Performance and Fault Tolerance of Preconditioned Iterative Solvers on Low-Power ARM Architectures

José I. Aliaga[1] Sandra Catalán[1] Charalampos Chalios[2]
Dimitrios S. Nikolopoulos[2] Enrique S. Quintana-Ortí[1]

[1]Departamento de Ingeniería y Ciencia de los Computadores, Universitat Jaume I, 12.071–Castellón de la Plana, (Spain)
{aliaga,catalans,quintana}@uji.es

[2]School of EEECS, Queen's University of Belfast (United Kingdom)
cchalios01,d.nikolopoulos@qub.ac.uk>

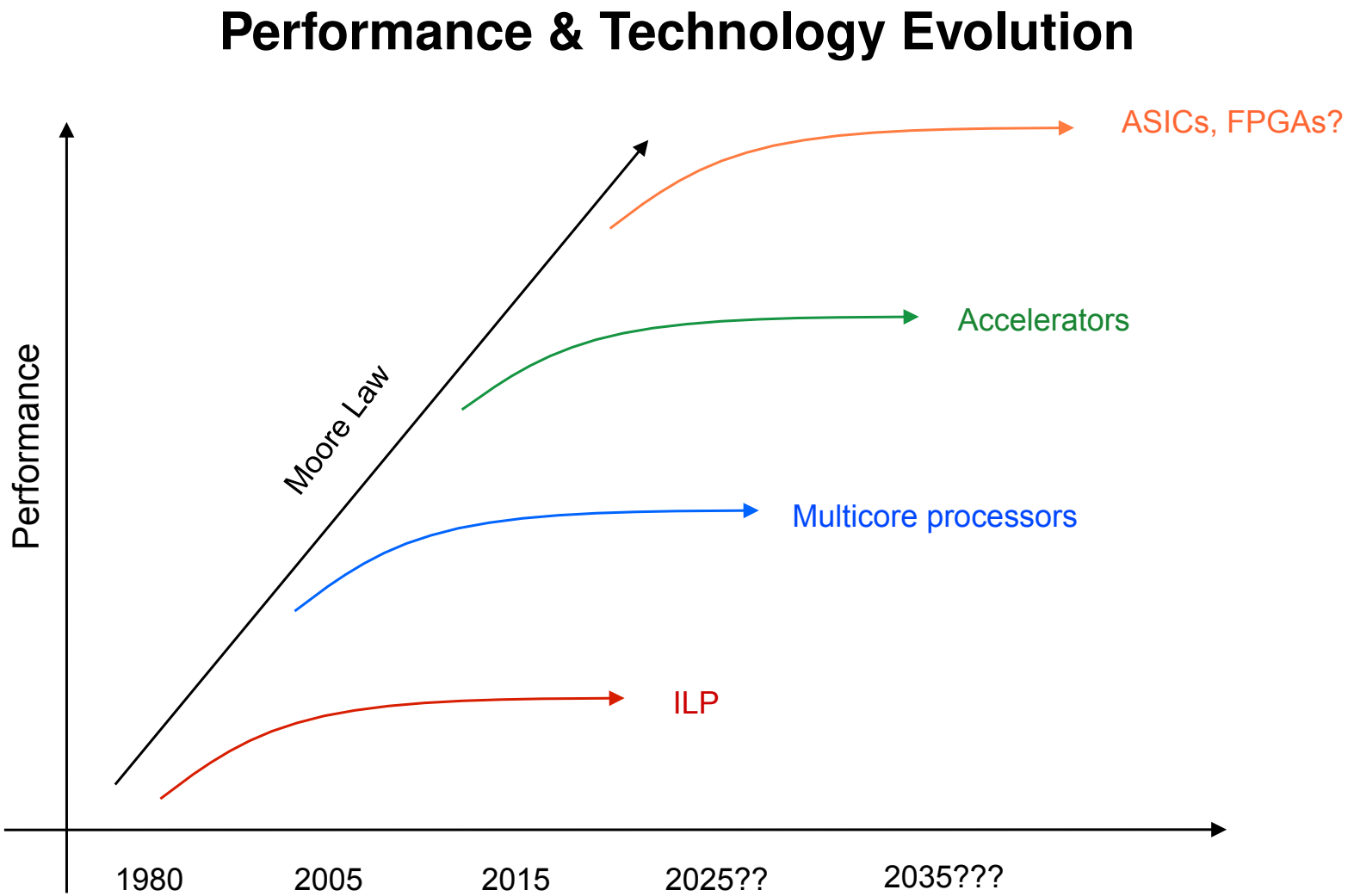
September, 2015

Motivation and Introduction

Performance & Technology Evolution

- Although Moore's Law seems exhausted, it still holds today
  - Solutions to reach sub-10nm technology will give more years
- Dennard scaling broke down around 2005-2007
  - Doubling of the frequency transistor is no longer free
  - Multicores were the alternative to improve performances
  - Challenge: reduce power consump. & energy dissipation per mm^2
- Dark Silicon and heterogeneous designs:
  - Only a fraction of silicon can be active at a moment
  - Use of specialized units reduce the energy consumption
- Near Threshold Voltage Computing (NTVC)
  - Allow to reduce the energy consumption
  - Increase the error rates due to logic failures

Motivation and Introduction



Motivation and Introduction

Performance & Technology Evolution

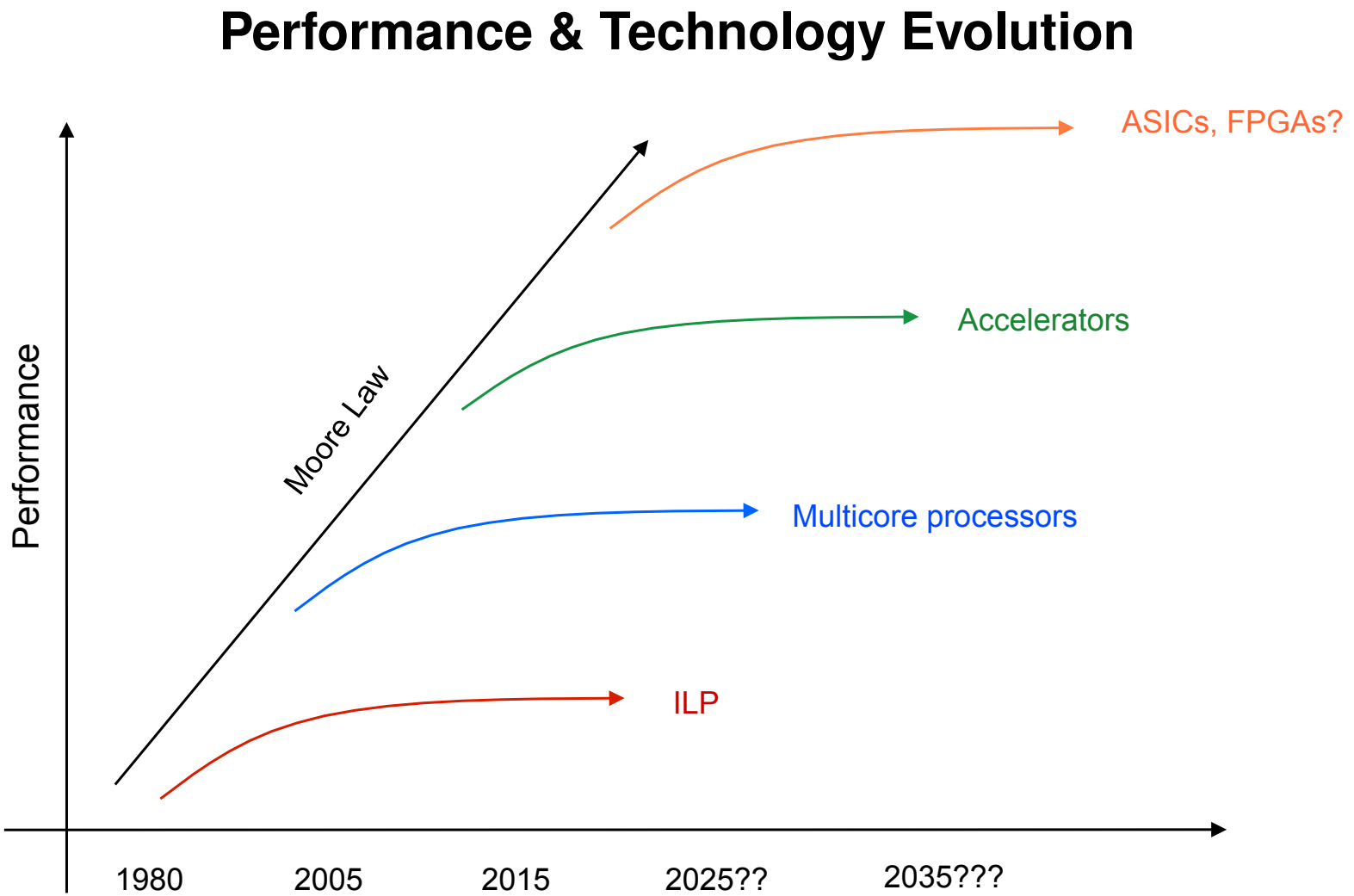
- Although Moore's Law seems exhausted, it still holds today
  - Solutions to reach sub-10nm technology will give more years
- Dennard scaling broke down around 2005-2007
  - Doubling of the frequency transistor is no longer free
  - Multicores were the alternative to improve performances
  - Challenge: reduce power consump. & energy dissipation per mm^2
- Dark Silicon and heterogeneous designs:
  - Only a fraction of silicon can be active at a moment
  - Use of specialized units reduce the energy consumption
- Near Threshold Voltage Computing (NTVC)
  - Allow to reduce the energy consumption
  - Increase the error rates due to logic failures

Motivation and Introduction

Performance & Technology Evolution

- Although Moore's Law seems exhausted, it still holds today
  - Solutions to reach sub-10nm technology will give more years
- Dennard scaling broke down around 2005-2007
  - Doubling of the frequency transistor is no longer free
  - Multicores were the alternative to improve performances
  - Challenge: reduce power consump. & energy dissipation per mm^2
- Dark Silicon and heterogeneous designs:
  - Only a fraction of silicon can be active at a moment
  - Use of specialized units reduce the energy consumption
- Near Threshold Voltage Computing (NTVC)
  - Allow to reduce the energy consumption
  - Increase the error rates due to logic failures

Motivation and Introduction



Motivation and Introduction

Performance & Technology Evolution

- Although Moore's Law seems exhausted, it still holds today
  - Solutions to reach sub-10nm technology will give more years
- Dennard scaling broke down around 2005-2007
  - Doubling of the frequency transistor is no longer free
  - Multicores were the alternative to improve performances
  - Challenge: reduce power consump. & energy dissipation per mm^2
- Dark Silicon and heterogeneous designs:
  - Only a fraction of silicon can be active at a moment
  - Use of specialized units reduce the energy consumption
- Near Threshold Voltage Computing (NTVC)
  - Allow to reduce the energy consumption
  - Increase the error rates due to logic failures

Motivation and Introduction

High Performance Benchmarks

- LINPACK benchmark solves a dense system of linear equations
  - Uses to build TOP500 and GREEN500 lists
  - Gives a good correction of peak performance
- Currently, many applications are defined by sparse matrices
  - The LINPACK performances are useless for real applications
- High Performance Conjugate Gradients (HPCG) benchmark
  - Solves a sparse linear system by using a iterative method
  - Computations and data access patterns more commonly found in applications

Motivation and Introduction

High Performance Benchmarks

- LINPACK benchmark solves a dense system of linear equations
  - Uses to build TOP500 and GREEN500 lists
  - Gives a good correction of peak performance
- Currently, many applications are defined by sparse matrices
  - The LINPACK performances are useless for real applications
- High Performance Conjugate Gradients (HPCG) benchmark
  - Solves a sparse linear system by using a iterative method
  - Computations and data access patterns more commonly found in applications

Motivation and Introduction

High Performance Benchmarks

- LINPACK benchmark solves a dense system of linear equations
  - Uses to build TOP500 and GREEN500 lists
  - Gives a good correction of peak performance
- Currently, many applications are defined by sparse matrices
  - The LINPACK performances are useless for real applications
- High Performance Conjugate Gradients (HPCG) benchmark
  - Solves a sparse linear system by using a iterative method
  - Computations and data access patterns more commonly found in applications

Motivation and Introduction

Objective: Analysis of ILUPACK on low-power ARM multicore

- ILUPACK (Incomplete LU PACKage)
  - Computes a Inverse-based Multilevel ILU preconditioner
  - Applies it to accelerate Krylov-subspace methods
  - In this talk, the task-parallel version is considered
  - Its behaviour is similar to HPCG benchmark
    - Data access patterns
    - Arithmetics-to-memory operations ratios
  - Better suited to multicore processor than HPCG
- For the low-power multicore ARM processors, we focus on
  - Scalability
  - Energy efficiency
  - Fault resilience

Motivation and Introduction

Objective: Analysis of ILUPACK on low-power ARM multicore

- ILUPACK (Incomplete LU PACKage)
  - Computes a Inverse-based Multilevel ILU preconditioner
  - Applies it to accelerate Krylov-subspace methods
  - In this talk, the task-parallel version is considered
  - Its behaviour is similar to HPCG benchmark
    - Data access patterns
    - Arithmetics-to-memory operations ratios
  - Better suited to multicore processor than HPCG
- For the low-power multicore ARM processors, we focus on
  - Scalability
  - Energy efficiency
  - Fault resilience

Motivation and Introduction

Objective: Analysis of ILUPACK on low-power ARM multicore

- ILUPACK (Incomplete LU PACKage)
  - Computes a Inverse-based Multilevel ILU preconditioner
  - Applies it to accelerate Krylov-subspace methods
  - In this talk, the task-parallel version is considered
  - Its behaviour is similar to HPCG benchmark
    - Data access patterns
    - Arithmetics-to-memory operations ratios
  - Better suited to multicore processor than HPCG
- For the low-power multicore ARM processors, we focus on
  - Scalability
  - Energy efficiency
  - Fault resilience

Motivation and Introduction

Objective: Analysis of ILUPACK on low-power ARM multicore

- ILUPACK (Incomplete LU PACKage)
  - Computes a Inverse-based Multilevel ILU preconditioner
  - Applies it to accelerate Krylov-subspace methods
  - In this talk, the task-parallel version is considered
  - Its behaviour is similar to HPCG benchmark
    - Data access patterns
    - Arithmetics-to-memory operations ratios
  - Better suited to multicore processor than HPCG
- For the low-power multicore ARM processors, we focus on
  - Scalability
  - Energy efficiency
  - Fault resilience

Outline

Outline

Inverse-based Multilevel IC factorization

Introducing the inverse-based approach into the IC factorization:

- The (root-free) sparse Cholesky decomposition computes

$$A = LDL^T$$

- In the Incomplete Cholesky (IC) decomp. A is factorized as

$$A = \tilde{L}\tilde{D}\tilde{L}^T + E$$

  where $E \in \mathbb{R}^{n,n}$ is a "small" perturbation matrix consisting of those entries dropped during factorization

- $M = \tilde{L}\tilde{D}\tilde{L}^T$ is applied to the original system $Ax = b$ in order to accelerate the convergence of the PCG solver

$$(\tilde{D}^{-1/2}\tilde{L}^{-1})A(\tilde{L}^{-T}\tilde{D}^{-1/2})$$

Inverse-based Multilevel IC factorization

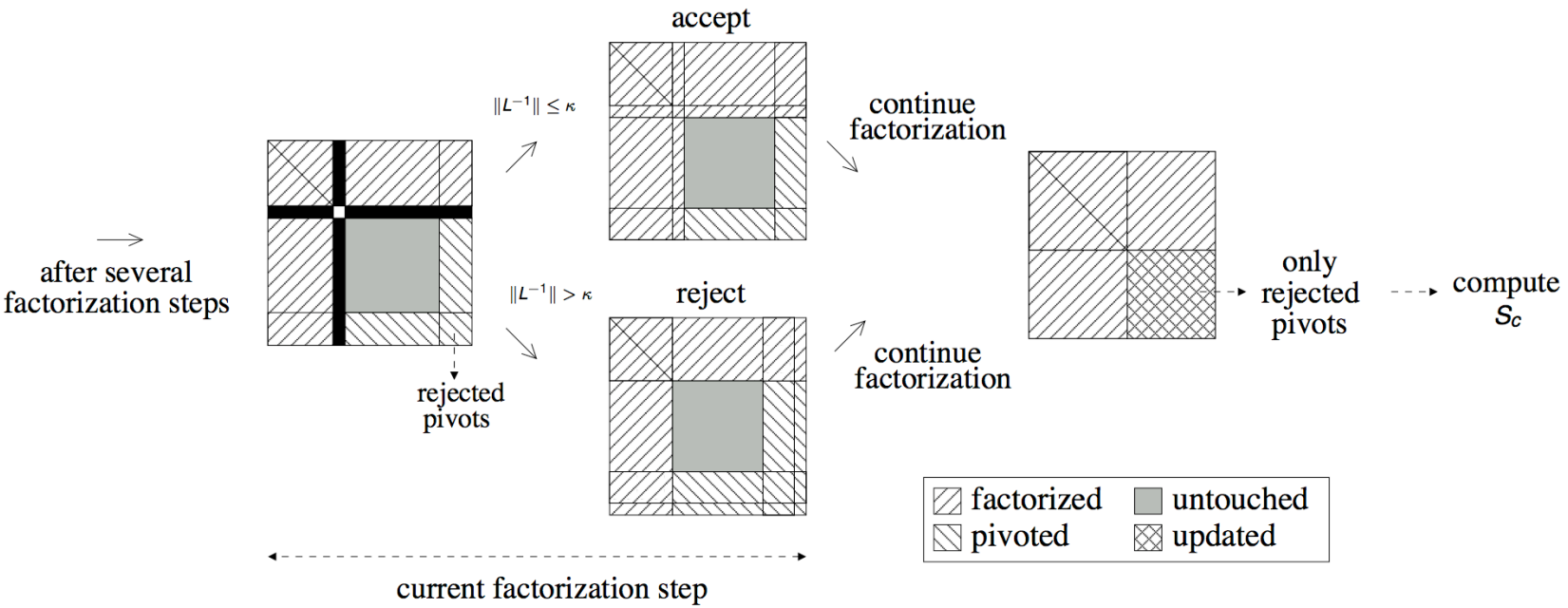
Introducing the inverse-based approach into the IC factorization:

- Inverse-based IC factorizations construct $M = \tilde{L}\tilde{D}\tilde{L}^T$, $\|\tilde{L}^{-1}\| \leq \kappa$
  - $\kappa = 5$ or $\kappa = 10$ are good choices for many problems
  - Pivoting and **multilevel** methods are required in general

Inverse-based Multilevel IC factorization

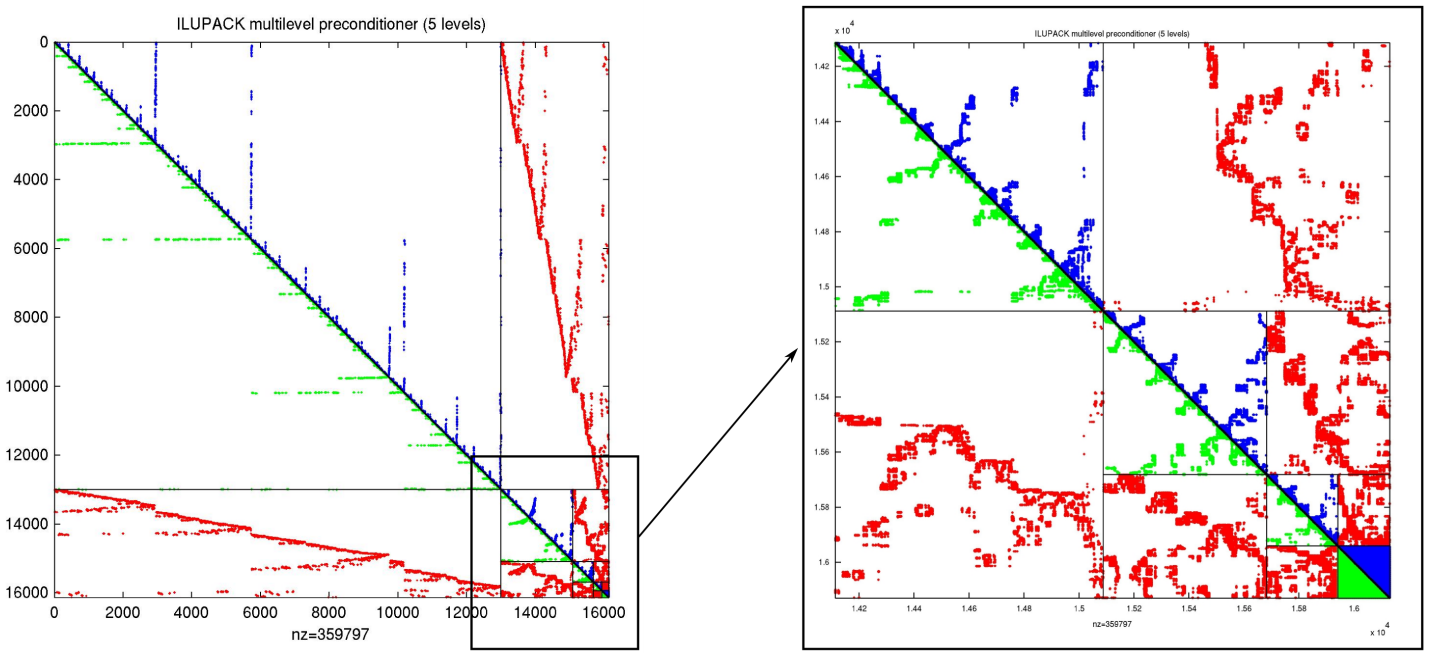
Introducing the inverse-based approach into the IC factorization:

- Inverse-based IC factorizations construct $M = \tilde{L}\tilde{D}\tilde{L}^T$, $\|\tilde{L}^{-1}\| \leq \kappa$
  - $\kappa = 5$ or $\kappa = 10$ are good choices for many problems
  - Pivoting and **multilevel** methods are required in general



Inverse-based Multilevel IC factorization

MIC factorization of five-point matrix arising from Laplace PDE discretization

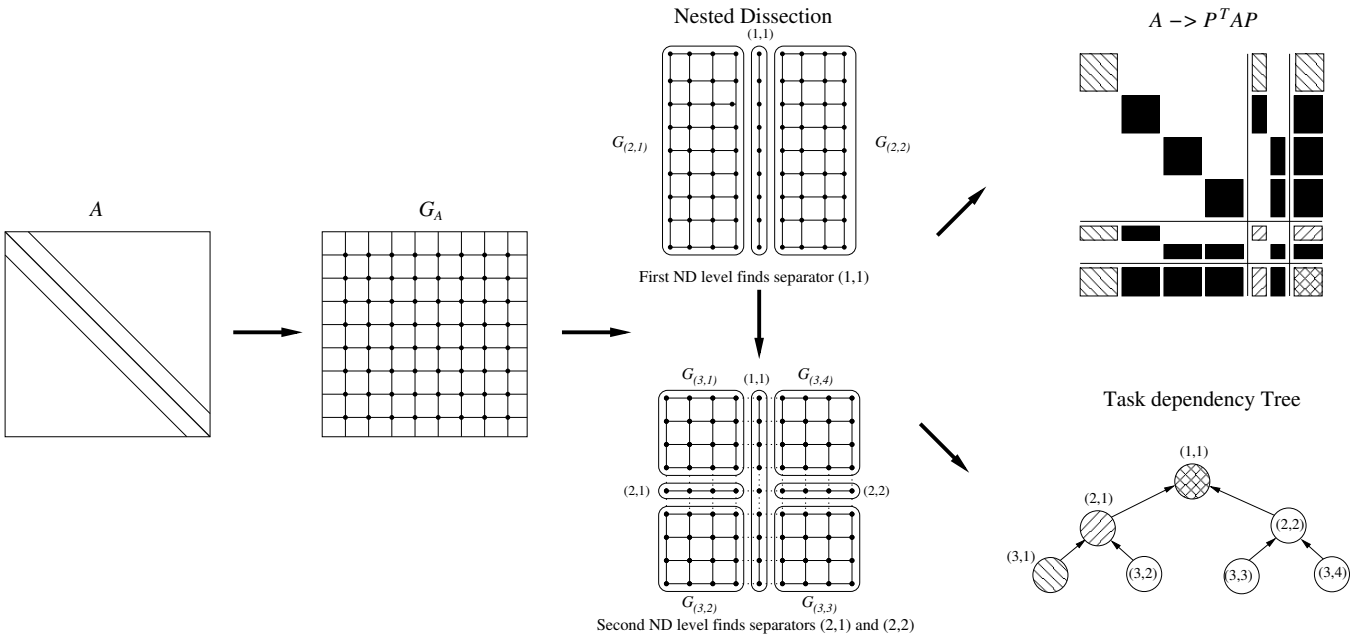


$\kappa = 5$ and $\tau = 10^{-3}$ leads to a MIC factorization with 5 levels

Parallelization of ILUPACK

The key to parallelize the preconditioner computation is to decompose the IC factorization into independent tasks:

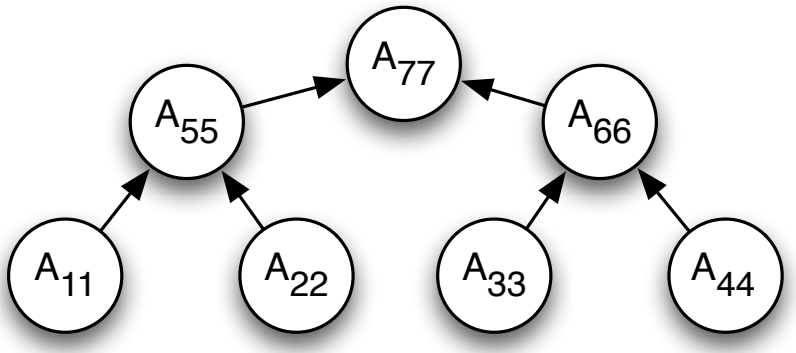
- Nested dissection (ND) reveals task-level parallelism
- METIS and SCOTCH libraries implement multilevel ND



Parallelization of ILUPACK

The key to parallelize the preconditioner computation is to decompose the IC factorization into independent tasks:

- The task dependency tree represents the behaviour of the computation

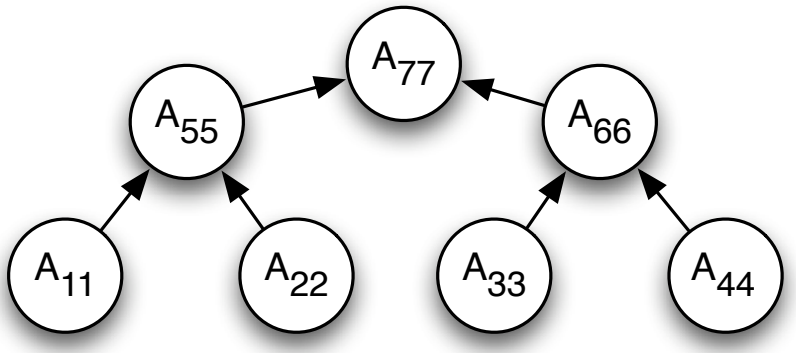


- Each node refers to a block to be factorized, while the edges define the data dependency.
- The nodes without any dependency can be factorized in parallel.
- The non-leaf nodes have to wait until its children have finalized.

Parallelization of ILUPACK

The key to parallelize the preconditioner computation is to decompose the IC factorization into independent tasks:

- The task dependency tree represents the behaviour of the computation

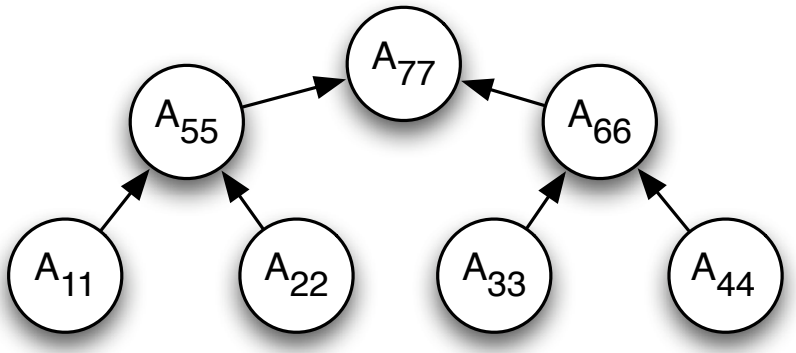


- Each node refers to a block to be factorized, while the edges define the data dependency.
- The nodes without any dependency can be factorized in parallel.
- The non-leaf nodes have to wait until its children have finalized.

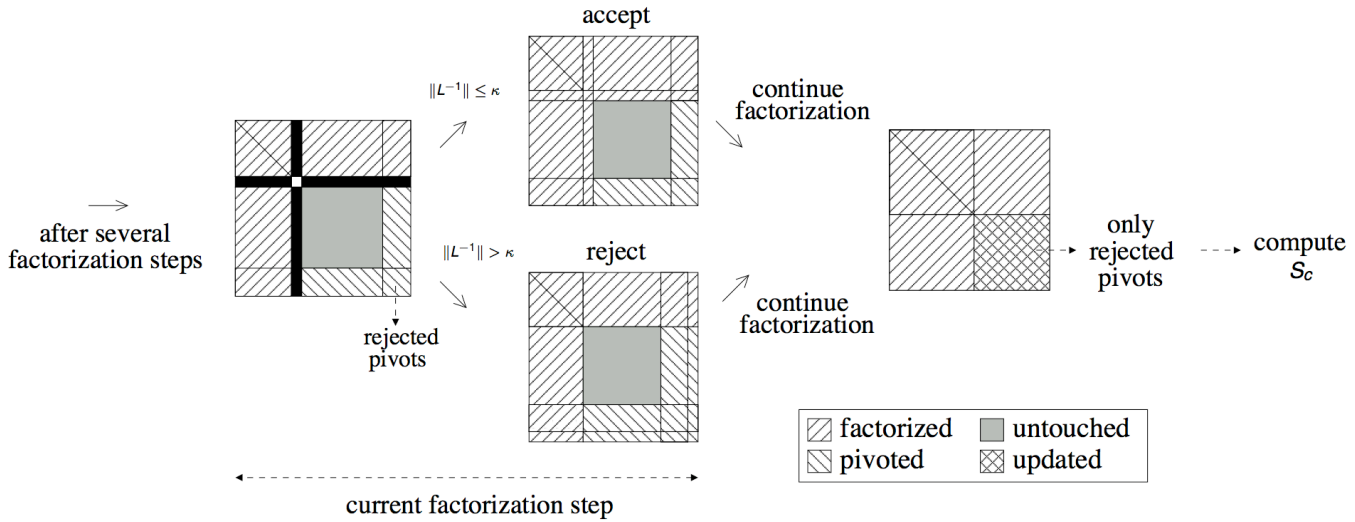
Parallelization of ILUPACK

The key to parallelize the preconditioner computation is to decompose the IC factorization into independent tasks:

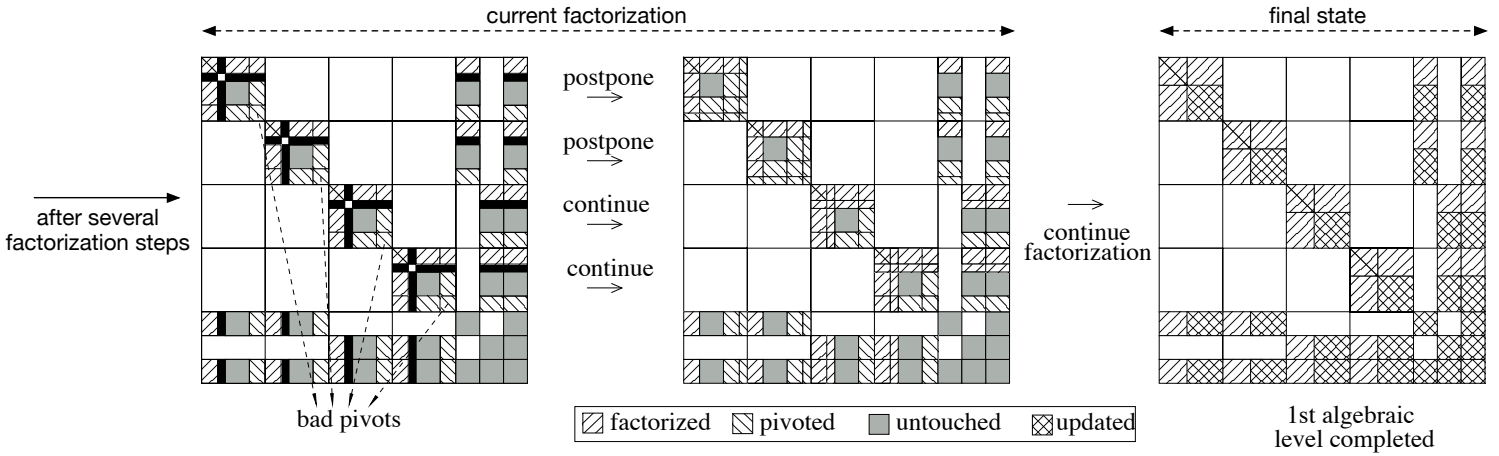
- The task dependency tree represents the behaviour of the computation



- Each node refers to a block to be factorized, while the edges define the data dependency.
- The nodes without any dependency can be factorized in parallel.
- The non-leaf nodes have to wait until its children have finalized.



inverse-based Multilevel IC factorization

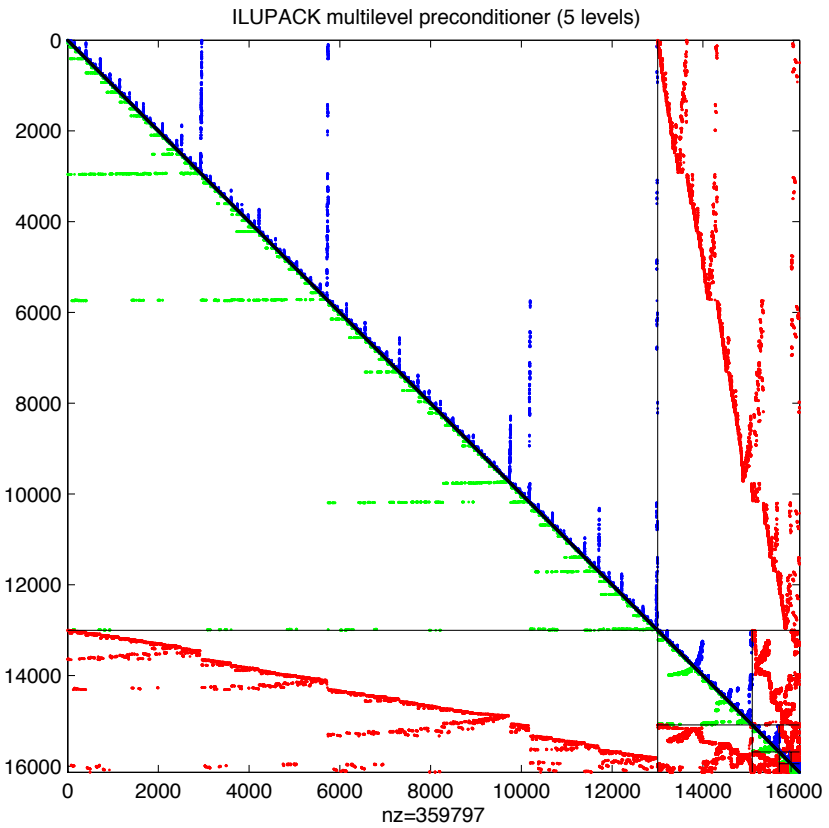


Parallel inverse-based Multilevel IC factorization

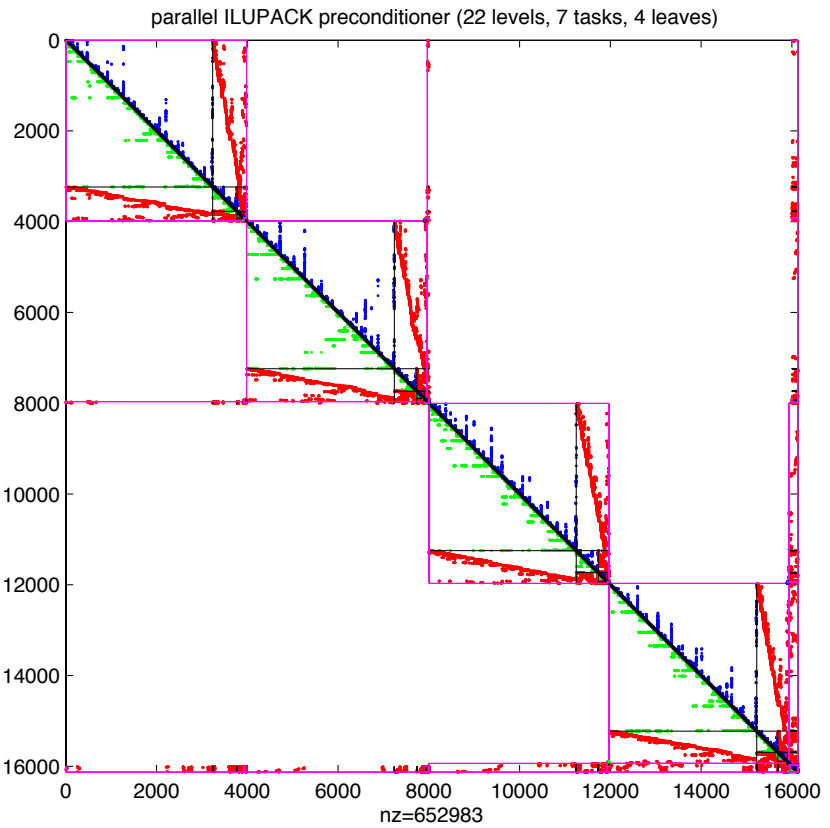
Parallel vs Sequential Inverse-based MIC factoriz.

Factorization of five-point matrix arising from Laplace PDE discretization

MIC (5 levels)



Parallel MIC (4 leaves, 22 levels)

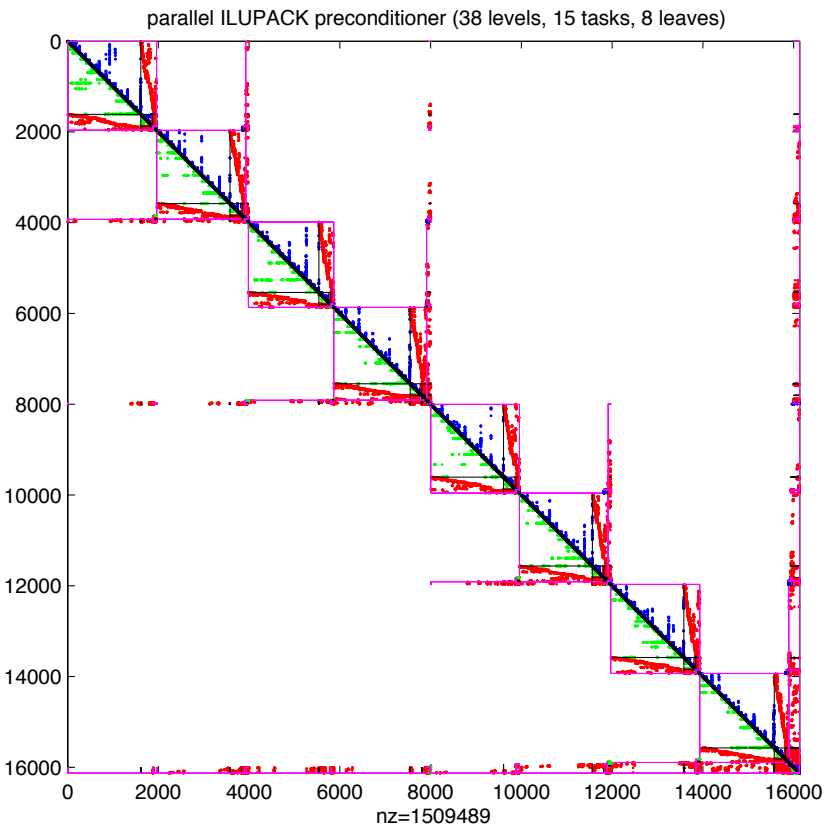


$\kappa = 5$ and $\tau = 10^{-3}$

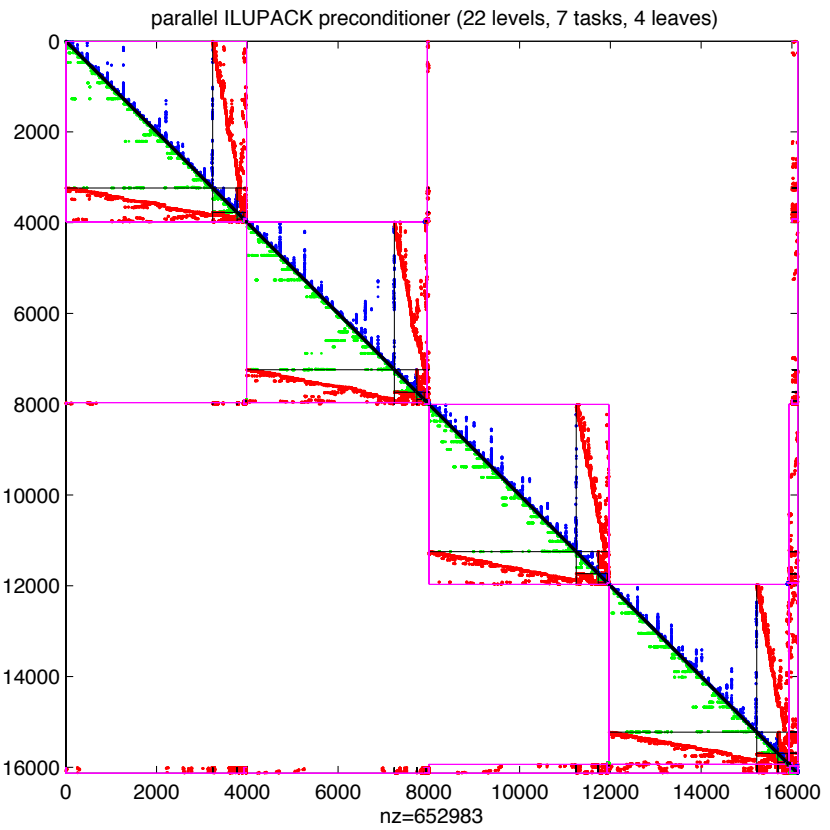
Parallel vs Sequential Inverse-based MIC factoriz.

Factorization of five-point matrix arising from Laplace PDE discretization

Parallel MIC (8 leaves, 38 levels)



Parallel MIC (4 leaves, 22 levels)



$\kappa = 5$ and $\tau = 10^{-3}$

Preconditioned Conjugate Gradient

Initialize $k, x_0, r_0, z_0, d_0, \beta_0, \tau_0$
while $((k < k_{max})\&\&(\tau_k > \tau_{min}))$
$w_k := Ad_k$
$\gamma_k := d_k^T w_k$
$\rho_k := \beta_k / \gamma_k$
$x_{k+1} := x_k + \rho_k d_k$
$r_{k+1} := r_k - \rho_k w_k$
$z_{k+1} := M^{-1} r_{k+1}$
$\alpha_k := \beta_k$
$\beta_{k+1} := r_{k+1}^T z_{k+1}$
$\alpha_k := \beta_{k+1} / \alpha_k$
$d_{k+1} := z_{k+1} + \alpha_k d_k$
$\tau_{k+1} := ||r_{k+1}||_2$
$k := k + 1$
end while

Preconditioned Conjugate Gradient

Initialize $k, x_0, r_0, z_0, d_0, \beta_0, \tau_0$	
while $((k < k_{max})\&\&(\tau_k > \tau_{min}))$	
$w_k := Ad_k$	SPMV
$\gamma_k := d_k^T w_k$	DOT
$\rho_k := \beta_k / \gamma_k$	
$x_{k+1} := x_k + \rho_k d_k$	AXPY
$r_{k+1} := r_k - \rho_k w_k$	AXPY
$z_{k+1} := M^{-1} r_{k+1}$	Preconditioning
$\alpha_k := \beta_k$	
$\beta_{k+1} := r_{k+1}^T z_{k+1}$	DOT
$\alpha_k := \beta_{k+1} / \alpha_k$	
$d_{k+1} := z_{k+1} + \alpha_k d_k$	AXPY-like
$\tau_{k+1} := \|r_{k+1}\|_2$	2-norm
$k := k + 1$	
end while	

Preconditioned Conjugate Gradient

Initialize $k, x_0, r_0, z_0, d_0, \beta_0, \tau_0$
while $((k < k_{max})\&\&(\tau_k > \tau_{min}))$
$w_k := Ad_k$
$\gamma_k := d_k^T w_k$
$\rho_k := \beta_k / \gamma_k$
$x_{k+1} := x_k + \rho_k d_k$
$r_{k+1} := r_k - \rho_k w_k$
$z_{k+1} := M^{-1} r_{k+1}$
$\alpha_k := \beta_k$
$\beta_{k+1} := r_{k+1}^T z_{k+1}$
$\alpha_k := \beta_{k+1} / \alpha_k$
$d_{k+1} := z_{k+1} + \alpha_k d_k$
$\tau_{k+1} := ||r_{k+1}||_2$
$k := k + 1$
end while

SPMV $\rightarrow$ on leaves
DOT $\rightarrow$ on leaves + reduction

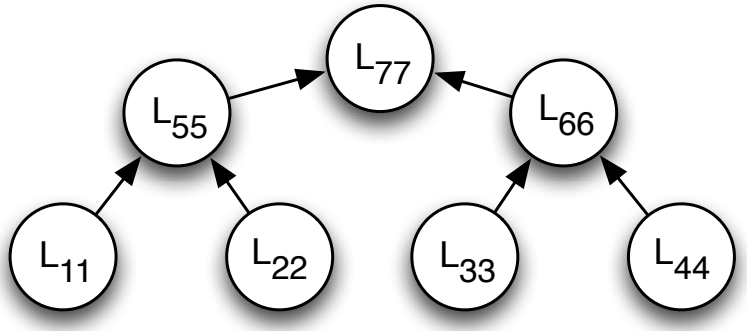
AXPY $\rightarrow$ on leaves
AXPY $\rightarrow$ on leaves
Preconditioning $\rightarrow$ cross. up + down

DOT $\rightarrow$ on leaves + reduction

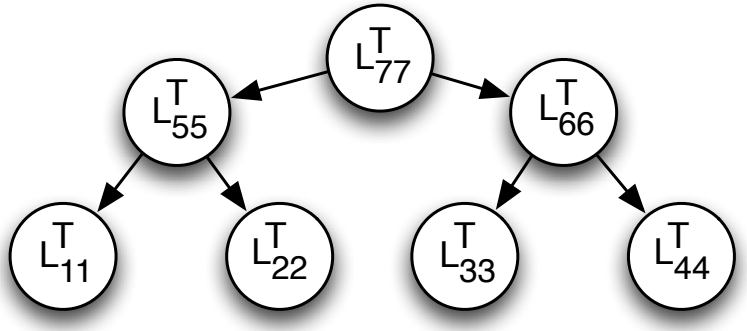
AXPY-like $\rightarrow$ on leaves
2-norm $\rightarrow$ on leaves + reduction

Preconditioner Application and/or Reduction

Preconditioner Forward Resolution (and/or **Reduction Step**):



Preconditioner Backward Resolution (and/or **Copy Result**):



Outline

Hardware Platform

- Exynos5 Odroid-XU development board assembled by Samsung
  - Single big.LITTLE system-on-chip (SoC)
    - ARM Cortex-A15 quad-core cluster
    - ARM Cortex-A7 quad-core cluster
    - The frequency reported by `cpufreq` driver determines the active cluster
  - 2 Gbytes of DDR3 RAM
- We target on low-power scenarios
  - Only consider the ARM Cortex-A7 cores
    - Each core has 32+32 (data+instruction)-KByte L1 cache
    - Each core shares a 512 KByte–L2 cache with its siblings
  - Frequencies $\in \{250, 300, 350, \ldots, 600\}$ MHz

Hardware Platform

- Exynos5 Odroid-XU development board assembled by Samsung
  - Single big.LITTLE system-on-chip (SoC)
    - ARM Cortex-A15 quad-core cluster
    - ARM Cortex-A7 quad-core cluster
    - The frequency reported by `cpufreq` driver determines the active cluster
  - 2 Gbytes of DDR3 RAM
- We target on low-power scenarios
  - Only consider the ARM Cortex-A7 cores
    - Each core has 32+32 (data+instruction)-KByte L1 cache
    - Each core shares a 512 KByte–L2 cache with its siblings
  - Frequencies $\in \{250, 300, 350, \ldots, 600\}$ MHz

Experiments

- Features of the parallel implementation:
  - Compiler: gcc 4.8.1 with the appropriate optimization flags
  - IEEE double precision arithmetic
  - Task tree with 8 leaves
  - Only the PCG results are reported
- Benchmark Matrix:
  - Finite difference discretization of a 3D Laplace problem
  - The grid size determines the size of the matrix
  - n=1,000,000 and nnz=6,940,000

Experiments

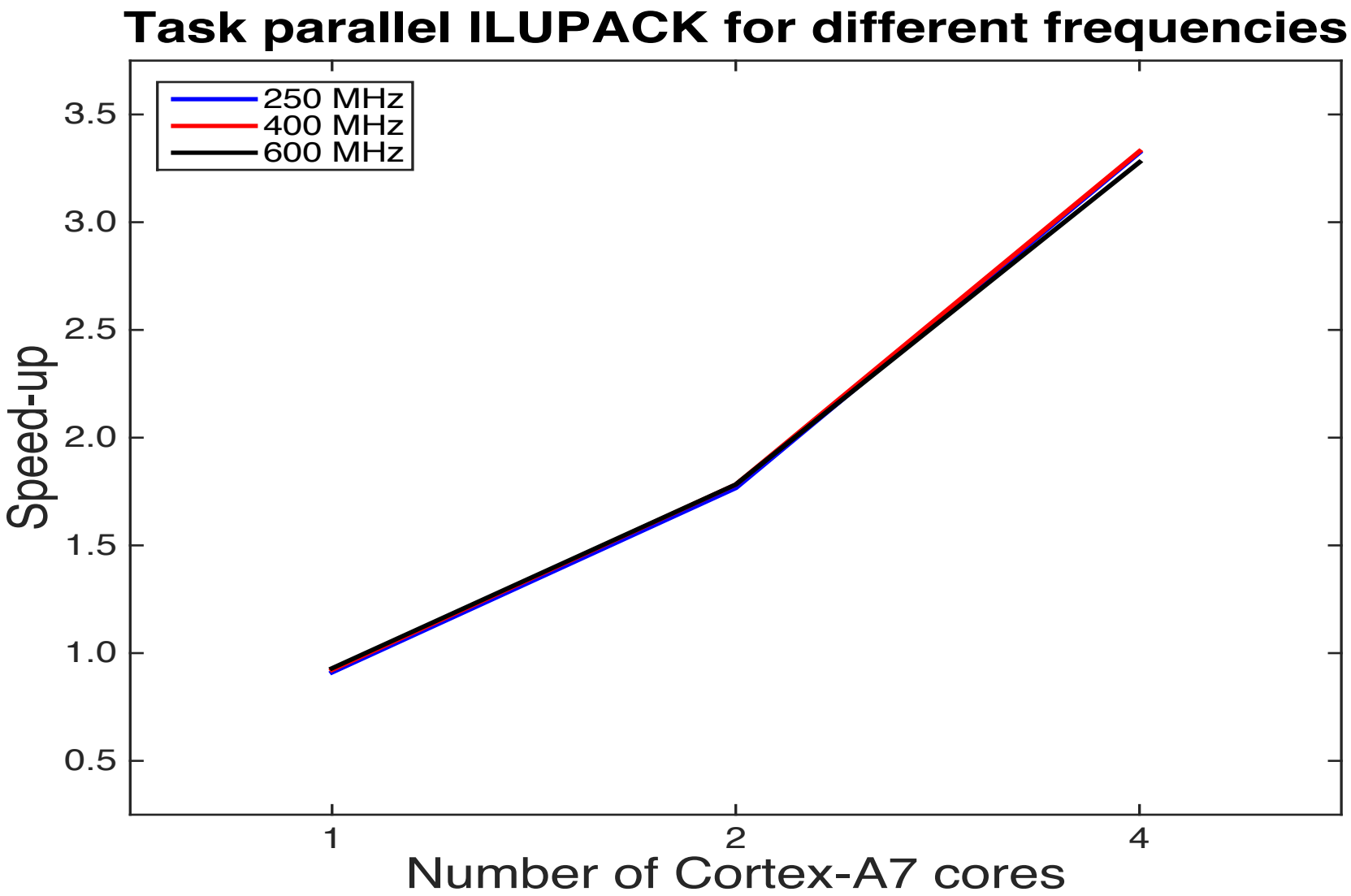
- Features of the parallel implementation:
  - Compiler: gcc 4.8.1 with the appropriate optimization flags
  - IEEE double precision arithmetic
  - Task tree with 8 leaves
  - Only the PCG results are reported
- Benchmark Matrix:
  - Finite difference discretization of a 3D Laplace problem
  - The grid size determines the size of the matrix
  - n=1,000,000 and nnz=6,940,000

Outline

First experiment

- Analyze the scalability of the solver:
  - Changing the number of cores and the frequency
  - The speed-up is computed with respect to the sequential code
  - The results show fair speed-ups
  - The speed-up are mostly independent of the operating frequency
- Features of the frequency-voltage scaling analysis:
  - Execution for different frequencies on 4 Cortex-A7 cores
  - Several metrics:
    - Execution time
    - Power and Energy, considering for both
      Total: $\Rightarrow$ Exynos5 SoC plus the memory DIMMs
      CPU: $\Rightarrow$ Cortex-A7 cluster
  - Polynomial regression is applied to interpolate the results

First experiment



First experiment

- Analyze the scalability of the solver:
  - Changing the number of cores and the frequency
  - The speed-up is computed with respect to the sequential code
  - The results show fair speed-ups
  - The speed-up are mostly independent of the operating frequency
- Features of the frequency-voltage scaling analysis:
  - Execution for different frequencies on 4 Cortex-A7 cores
  - Several metrics:
    - Execution time
    - Power and Energy, considering for both
      Total: $\Rightarrow$ Exynos5 SoC plus the memory DIMMs
      CPU: $\Rightarrow$ Cortex-A7 cluster
  - Polynomial regression is applied to interpolate the results

Analysis for Computational Performance

- The time and the frequency are usually inversely proportional
- To be more precise, some features have to be analyzed
  - To know if the code is compute- or memory-bounded
  - The effect of frequency on the memory bandwidth
- In our experiments,
  - ILUPACK combines compute- and memory-bounded operations
  - In Cortex-A7 cluster the frequency governs the memory bandwidth
- A quadratic polynomial as a function of the frequency works

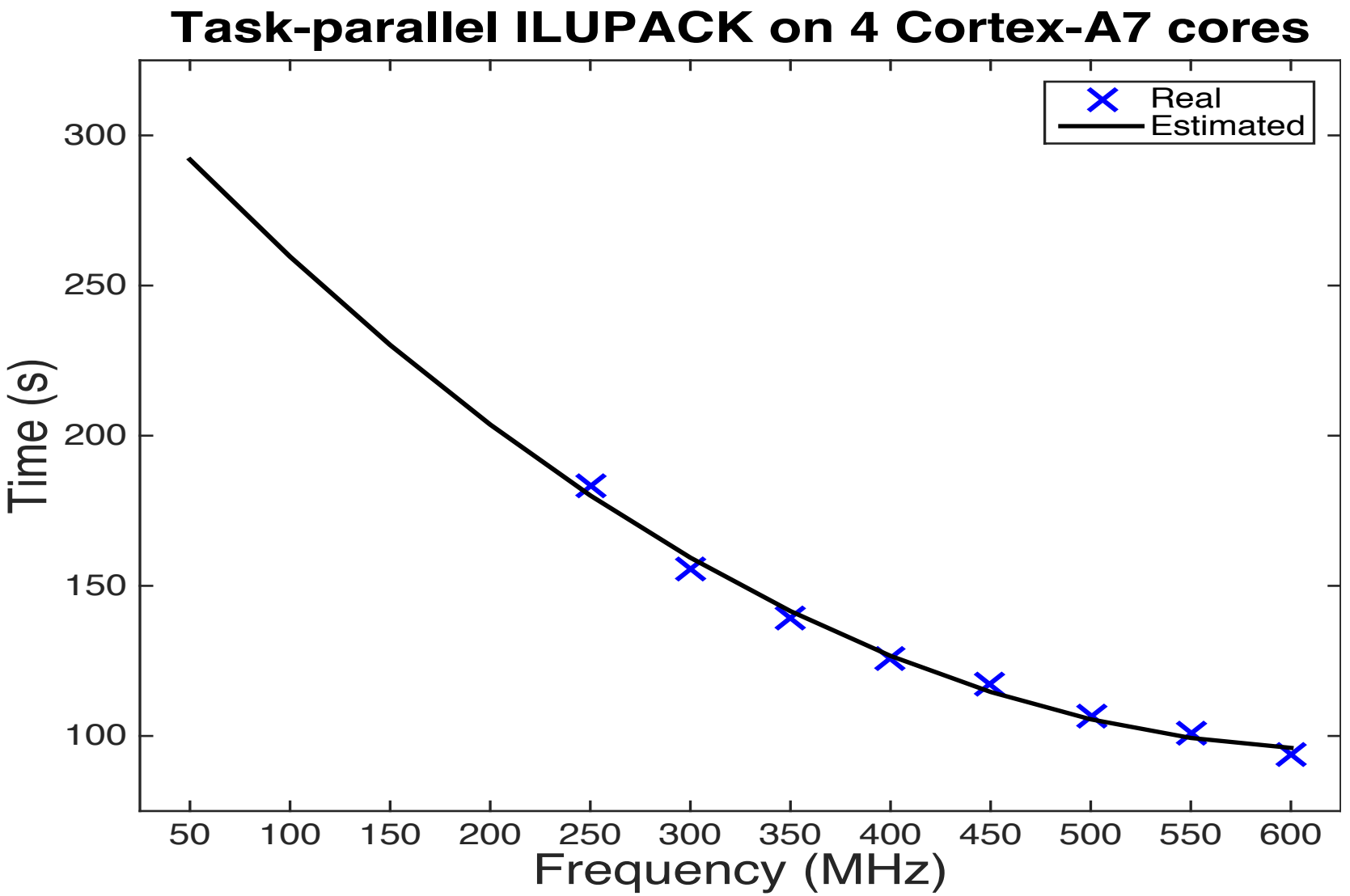
$$T(f) = 5.80\text{E-}4\, f^2 - 7.33\text{E-}1\, f + 3.27\text{E+}2\ s$$
$$1 - r2 = 6.69\text{E-}3$$

Analysis for Computational Performance

- The time and the frequency are usually inversely proportional
- To be more precise, some features have to be analyzed
  - To know if the code is compute- or memory-bounded
  - The effect of frequency on the memory bandwidth
- In our experiments,
  - ILUPACK combines compute- and memory-bounded operations
  - In Cortex-A7 cluster the frequency governs the memory bandwidth
- A quadratic polynomial as a function of the frequency works

$$T(f) = 5.80\text{E-}4\, f^2 - 7.33\text{E-}1\, f + 3.27\text{E+}2\ s$$
$$1 - r2 = 6.69\text{E-}3$$

Analysis for Computational Performance



Analysis for Computational Performance

- The time and the frequency are usually inversely proportional
- To be more precise, some features have to be analyzed
  - To know if the code is compute- or memory-bounded
  - The effect of frequency on the memory bandwidth
- In our experiments,
  - ILUPACK combines compute- and memory-bounded operations
  - In Cortex-A7 cluster the frequency governs the memory bandwidth
- A quadratic polynomial as a function of the frequency works

$$T(f) = 5.80\text{E-}4\, f^2 - 7.33\text{E-}1\, f + 3.27\text{E+}2\, s$$
$$1 - r2 = 6.69\text{E-}3$$

Analysis for power and Energy Efficiency

- The relationship between power and frequency is cubic

$$P \propto V^2 * f \ , \ V \propto f \ \Rightarrow \ P \propto f^3 \ , \ P \propto V^3$$

obtained that

$$P_T(f) = 1.46\text{E-}9\, f^3 + 2.76\text{E-}7\, f^2 - 7.60\text{E-}5\, f + 2.18\text{E-}1 \ W$$
$$1 - r2 = 5.72\text{E-}4$$

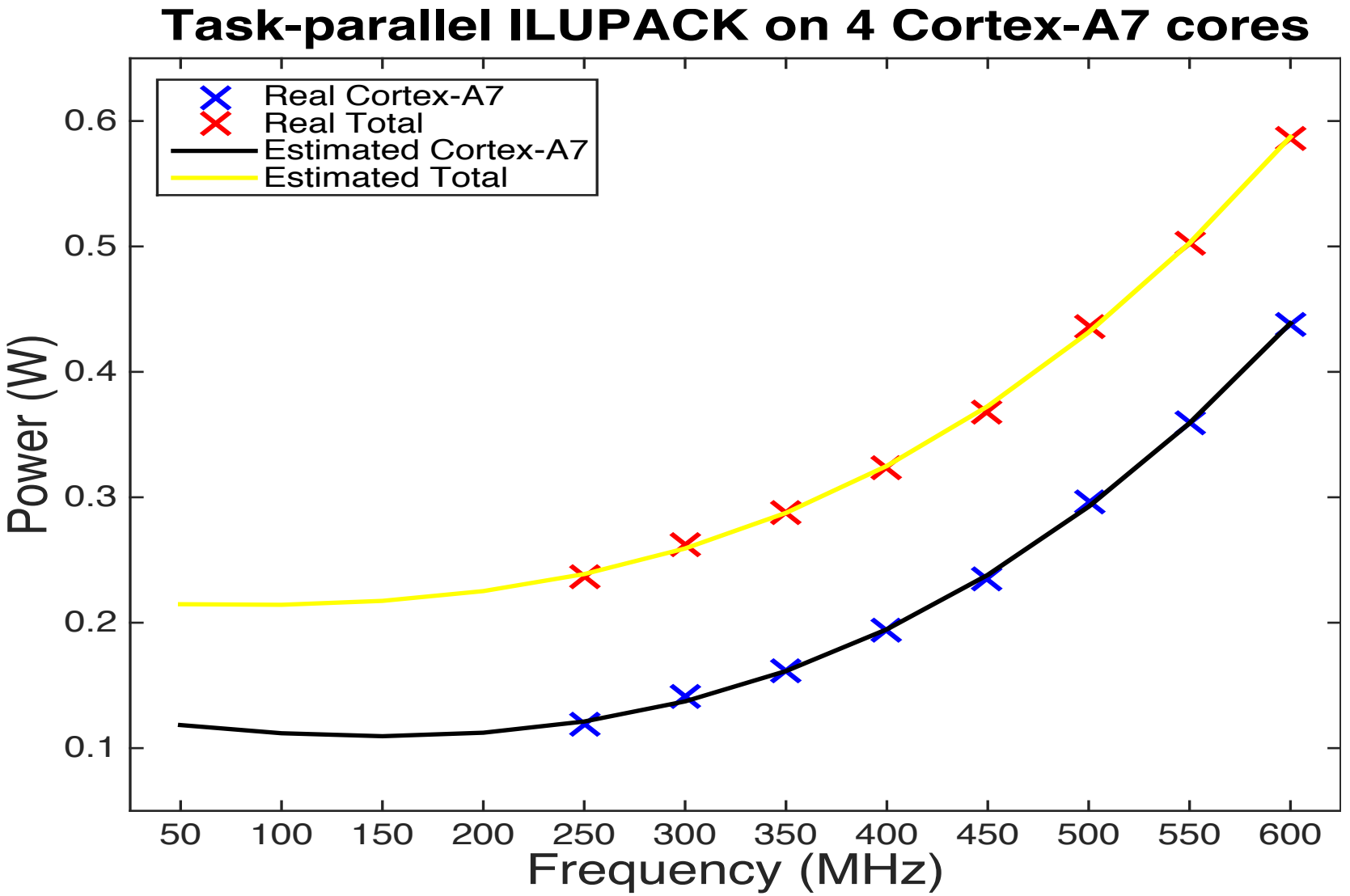
- The energy can be estimated by a quadratic polynomial:

$$E_T(f) = 2.43\text{E-}4\, f^2 - 1.70\text{E-}1\, f + 7.04\text{E+}1$$
$$1 - r2 = 7.36\text{E-}3$$

- The optimal configuration is different for total and for cluster:

$$E_T^{opt} = E_T(350) \ , \ E_{A7}^{opt} = E_{A7}(250) \Rightarrow f_{opt} = 300$$

Analysis for power and Energy Efficiency



Analysis for power and Energy Efficiency

- The relationship between power and frequency is cubic

$$P \propto V^2 * f \ , \ V \propto f \ \Rightarrow \ P \propto f^3 \ , \ P \propto V^3$$

obtained that

$$P_T(f) = 1.46\text{E-}9\, f^3 + 2.76\text{E-}7\, f^2 - 7.60\text{E-}5\, f + 2.18\text{E-}1 \ W$$
$$1 - r2 = 5.72\text{E-}4$$

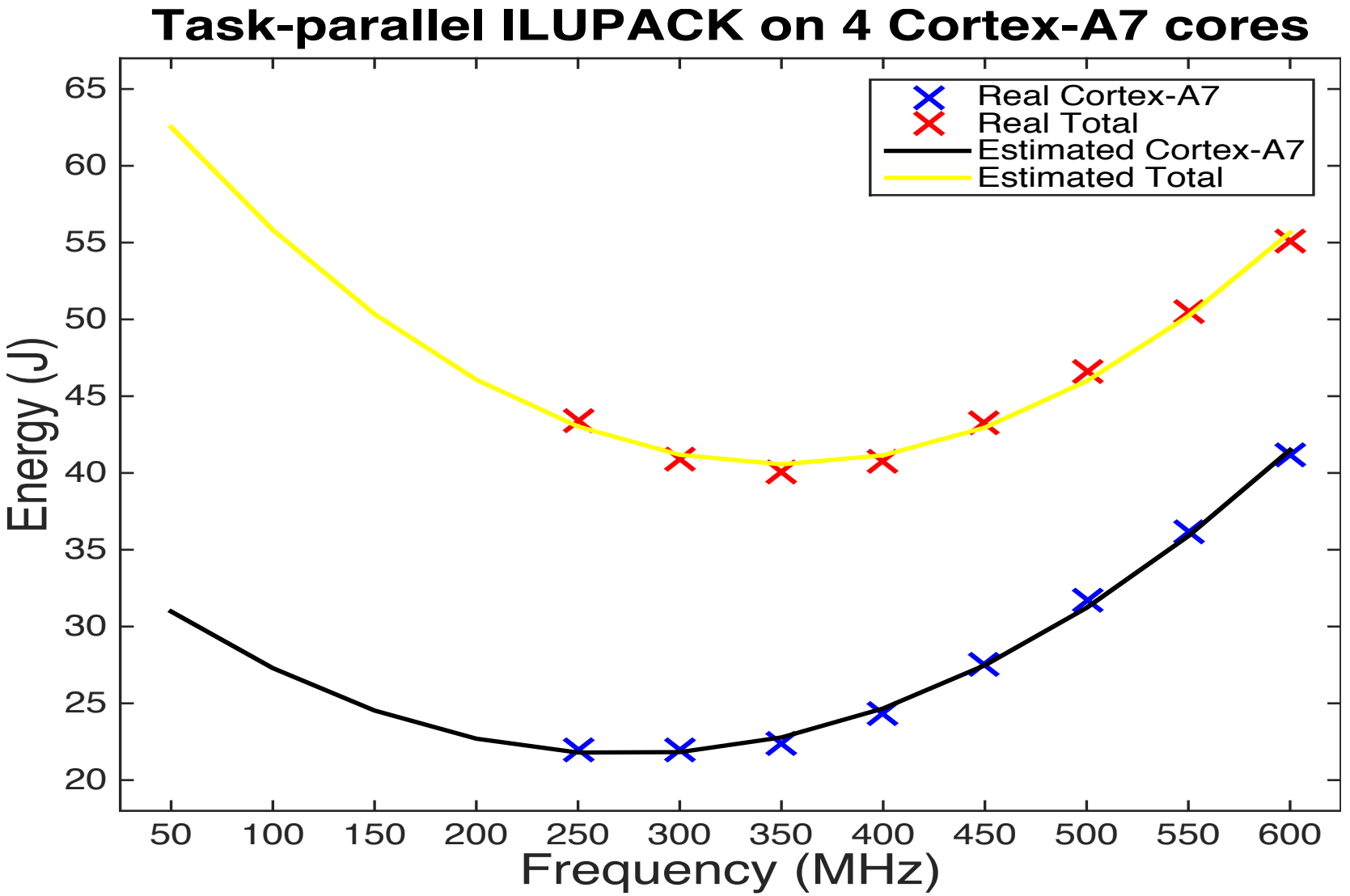
- The energy can be estimated by a quadratic polynomial:

$$E_T(f) = 2.43\text{E-}4\, f^2 - 1.70\text{E-}1\, f + 7.04\text{E+}1$$
$$1 - r2 = 7.36\text{E-}3$$

- The optimal configuration is different for total and for cluster:

$$E_T^{opt} = E_T(350) \ , \ E_{A7}^{opt} = E_{A7}(250) \Rightarrow f_{opt} = 300$$

Analysis for power and Energy Efficiency



Analysis for power and Energy Efficiency

- The relationship between power and frequency is cubic

$$P \propto V^2 * f \ , \ V \propto f \ \Rightarrow \ P \propto f^3 \ , \ P \propto V^3$$

obtained that

$$P_T(f) = 1.46\text{E-}9\, f^3 + 2.76\text{E-}7\, f^2 - 7.60\text{E-}5\, f + 2.18\text{E-}1 \ W$$
$$1 - r2 = 5.72\text{E-}4$$

- The energy can be estimated by a quadratic polynomial:

$$E_T(f) = 2.43\text{E-}4\, f^2 - 1.70\text{E-}1\, f + 7.04\text{E+}1$$
$$1 - r2 = 7.36\text{E-}3$$

- The optimal configuration is different for total and for cluster:

$$E_T^{opt} = E_T(350) \ , \ E_{A7}^{opt} = E_{A7}(250) \Rightarrow f_{opt} = 300$$

Iso-power configurations

- Not always maximum frequency (f_M) implies best performances
- The power consumption of the optimal frequency is smaller
- Iso-power determines how many systems working at f_{opt} dissipate the same power than one system working at f_M

$$P_{\mathrm{T}}(f_{\mathrm{M}})/P_{\mathrm{T}}(f_{\mathrm{opt}}) = 0.587/0.259 = 2.26\times$$
$$P_{\mathrm{A7}}(f_{\mathrm{M}})/P_{\mathrm{A7}}(f_{\mathrm{opt}}) = 0.438/0.137 = 3.19\times$$

- Given that the fraction of a system is not useful

$$C_{\mathrm{T}}(f_{\mathrm{M}}, f_{\mathrm{opt}}) = \lfloor 2.26 \rfloor = 2$$
$$C_{\mathrm{A7}}(f_{\mathrm{M}}, f_{\mathrm{opt}}) = \lfloor 3.19 \rfloor = 3$$

Iso-power configurations

- Not always maximum frequency (f_M) implies best performances
- The power consumption of the optimal frequency is smaller
- Iso-power determines how many systems working at f_{opt} dissipate the same power than one system working at f_M

$$P_{\mathrm{T}}(f_{\mathrm{M}})/P_{\mathrm{T}}(f_{\mathrm{opt}}) = 0.587/0.259 = 2.26\times$$
$$P_{\mathrm{A7}}(f_{\mathrm{M}})/P_{\mathrm{A7}}(f_{\mathrm{opt}}) = 0.438/0.137 = 3.19\times$$

- Given that the fraction of a system is not useful

$$C_{\mathrm{T}}(f_{\mathrm{M}}, f_{\mathrm{opt}}) = \lfloor 2.26 \rfloor = 2$$
$$C_{\mathrm{A7}}(f_{\mathrm{M}}, f_{\mathrm{opt}}) = \lfloor 3.19 \rfloor = 3$$

Iso-power configurations

- Not always maximum frequency (f_M) implies best performances
- The power consumption of the optimal frequency is smaller
- Iso-power determines how many systems working at f_{opt} dissipate the same power than one system working at f_M

$$P_{\mathrm{T}}(f_{\mathrm{M}})/P_{\mathrm{T}}(f_{\mathrm{opt}}) = 0.587/0.259 = 2.26\times$$
$$P_{\mathrm{A7}}(f_{\mathrm{M}})/P_{\mathrm{A7}}(f_{\mathrm{opt}}) = 0.438/0.137 = 3.19\times$$

- Given that the fraction of a system is not useful

$$C_{\mathrm{T}}(f_{\mathrm{M}}, f_{\mathrm{opt}}) = \lfloor 2.26 \rfloor = 2$$
$$C_{\mathrm{A7}}(f_{\mathrm{M}}, f_{\mathrm{opt}}) = \lfloor 3.19 \rfloor = 3$$

Iso-power configurations

- Assuming perfect scalability,
  - The execution time will be reduced

$$\bar{T}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = T(f_{\text{opt}})/2 = 159.3/2 = 79.6s$$
$$\bar{T}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = T(f_{\text{opt}})/3 = 159.3/3 = 53.1s$$

  - Where the energy consumption will be

$$\bar{E}_{\text{T}}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = \bar{T}(f_{\text{opt}}, 2) \cdot P_{\text{T}}(f_{\text{M}}) = 79.6 \cdot 0.587 = 46.7W$$
$$\bar{E}_{\text{A7}}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = \bar{T}(f_{\text{opt}}, 3) \cdot P_{\text{A7}}(f_{\text{M}}) = 53.1 \cdot 0.438 = 23.2W$$

  - This represents a gain in energy efficiency

$$E_{\text{T}}(f_{\text{M}})/\bar{E}_{\text{T}}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = 55.8/46.7 = 1.19\times$$
$$E_{\text{A7}}(f_{\text{M}})/\bar{E}_{\text{A7}}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = 41.3/23.2 = 1.77\times$$

- Too optimistic assumption is compensated with rounding down

Iso-power configurations

- Assuming perfect scalability,
  - The execution time will be reduced

$$\bar{T}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = T(f_{\text{opt}})/2 = 159.3/2 = 79.6s$$
$$\bar{T}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = T(f_{\text{opt}})/3 = 159.3/3 = 53.1s$$

  - Where the energy consumption will be

$$\bar{E}_{\text{T}}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = \bar{T}(f_{\text{opt}}, 2) \cdot P_{\text{T}}(f_{\text{M}}) = 79.6 \cdot 0.587 = 46.7W$$
$$\bar{E}_{\text{A7}}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = \bar{T}(f_{\text{opt}}, 3) \cdot P_{\text{A7}}(f_{\text{M}}) = 53.1 \cdot 0.438 = 23.2W$$

  - This represents a gain in energy efficiency

$$E_{\text{T}}(f_{\text{M}})/\bar{E}_{\text{T}}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = 55.8/46.7 = 1.19\times$$
$$E_{\text{A7}}(f_{\text{M}})/\bar{E}_{\text{A7}}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = 41.3/23.2 = 1.77\times$$

- Too optimistic assumption is compensated with rounding down

Iso-power configurations

- Assuming perfect scalability,
  - The execution time will be reduced

$$\bar{T}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = T(f_{\text{opt}})/2 = 159.3/2 = 79.6s$$
$$\bar{T}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = T(f_{\text{opt}})/3 = 159.3/3 = 53.1s$$

  - Where the energy consumption will be

$$\bar{E}_{\text{T}}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = \bar{T}(f_{\text{opt}}, 2) \cdot P_{\text{T}}(f_{\text{M}}) = 79.6 \cdot 0.587 = 46.7W$$
$$\bar{E}_{\text{A7}}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = \bar{T}(f_{\text{opt}}, 3) \cdot P_{\text{A7}}(f_{\text{M}}) = 53.1 \cdot 0.438 = 23.2W$$

  - This represents a gain in energy efficiency

$$E_{\text{T}}(f_{\text{M}})/\bar{E}_{\text{T}}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = 55.8/46.7 = 1.19\times$$
$$E_{\text{A7}}(f_{\text{M}})/\bar{E}_{\text{A7}}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = 41.3/23.2 = 1.77\times$$

- Too optimistic assumption is compensated with rounding down

Iso-power configurations

- Assuming perfect scalability,
  - The execution time will be reduced

$$\bar{T}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = T(f_{\text{opt}})/2 = 159.3/2 = 79.6s$$
$$\bar{T}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = T(f_{\text{opt}})/3 = 159.3/3 = 53.1s$$

  - Where the energy consumption will be

$$\bar{E}_{\text{T}}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = \bar{T}(f_{\text{opt}}, 2) \cdot P_{\text{T}}(f_{\text{M}}) = 79.6 \cdot 0.587 = 46.7W$$
$$\bar{E}_{\text{A7}}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = \bar{T}(f_{\text{opt}}, 3) \cdot P_{\text{A7}}(f_{\text{M}}) = 53.1 \cdot 0.438 = 23.2W$$

  - This represents a gain in energy efficiency

$$E_{\text{T}}(f_{\text{M}})/\bar{E}_{\text{T}}(f_{\text{opt}}, C_{\text{T}}(f_{\text{M}}, f_{\text{opt}})) = 55.8/46.7 = 1.19\times$$
$$E_{\text{A7}}(f_{\text{M}})/\bar{E}_{\text{A7}}(f_{\text{opt}}, C_{\text{A7}}(f_{\text{M}}, f_{\text{opt}})) = 41.3/23.2 = 1.77\times$$

- Too optimistic assumption is compensated with rounding down

Iso-power configurations

Freq. (MHz) f_i	$P(f_M)/P(f_i)$		$C_i = C(f_M, f_i) = \lfloor P(f_M)/P(f_i) \rfloor$		$\bar{T}(f_i, C_i) = T(f_i)/C(f_i)$		$\bar{E}(f_i, C_i) = \bar{T}(f_i, C_i) \cdot P(f_M)$		Gain/loss= $E(f_M)/\bar{E}(f_i, C_i)$	
	Total	A7	Total	A7	Total	A7	Total	A7	Total	A7
50	2.73	3.71	2	3	145.9	97.2	85.6	42.6	0.65	0.96
100	2.74	3.94	2	3	129.7	86.5	76.1	37.8	0.73	1.09
150	2.70	4.01	2	4	115.0	57.5	67.5	25.1	0.82	1.63
200	2.60	3.91	2	3	101.8	67.8	59.7	29.7	0.93	1.38
250	2.45	3.61	2	3	90.0	60.0	52.8	26.2	1.05	1.57
300	2.26	3.19	2	3	79.6	53.1	46.7	23.2	1.19	1.77
350	2.04	2.72	2	2	70.7	70.7	41.5	30.9	1.34	1.33
400	1.80	2.25	1	2	126.6	63.3	74.3	27.7	0.75	1.48
450	1.57	1.84	1	1	114.6	114.6	67.2	50.1	0.82	0.82
500	1.36	1.50	1	1	105.5	105.5	61.9	46.2	0.90	0.89
550	1.16	1.22	1	1	99.3	99.3	58.2	43.4	0.95	0.94
600	1.00	1.00	1	1	96.0	96.0	55.8	41.3	1.00	1.00

Outline

NTVC technology

- The key is to reduce voltage while keeping the frequency
  - Reduction of power $\rightarrow$ additional core concurrency
  - Hardware less reliable $\rightarrow$ additional mechanism
- We only consider corruptions on floating point arithmetic operations, degrading the convergence rate of the PCG
- The power are related to voltage and frequency

$$P \propto V^2 f$$

and current technology sets V proporcionally to f, for safety

$$V \propto f$$

therefore any change on V will also change f (not considered)

NTVC technology

- The key is to reduce voltage while keeping the frequency
  - Reduction of power $\rightarrow$ additional core concurrency
  - Hardware less reliable $\rightarrow$ additional mechanism
- We only consider corruptions on floating point arithmetic operations, degrading the convergence rate of the PCG
- The power are related to voltage and frequency

$$P \propto V^2 f$$

and current technology sets V proporcionally to f, for safety

$$V \propto f$$

therefore any change on V will also change f (not considered)

NTVC technology

- The key is to reduce voltage while keeping the frequency
  - Reduction of power $\rightarrow$ additional core concurrency
  - Hardware less reliable $\rightarrow$ additional mechanism
- We only consider corruptions on floating point arithmetic operations, degrading the convergence rate of the PCG
- The power are related to voltage and frequency

$$P \propto V^2 f$$

and current technology sets V proporcionally to f, for safety

$$V \propto f$$

therefore any change on V will also change f (not considered)

NTVC technology

- Given the relations between time, power and energy

$$T \propto f \ , \ P \propto V^2 f \ , \ E = T * P$$

- If no errors occurs,

$$\hat{V} = V/\sigma \ , \ \hat{f} = f \ \Rightarrow \ \hat{T} = T \ , \ \hat{P} = P/\sigma^2 \ , \ \hat{E} = E/\sigma^2$$

- If some errors occurs, there are other configurations:
  - Improve power/energy at the expense of time to solution
  - Improve time to solution/energy using additional hardware

NTVC technology

- Given the relations between time, power and energy

$$T \propto f \ , \ P \propto V^2 f \ , \ E = T * P$$

- If some errors occurs, there are other configurations:
  - Improve power/energy at the expense of time to solution

$$\hat{P} = P/s \ , \ \hat{f} = f \ \Rightarrow \ \hat{V} = V/\sqrt{s}$$
$$\hat{e} \rightarrow \hat{d} \ \Rightarrow \ \hat{T} = T * \hat{d} \ , \ \hat{E} = E * (\hat{d}/s)$$

  - Improve time to solution/energy using additional hardware

NTVC technology

- Given the relations between time, power and energy

$$T \propto f \ , \ P \propto V^2 f \ , \ E = T * P$$

- If some errors occurs, there are other configurations:
  - Improve power/energy at the expense of time to solution
  - Improve time to solution/energy using additional hardware

$$\hat{P} = P/s \ , \ \hat{T} = T \xrightarrow{iso-power} \tilde{P} = \hat{P} * s = P \ , \ \bar{T} = \hat{T}/s = T/s$$

$$\tilde{e} \rightarrow \tilde{c} \ , \ \tilde{e} \rightarrow \tilde{d} \xrightarrow{degradation} \tilde{T} = (\bar{T} + \tilde{c}) * \tilde{d} = T * \tilde{d}/s + \tilde{c} * \tilde{d}$$

$$\xrightarrow{degradation} \tilde{E} = \tilde{T} * \tilde{P} = E * (\tilde{d}/s) + \tilde{c} * \tilde{d} * P$$

NTVC technology

- Given the relations between time, power and energy

$$T \propto f \ , \ P \propto V^2 f \ , \ E = T * P$$

- If no errors occurs,

$$\hat{V} = V/\sigma \ , \ \hat{f} = f \ \Rightarrow \ \hat{T} = T \ , \ \hat{P} = P/\sigma^2 \ , \ \hat{E} = E/\sigma^2$$

- If some errors occurs, there are other configurations:
  - Improve power/energy at the expense of time to solution

$$\hat{e} \rightarrow \hat{d} \ , \hat{P} = P/s \ , \ \hat{f} = f \ \Rightarrow \ \hat{T} = T * \hat{d} \ , \ \hat{V} = V/\sqrt{s} \ , \ \hat{E} = E * (\hat{d}/s)$$

  - Improve time to solution/energy using additional hardware

$$\tilde{P} = P \ , \ \tilde{e} \rightarrow \tilde{c}, \tilde{d} \ \Rightarrow \ \tilde{T} = (\bar{T} + \tilde{c}) * \tilde{d} = T * \tilde{d}/s + \tilde{c} * \tilde{d}$$
$$\Rightarrow \tilde{E} = \tilde{T} * \tilde{P} = E * (\tilde{d}/s) + \tilde{c} * \tilde{d} * P$$

NTVC technology

- Given the relations between time, power and energy

$$T \propto f \ , \ P \propto V^2 f \ , \ E = T * P$$

- If no errors occurs,

$$\hat{V} = V/\sigma \ , \ \hat{f} = f \ \Rightarrow \ \hat{T} = T \ , \ \hat{P} = P/\sigma^2 \ , \ \hat{E} = E/\sigma^2$$

- If some errors occurs, there are other configurations:
  - Improve power/energy at the expense of time to solution

$$\hat{e} \rightarrow \hat{d} \ , \hat{P} = P/s \ , \ \hat{f} = f \ \Rightarrow \ \hat{T} = T * \hat{d} \ , \ \hat{V} = V/\sqrt{s} \ , \ \hat{E} = E * (\hat{d}/s)$$

  - Improve time to solution/energy using additional hardware

$$\tilde{P} = P \ , \ \tilde{e} \rightarrow \tilde{c}, \tilde{d} \ \Rightarrow \ \tilde{T} = (\bar{T} + \tilde{c}) * \tilde{d} = T * \tilde{d}/s + \tilde{c} * \tilde{d}$$
$$\Rightarrow \tilde{E} = \tilde{T} * \tilde{P} = E * (\tilde{d}/s) + \tilde{c} * \tilde{d} * P_c$$

Conclusions

- The results determine an optimal frequency for energy efficiency

$$E_T^{opt} = E_T(350)\ ,\ E_{A7}^{opt} = E_{A7}(250) \Rightarrow f_{opt} = 300$$

- The iso-power configurations improves the performances
  - Increase the number of units at f_{opt} to consume as $P(f_M)$
  - Reduce the execution time and improve the energy efficiency
- We have analyzed how NTVC can be exploited
  - Without errors, the reductions are directly moved to energy

$$\hat{V} = V/\sigma \ \Rightarrow\ \hat{E} = E/\sigma^2$$

  - With errors. the energy are related with the degradation

Conclusions

- The results determine an optimal frequency for energy efficiency

$$E_T^{opt} = E_T(350)\ ,\ E_{A7}^{opt} = E_{A7}(250) \Rightarrow f_{opt} = 300$$

- The iso-power configurations improves the performances
  - Increase the number of units at f_{opt} to consume as $P(f_M)$
  - Reduce the execution time and improve the energy efficiency

$$E_{\mathrm{T}}(f_{\mathrm{M}})/\bar{E}_{\mathrm{T}}(f_{\mathrm{opt}}, C_{\mathrm{T}}(f_{\mathrm{M}}, f_{\mathrm{opt}})) = 1.19\times$$
$$E_{\mathrm{A7}}(f_{\mathrm{M}})/\bar{E}_{\mathrm{A7}}(f_{\mathrm{opt}}, C_{\mathrm{A7}}(f_{\mathrm{M}}, f_{\mathrm{opt}})) = 1.77\times$$

- We have analyzed how NTVC can be exploited
  - Without errors, the reductions are directly moved to energy

$$\hat{V} = V/\sigma \ \Rightarrow\ \hat{E} = E/\sigma^2$$

  - With errors. the energy are related with the degradation

Conclusions

- The results determine an optimal frequency for energy efficiency

$$E_T^{opt} = E_T(350)\ ,\ E_{A7}^{opt} = E_{A7}(250) \Rightarrow f_{opt} = 300$$

- The iso-power configurations improves the performances
  - Increase the number of units at f_{opt} to consume as $P(f_M)$
  - Reduce the execution time and improve the energy efficiency

$$E_{\mathrm{T}}(f_{\mathrm{M}})/\bar{E}_{\mathrm{T}}(f_{\mathrm{opt}},\dots) = 1.19\times\ ,\ E_{\mathrm{A7}}(f_{\mathrm{M}})/\bar{E}_{\mathrm{A7}}(f_{\mathrm{opt}},\dots) = 1.77\times$$

- We have analyzed how NTVC can be exploited
  - Without errors, the reductions are directly moved to energy

$$\hat{V} = V/\sigma \ \Rightarrow\ \hat{E} = E/\sigma^2$$

  - With errors. the energy are related with the degradation

$$\hat{P} = P/s \ \Rightarrow \hat{E} = E * (\hat{d}/s)$$

$$\tilde{e} \rightarrow \tilde{c}, \tilde{d} \ \Rightarrow \tilde{E} = E * \tilde{d}/s + \tilde{c} * \tilde{d} * P_c$$

Thanks for your attention !

Questions ?