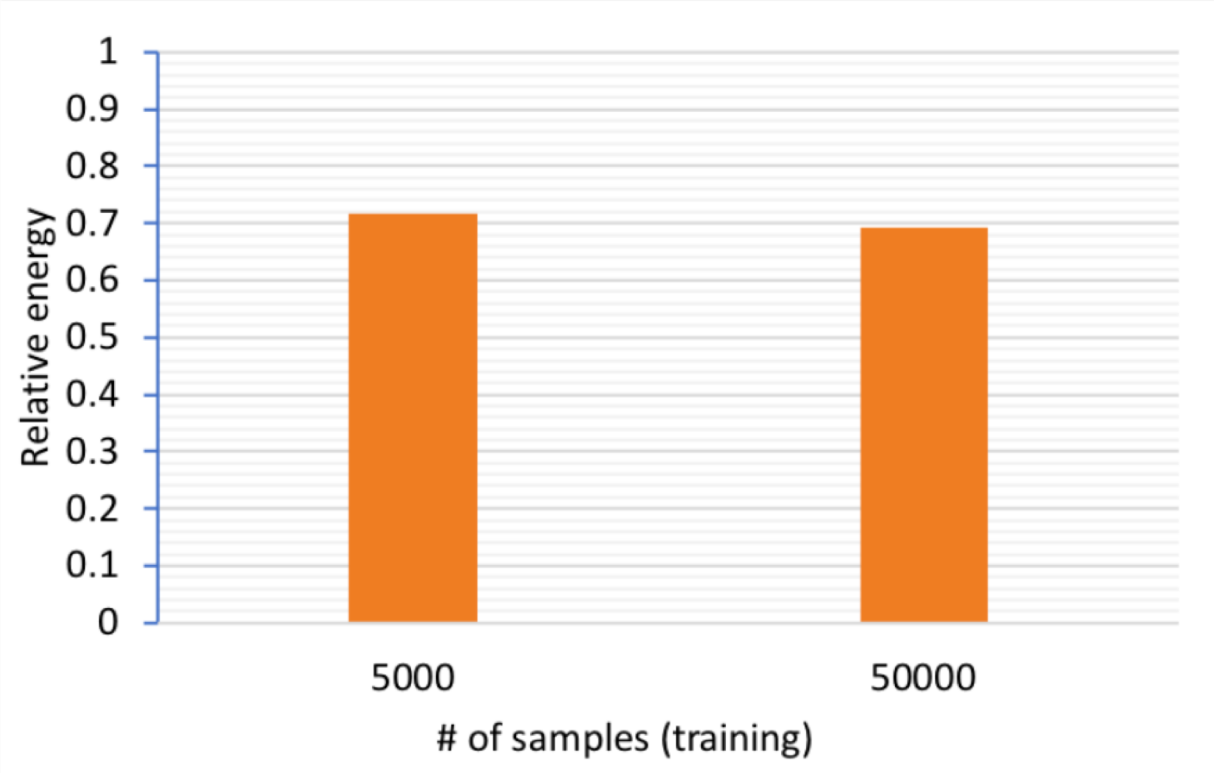
Case Study: Transprecision Computing for KRLS

Execution Time Reduction



Case Study: Transprecision Computing for KRLS

Energy Consumption Reduction



Case Study: Transprecision Computing for KRLS

NOTICE

No Accuracy Loss:
Prediction Accuracies are IDENTICAL (up to 6 digits) between Full precision arithmetic KRLS and Mixed precision KRLS

Speedups and Energy Savings:
Mixed precision KRLS achieves 1.5X Speedups and Energy Savings over Full precision KRLS for Training

Conclusions

Transprecision Computing:
Minimising Execution Time without accuracy loss and HW resource increment

Mixed precision KRLS by Transprecision Techniques brought 1.5X Speedups and Energy Savings without Accuracy Loss compared to Full precision KRLS

Transprecision Computing can be a crucial paradigm for ManyCore in order to achieves both Speedups and Energy Savings.

Transprecision Computing Projects in QUB

Entrans: Energy Efficient Transprecision Techniques for Linear Solver

Jun. 2018 – May 2020

Regularized Transprecision Computing

(No Accuracy Loss, No disruptive HW techniques allowed)

- Aim: To seek energy savings for all types of linear solvers by utilizing transprecision techniques for MultiCore/GPUs.
- H2020 Marie Sklodowska Curie Action Individual Project

OPRECOMP: Open Transprecision Computing

Transprecision Computing

Jan. 2017 – Dec. 2020

(Minor Accuracy Loss, Disruptive HW techniques allowed)

- Aim: To seek energy savings for Machine Learning/Scientific/IoT by utilizing transprecision/approximate computing techniques for MultiCore/GPUs/FPGAs.
- H2020 Project

Thank you very much

Any Questions?