High-performance and automatic computing

Paolo Bientinesi

AICES, RWTH Aachen
pauldj@aices.rwth-aachen.de

October 22nd, 2013
Goethe Universität Frankfurt am Main

Overview

- **High-performance computing**
  numerical computations, parallel architectures
  $\rightarrow$ time-to-solution, efficiency, scalability, ...

Overview

- **High-performance computing**
  numerical computations, parallel architectures
  $\rightarrow$ time-to-solution, efficiency, scalability, $\ldots$

- **Automatic computing**
  optimization, search, but also derivation & deduction
  $\rightarrow$ range of algorithms, productivity

Overview

- **High-performance computing**
  numerical computations, parallel architectures
  $\rightarrow$ time-to-solution, efficiency, scalability, ...

- **Automatic computing**
  optimization, search, but also derivation & deduction
  $\rightarrow$ range of algorithms, productivity

- **Applications**
  $\rightarrow$ application-specific properties & needs, large scale, full code

methods & applications

- P. Bientinesi
- E. Di Napoli — Electronic structure calculations
- M. Petschow — Parallel eigensolvers
- D. Fabregat — Automation, Computational biology
- D. Tameling* — Molecular dynamics
- E. Peise — Performance modeling & prediction
- L. Beyer — Density Functional Theory
- F. Kürten, Y. Madzhunkov, A. Frank, P. Springer, . . .

methods & applications

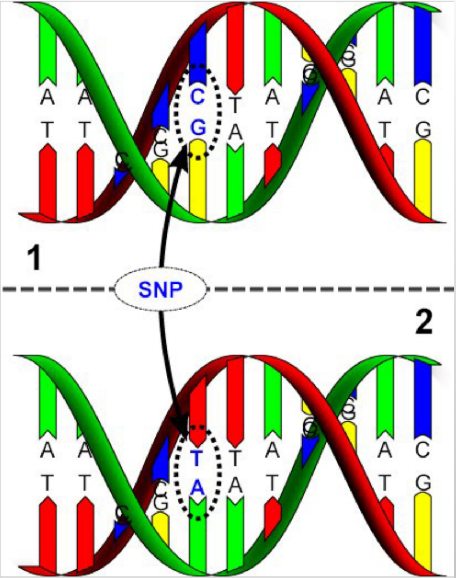
- P. Bientinesi
- E. Di Napoli — Electronic structure calculations
- M. Petschow — Parallel eigensolvers
- D. Fabregat — Automation, Computational biology
- D. Tameling* — Molecular dynamics
- E. Peise — Performance modeling & prediction
- L. Beyer — Density Functional Theory
- F. Kürten, Y. Madzhunkov, A. Frank, P. Springer, ...

D. Fabregat

E. Peise

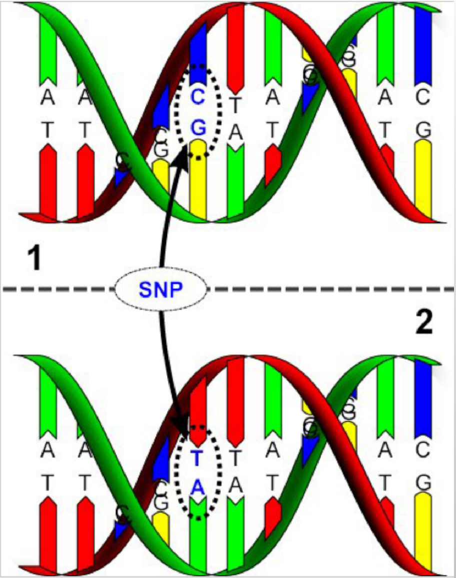
Y. Aulchenko

Genome-Wide Association Studies



Source: David Hall

Genome-Wide Association Studies



Source: David Hall

Yurii

Paolo

Genome-Wide Association Studies

Yurii	Paolo
"Mixed models"	???

Genome-Wide Association Studies

Yurii	Paolo
"Mixed models"	???
Linear regression with non-independent outcomes	???

Genome-Wide Association Studies

Yurii	Paolo
"Mixed models"	???
Linear regression with non-independent outcomes	???
Generalized least-square problems	...

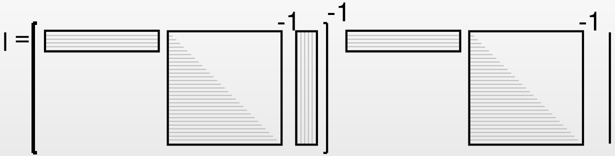
Genome-Wide Association Studies

Yurii	Paolo
"Mixed models"	???
Linear regression with non-independent outcomes	???
Generalized least-square problems	...

$$b := \left(X^T M^{-1} X\right)^{-1} X^T M^{-1} y$$

- Inputs: $M \in \mathbb{R}^{n \times n}, \quad X \in \mathbb{R}^{n \times p}, \quad y \in \mathbb{R}^n$
- Output: $b \in \mathbb{R}^p$

$\star$**To be repeated millions of times**$\star$



Genome-wide association analysis

- y: **phenotype** (outcome; vector of observations)

  E.g.: height, blood pressure for a set of people

- X: **genome measurements** and covariates (design matrix; predictors)

  E.g.: sex and age over height

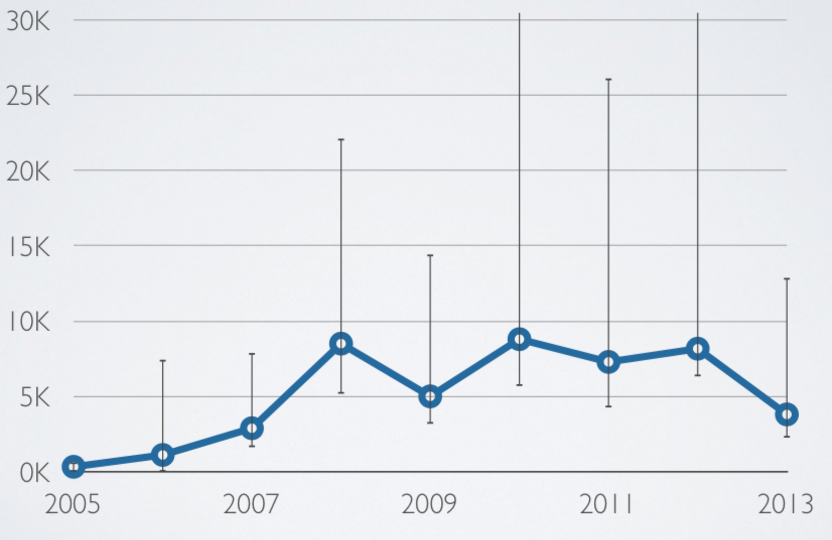
- M: **dependencies** between observations

  E.g.: tall parents have tall children

- b: **relation** between a variation in the outcome (y) and a variation in the genome sequence (X)

Stats

n: Population size



Stats

m: Number of SNPs

Problem definition (1)

$$b := \left(X^T M^{-1} X\right)^{-1} X^T M^{-1} y$$

"to be repeated millions of times"

Problem definition (1)

$$b := \left(X^T M^{-1} X\right)^{-1} X^T M^{-1} y$$

"to be repeated millions of times"

$$\Downarrow$$

$$b_i := \left(X_i^T M_i^{-1} X_i\right)^{-1} X_i^T M_i^{-1} y_i$$

for $i = 1, \dots, m$

Problem definition (1)

$$b := \left(X^T M^{-1} X\right)^{-1} X^T M^{-1} y$$

"to be repeated millions of times"

$$\Downarrow$$

$$b_i := \left(X_i^T M_i^{-1} X_i\right)^{-1} X_i^T M_i^{-1} y_i$$

for $i = 1, \ldots, m$

$$\Downarrow$$

Problem size

$M_i \in \mathbb{R}^{n \times n}$	$1000 \leq n \leq 20k$+	7.5MBs – 3GBs
$X_i \in \mathbb{R}^{n \times p}$	$3 \leq p \leq 20$	30 – 625KBs
$y_j \in \mathbb{R}^n$		8 – 780KBs
$b_i \in \mathbb{R}^p$		24 – 160 Bytes
Total	$10^6 \leq m \leq 10^8$	7.5 – 3000 TBs

Problem definition (2)

$$b_i := \left(X_i^T M_i^{-1} X_i\right)^{-1} X_i^T M_i^{-1} y_i$$

Problem definition (2)

$$b_i := \left(X_i^T M_i^{-1} X_i\right)^{-1} X_i^T M_i^{-1} y_i$$

$$\Downarrow$$

$$b_i := \left(X_i^T M^{-1} X_i\right)^{-1} X_i^T M^{-1} y$$

and $X_i = [X_L | X_{Ri}]$,

for $i = 1, \dots, m$

$$\Downarrow$$

Problem size

$M \in \mathbb{R}^{n \times n}$	$1000 \leq n \leq 100k$	7.5MBs – 74.5GBs
$X_{Ri} \in \mathbb{R}^{n}$		8 – 780KBs
$b_i \in \mathbb{R}^{p}$		24 – 160 Bytes
Total	$10^6 \leq m \leq 10^8$	74GBs – 7 TBs

Problem definition (3)

$$b_i := \left(X_i^T M^{-1} X_i\right)^{-1} X_i^T M^{-1} y$$

Problem definition (3)

$$b_i := \left(X_i^T M^{-1} X_i\right)^{-1} X_i^T M^{-1} y$$

$$\Downarrow$$

$$b_{ij} := \left(X_i^T M_j^{-1} X_i\right)^{-1} X_i^T M_j^{-1} y_j$$

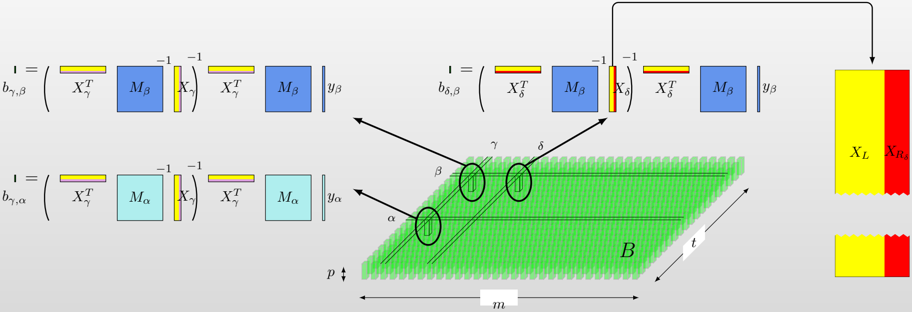
and

$M_j = \sigma_j(\Phi + h_j I)$,

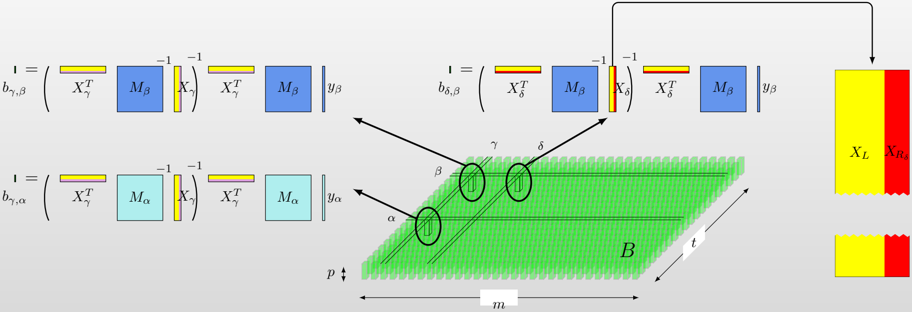
for $i = 1, \ldots, m$ and $j = 1, \ldots, t$

Moreover, either $t = 1$ or $t \leq 10^5$.

GWAS: complete problem definition



GWAS: complete problem definition



Problem size

$M \in \mathbb{R}^{n\times n}$	$1000 \le n \le 100k$	7.5MBs – 74.5GBs
$X_{Ri}, y_j \in \mathbb{R}^n$		8 – 780KBs
$b_{ij} \in \mathbb{R}^p$	$3 \le p \le 20$	24 – 160 Bytes
Total	$m \le 10^8,\ t \le 10^5$	1.5 – 100s TBs

Vision: domain-specific compiler

$\alpha, \beta, \gamma \in \mathbb{R}$

$$\mu := \left(\gamma * \alpha^{-1} * \gamma\right)^{-1} * \beta$$

$$\Downarrow$$

compiler

$$\Downarrow$$

code for $\mu := \dfrac{\alpha\beta}{\gamma^2}$

Vision: domain-specific compiler

$\alpha, \beta, \gamma \in \mathbb{R}$

$$\mu := \left(\gamma * \alpha^{-1} * \gamma\right)^{-1} * \beta$$

$$\Downarrow$$

compiler

$$\Downarrow$$

code for $\mu := \dfrac{\alpha\beta}{\gamma^2}$

Nov. 1954: *The IBM Mathematical FORmula TRANslating System, FORTRAN*

`“a set of programs to accept a concise formulation of a problem and to produce automatically a solution of the problem”`

Automatic generation

$$Y \in \mathbb{R}^{n \times p}, \quad A \in \mathbb{R}^{n \times n}, \quad b \in \mathbb{R}^n$$

$$v := (Y^T * A^{-1} * Y)^{-1} * b$$

$$\Downarrow$$

$<$ **lin. alg. compiler** $>$

$$\Downarrow$$

algorithm, code

Automatic generation

$$Y \in \mathbb{R}^{n \times p}, \quad A \in \mathbb{R}^{n \times n}, \quad b \in \mathbb{R}^{n}$$

$$v := (Y^T * A^{-1} * Y)^{-1} * b$$

$$\Downarrow$$

$<$ **lin. alg. compiler** $>$

$$\Downarrow$$

algorithm, code

Matrix algebra

Sequences of operations

Inference of properties

Cost analysis

Building blocks

Code generation

Algorithms generated

Algorithm 1	Algorithm 2	...	Algorithm 20	...
$LL^T = M$	$LL^T = M$		$ZWZ^T = \Phi$	
$X := L^{-1}X$	$X := L^{-1}X$		$D := (hW + (1-h)I)^{-1}$	
$S := X^TX$	$QR := X$		$KK^T = D$	
$GG^T = S$	$y := L^{-1}y$		$X := Z^TX$	
$y := L^{-1}y$	$b := Q^Ty$		$X := K^TX$	
$b := X^Ty$	$b := R^{-1}b$		$QR := X$	
$b := G^{-1}b$			$y := L^{-1}y$	
$b := G^{-T}b$			$b := Q^Ty$	
			$b := R^{-1}b$	

Many algorithms! Predictions?

Many algorithms! Predictions?

Flop count – rough estimate

	Alg. 1	Alg. 2	Alg. 20
Single instance ($t = 1$)	$\mathbf{O(n^3)}$	$O(n^3)$	$O(n^3)$
2D sequence ($t \gg 1$)	$O(tn^3 + mtn^2)$	$O(tn^3 + mtn^2)$	$\mathbf{O(n^3 + mtn)}$

Many algorithms! Predictions?

Flop count – rough estimate

	Alg. 1	Alg. 2	Alg. 20
Single instance ($t = 1$)	$\mathbf{O(n^3)}$	$O(n^3)$	$O(n^3)$
2D sequence ($t \gg 1$)	$O(tn^3 + mtn^2)$	$O(tn^3 + mtn^2)$	$\mathbf{O(n^3 + mtn)}$

Analytic models

Roman Iakymchuk

Model-based prediction

Elmar Peise

Algorithm → implementations

operands

X	input	100s GBs – 2 TBs	streaming from disk
y	input	1 – 10 GBs	streaming from disk
M	input	MBs – 80 GBs	read once
b	output	100s MBs or 10s TBs	streaming to disk

Algorithm $\rightarrow$ implementations

operands

X	input	100s GBs – 2 TBs	streaming from disk
y	input	1 – 10 GBs	streaming from disk
M	input	MBs – 80 GBs	read once
b	output	100s MBs or 10s TBs	streaming to disk

Does M fit in memory?

Algorithm $\rightarrow$ implementations

operands

X	input	100s GBs – 2 TBs	streaming from disk
y	input	1 – 10 GBs	streaming from disk
M	input	MBs – 80 GBs	read once
b	output	100s MBs or 10s TBs	streaming to disk

Does M fit in memory?

- YES $\Rightarrow$ single node + multithreading
  streaming HD$\leftrightarrow$CPU, double buffering, in-core implementation

Algorithm $\rightarrow$ implementations

operands

X	input	100s GBs – 2 TBs	streaming from disk
y	input	1 – 10 GBs	streaming from disk
M	input	MBs – 80 GBs	read once
b	output	100s MBs or 10s TBs	streaming to disk

Does M fit in memory?

- YES $\Rightarrow$ single node + multithreading
  streaming HD$\leftrightarrow$CPU, double buffering, in-core implementation

  Does M fit in GPU-memory?

  - Yes $\Rightarrow$ accelerator
    streaming HD$\leftrightarrow$CPU$\leftrightarrow$GPU, triple+double buffering, GPU implementation

Algorithm $\rightarrow$ implementations

operands

X	input	100s GBs – 2 TBs	streaming from disk
y	input	1 – 10 GBs	streaming from disk
M	input	MBs – 80 GBs	read once
b	output	100s MBs or 10s TBs	streaming to disk

Does M fit in memory?

- YES $\Rightarrow$ single node + multithreading
  streaming HD$\leftrightarrow$CPU, double buffering, in-core implementation

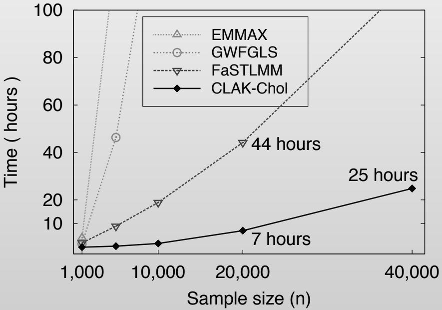
  Does M fit in GPU-memory?

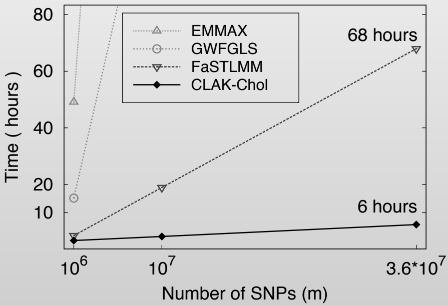
  - Yes $\Rightarrow$ accelerator
    streaming HD$\leftrightarrow$CPU$\leftrightarrow$GPU, triple+double buffering, GPU implementation

- NO $\Rightarrow$ distributed memory + MPI
  partitioning + streaming HD$\leftrightarrow$CPUs, double buffering, data distribution

Results $t = 1$

Single-trait analysis

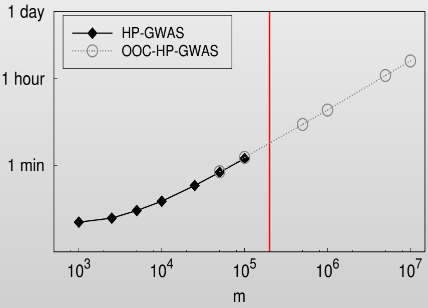




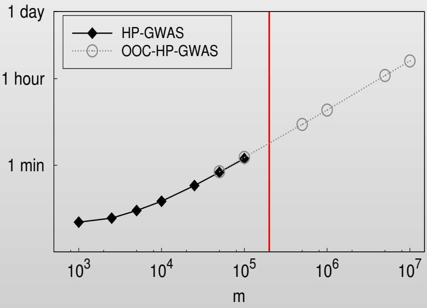
Results $t = 1$, large-scale

Single node

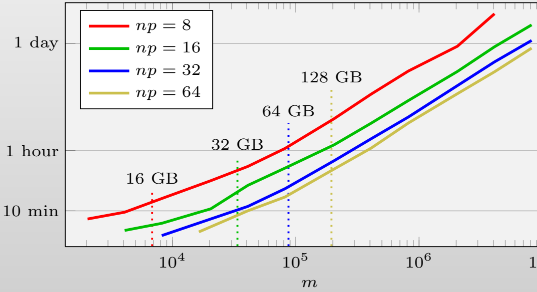
MPI



Single node

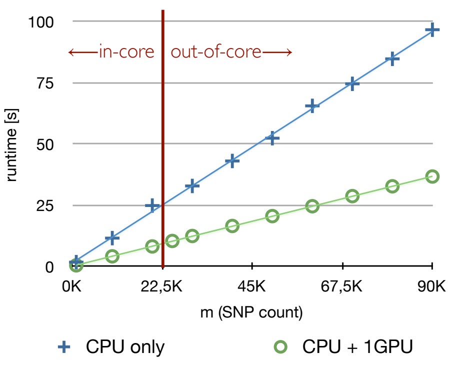


MPI

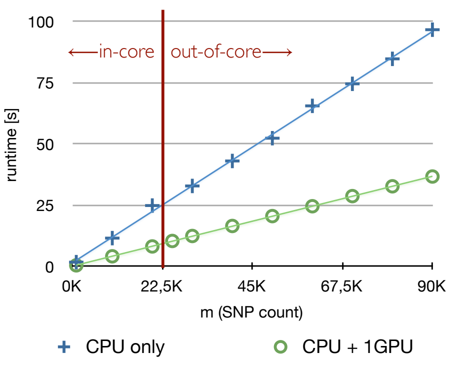


1 GPU

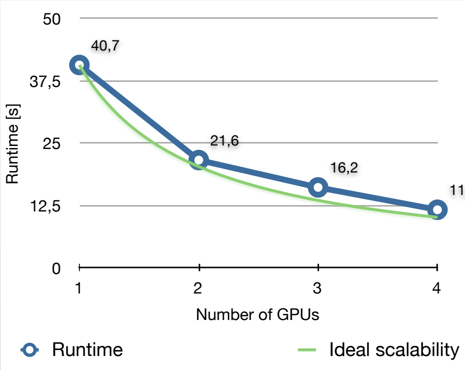
Scalability



1 GPU

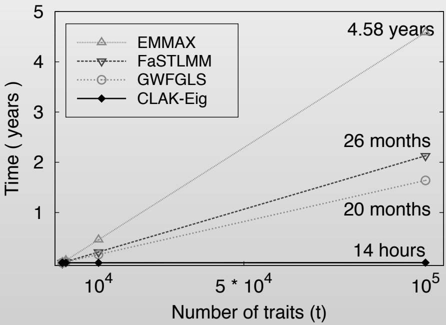


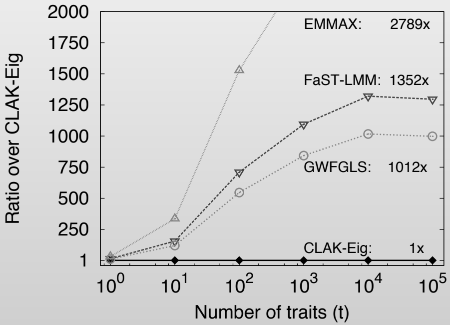
Scalability



Results $t \gg 1$

Multi-trait analysis, "OMICS"-data





Final comments

HPC's perspective

- In-core efficiency
- How to sustain efficiency?

Final comments

HPC's perspective

- In-core efficiency
- How to sustain efficiency?

App's perspective

- Many data formats
- Missing data, bogus data
- Output data. Post-processing?
- New features

Final comments

HPC's perspective

- In-core efficiency
- How to sustain efficiency?

App's perspective

- Many data formats
- Missing data, bogus data
- Output data. Post-processing?
- New features

- HUGE gap:
  algorithm $\leftrightarrow$ optimized implementation
  (data management, parallelism)

Final comments

HPC's perspective

- In-core efficiency
- How to sustain efficiency?

App's perspective

- Many data formats
- Missing data, bogus data
- Output data. Post-processing?
- New features

- HUGE gap:
  algorithm $\leftrightarrow$ optimized implementation
  (data management, parallelism)
- Development cycle: several months!

Final comments

HPC's perspective

- In-core efficiency
- How to sustain efficiency?

App's perspective

- Many data formats
- Missing data, bogus data
- Output data. Post-processing?
- New features

- HUGE gap:
  algorithm $\leftrightarrow$ optimized implementation
  (data management, parallelism)
- Development cycle: several months!
- How to deal with BIG problems?
  Expose knowledge, exploit knowledge