Genome-Wide Association Studies (GWAS)

With D. Fabregat, E. Peise, L. Beyer, A. Frank, and Y. Aulchenko

Paolo Bientinesi

pauldj@cs.umu.se

DCSE Summerschool

Numerical Linear Algebra on High Performance Computers

June 06, 2023

TU Delft

UMEÅ UNIVERSITY

Context: Linear Algebra Expressions

Signal Processing	$x := \left(A^{-T}B^TBA^{-1} + R^TLR\right)^{-1} A^{-T}B^TBA^{-1}y$	$R \in \mathbb{R}^{n-1\times n}$, UT; $L \in \mathbb{R}^{n-1\times n-1}$, DI
Kalman Filter	$K_k := P_k^bH^T(HP_k^bH^T + R)^{-1};\ x_k^a := x_k^b + K_k(z_k - Hx_k^b);\ P_k^a := (I - K_KH)\,P_k^b$	
Ensemble Kalman Filter	$X^a := X^b + \left(B^{-1} + H^TR^{-1}H\right)^{-1}\left(Y - HX^b\right)$	$B \in \mathbb{R}^{N\times N}$ SSPD; $R \in \mathbb{R}^{m\times m}$, SSPD
Ensemble Kalman Filter	$\delta X := \left(B^{-1} + H^TR^{-1}H\right)^{-1} H^TR^{-1}\left(Y - HX^b\right)$	
Ensemble Kalman Filter	$\delta X := XV^T\left(R + HX(HX)^T\right)^{-1}\left(Y - HX^b\right)$	
Image Restoration	$x_k := (H^TH + \lambda\sigma^2I_n)^{-1}(H^Ty + \lambda\sigma^2(v_{k-1} - u_{k-1}))$	
Image Restoration	$H^\dagger := H^T(HH^T)^{-1};\ y_k := H^\dagger y + (I_n - H^\dagger H)x_k$	
Rand. Matrix Inversion	$X_{k+1} := S(S^TAS)^{-1}S^T + (I_n - S(S^TAS)^{-1}S^TA)X_k(I_n - AS(S^TAS)^{-1}S^T)$	
Rand. Matrix Inversion	$X_{k+1} := X_k + WA^TS(S^TAWA^TS)^{-1}S^T(I_n - AX_k)$	$W \in \mathbb{R}^{n\times n}$, SPD
Rand. Matrix Inversion	$X_{k+1} := X_k + (I_n - X_kA^T)S(S^TA^TWAS)^{-1}S^TA^TW$	
Rand. Matrix Inversion	$\Lambda := S(S^TAWAS)^{-1}S^T;\ \Theta := \Lambda AW;\ M_k := X_kA - I$ $X_{k+1} := X_k - M_k\Theta - (M_k\Theta)^T + \Theta^T(AX_kA - A)\Theta$	

Generalized Least Squares	$b := (X^T M^{-1} X)^{-1} X^T M^{-1} y$	$n > m$; $M \in \mathbb{R}^{n \times n}$, SPD; $X \in \mathbb{R}^{n \times m}$; $y \in \mathbb{R}^{n \times 1}$
Stochastic Newton	$B_k := \frac{k}{k-1} B_{k-1}(I_n - A^T W_k((k-1)I_l + W_k^T A B_{k-1} A^T W_k)^{-1} W_k^T A B_{k-1})$	
Optimization	$x_f := W A^T (A W A^T)^{-1}(b - Ax);\;\; x_o := W(A^T(AWA^T)^{-1} Ax - c)$	
Optimization	$x := W(A^T(AWA^T)^{-1} b - c)$	
Triangular Matrix Inv.	$X_{10} := L_{10} L_{00}^{-1};\;\; X_{20} := L_{20} + L_{22}^{-1} L_{21} L_{11}^{-1} L_{10};\;\; X_{11} := L_{11}^{-1};\;\; X_{21} := -L_{22}^{-1} L_{21}$	
Tikhonov Regularization	$x := (A^T A + \Gamma^T \Gamma)^{-1} A^T b$	$A \in \mathbb{R}^{n \times m}$; $\Gamma \in \mathbb{R}^{m \times m}$; $b \in \mathbb{R}^{n \times 1}$
Tikhonov Regularization	$x := (A^T A + \alpha^2 I)^{-1} A^T b$	
Gen. Tikhonov Reg.	$x := (A^T P A + Q)^{-1}(A^T P b + Q x_0)$	$P \in \mathbb{R}^{n \times n}$, SSPD; $Q \in \mathbb{R}^{m \times m}$, SSPD; $x_0 \in \mathbb{R}^{m \times 1}$
Gen. Tikhonov reg.	$x := x_0 + (A^T P A + Q)^{-1}(A^T P(b - A x_0))$	
LMMSE estimator	$K_{t+1} := C_t A^T (A C_t A^T + C_z)^{-1};\; x_{t+1} := x_t + K_{t+1}(y - A x_t);\; C_{t+1} := (I - K_{t+1} A) C_t$	
LMMSE estimator	$x_{\text{out}} = C_X A^T (A C_X A^T + C_Z)^{-1}(y - Ax) + x$	
LMMSE estimator	$x_{\text{out}} := (A^T C_Z^{-1} A + C_X^{-1})^{-1} A^T C_Z^{-1}(y - Ax) + x$	

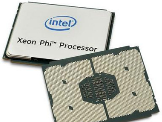
$$K_k := P_k^b H^T (H P_k^b H^T + R)^{-1}; \quad x_k^a := x_k^b + K_k(z_k - H x_k^b); \quad P_k^a := (I - K_K H)\, P_k^b$$

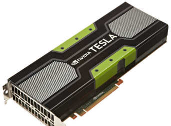
$$\begin{cases} C_\dagger := PCP^T + Q \\ K := C_\dagger H^T (H C_\dagger H^T)^{-1} \end{cases}$$

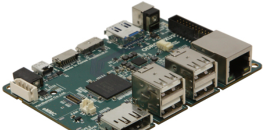
$$\Lambda := S(S^T AWAS)^{-1} S^T; \; \Theta := \Lambda AW; \; M_k := X_k A - I$$
$$X_{k+1} := X_k - M_k \Theta - (M_k \Theta)^T + \Theta^T (A X_k A - A)\Theta$$

$$x := A(B^T B + A^T R^T \Lambda R A)^{-1} B^T B A^{-1} y \qquad \ldots \qquad E := Q^{-1} U (I + U^T Q^{-1} U)^{-1} U^T$$

?







$$K_k := P_k^b H^T (H P_k^b H^T + R)^{-1}; \quad x_k^a := x_k^b + K_k (z_k - H x_k^b); \quad P_k^a := (I - K_K H) P_k^b$$

$$\begin{cases} C_\dagger := P C P^T + Q \\ K := C_\dagger H^T (H C_\dagger H^T)^{-1} \end{cases}$$

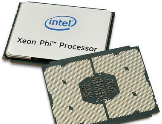
$$\Lambda := S(S^T A W A S)^{-1} S^T; \ \Theta := \Lambda A W; \ M_k := X_k A - I$$
$$X_{k+1} := X_k - M_k \Theta - (M_k \Theta)^T + \Theta^T (A X_k A - A) \Theta$$

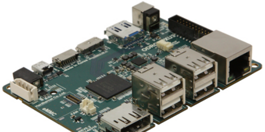
$$x := A(B^T B + A^T R^T \Lambda R A)^{-1} B^T B A^{-1} y \quad \ldots \quad E := Q^{-1} U (I + U^T Q^{-1} U)^{-1} U^T$$

$\boxed{y := \alpha x + y}$ $\boxed{LU = A}$ $\cdots$ $\boxed{C := \alpha A B + \beta C}$

$\boxed{X := A^{-1} B}$ $\boxed{C := A B^T + B A^T + C}$ $\boxed{X := L^{-1} M L^{-T}}$ $\boxed{QR = A}$

... **BLAS** **LAPACK** ...





$$K_k := P_k^b H^T (H P_k^b H^T + R)^{-1}; \quad x_k^a := x_k^b + K_k(z_k - H x_k^b); \quad P_k^a := (I - K_K H)\, P_k^b$$

$$\begin{cases} C_\dagger := PCP^T + Q \\ K := C_\dagger H^T (H C_\dagger H^T)^{-1} \end{cases}$$

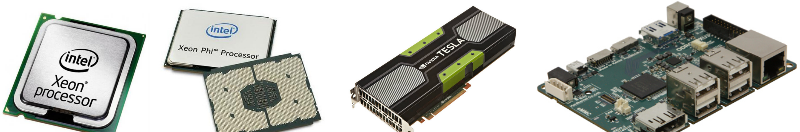
$$\Lambda := S(S^T AWAS)^{-1} S^T; \; \Theta := \Lambda AW; \; M_k := X_k A - I$$
$$X_{k+1} := X_k - M_k \Theta - (M_k \Theta)^T + \Theta^T (A X_k A - A)\Theta$$

$$x := A(B^T B + A^T R^T \Lambda R A)^{-1} B^T B A^{-1} y \qquad \ldots \qquad E := Q^{-1} U (I + U^T Q^{-1} U)^{-1} U^T$$

$\boxed{y := \alpha x + y}$ $\boxed{LU = A}$ $\cdots$ $\boxed{C := \alpha AB + \beta C}$

$\boxed{X := A^{-1} B}$ $\boxed{C := AB^T + BA^T + C}$ $\boxed{X := L^{-1} M L^{-T}}$ $\boxed{QR = A}$

... **BLAS** **LAPACK** ...



$$K_k := P_k^b H^T (H P_k^b H^T + R)^{-1}; \quad x_k^a := x_k^b + K_k(z_k - H x_k^b); \quad P_k^a := (I - K_K H)\, P_k^b$$

$$\begin{cases} C_\dagger := PCP^T + Q \\ K := C_\dagger H^T (H C_\dagger H^T)^{-1} \end{cases}$$

$$\Lambda := S(S^T AWAS)^{-1} S^T; \ \Theta := \Lambda AW; \ M_k := X_k A - I$$
$$X_{k+1} := X_k - M_k\Theta - (M_k\Theta)^T + \Theta^T(AX_kA - A)\Theta$$

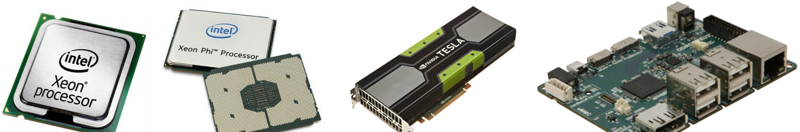
$$x := A(B^T B + A^T R^T \Lambda R \ldots$$ $$\ldots{}^{-1} U(I + U^T Q^{-1} U)^{-1} U^T$$

LINEAR ALGEBRA MAPPING PROBLEM

$\boxed{y := \alpha x + y}$ $\boxed{LU = A}$ $\cdots$ $\boxed{C := \alpha AB + \beta C}$

$\boxed{X := A^{-1}B}$ $\boxed{C := AB^T + BA^T + C}$ $\boxed{X := L^{-1}ML^{-T}}$ $\boxed{QR = A}$

... **BLAS** **LAPACK** ...



$$K_k := P_k^b H^T (H P_k^b H^T + R)^{-1}; \quad x_k^a := x_k^b + K_k(z_k - H x_k^b); \quad P_k^a := (I - K_K H)\, P_k^b$$

$$\begin{cases} C_\dagger := P C P^T + Q \\ K := C_\dagger H^T (H C_\dagger H^T)^{-1} \end{cases}$$

$$\Lambda := S(S^T A W A S)^{-1} S^T; \ \Theta := \Lambda A W; \ M_k := X_k A - I$$
$$X_{k+1} := X_k - M_k \Theta - (M_k \Theta)^T + \Theta^T (A X_k A - A)\Theta$$

$x := A(B^T B + A^T R^T \Lambda$ … $^1 U(I + U^T Q^{-1} U)^{-1} U^T$

LINEAR ALGEBRA MAPPING PROBLEM

$\boxed{y := \alpha x + y}$ $\boxed{LU = A}$ $\cdots$ $\boxed{C := \alpha A B + \beta C}$

$\boxed{X := A^{-1} B}$ $\boxed{C := A B^T + B A^T + C}$ $\boxed{X := L^{-1} M L^{-T}}$ $\boxed{QR = A}$

C. Psarras, H. Barthels, P. Bientinesi, [arXiv:1911.09421]
"The Linear Algebra Mapping Problem. Current state of linear algebra languages and libraries", ACM TOMS, 2022

A. Sankaran, N.A. Alashti, C. Psarras, P. Bientinesi, [arXiv:2202.09888]
"Benchmarking the Linear Algebra Awareness of TensorFlow and PyTorch", iWAPT-22

H. Barthels, C. Psarras, P. Bientinesi, [arXiv:1912.12924]
"Linnea: Automatic Generation of Efficient Linear Algebra Programs", ACM TOMS, 2021

Linear Algebra Mapping Problem (“LAMP”)

Linear Algebra Mapping Problem ("LAMP")

- $\mathcal{E}$: a sequence of explicit assignments $var_i := EXP_i$
- $\mathcal{K}$: a set of available computational building blocks e.g., BLAS, LAPACK, ...
- $\mathcal{M}$: a cost function defined over $\mathcal{K}^+$ #FLOPs, exec. time, #mem.ops, stability

Linear Algebra Mapping Problem ("LAMP")

- $\mathcal{E}$: a sequence of explicit assignments $var_i := EXP_i$
- $\mathcal{K}$: a set of available computational building blocks e.g., BLAS, LAPACK, ...
- $\mathcal{M}$: a cost function defined over $\mathcal{K}^+$ #FLOPs, exec. time, #mem.ops, stability

LAMP:

Find a sequence of calls to building blocks in $\mathcal{K}$, optimal according to $\mathcal{M}$, that computes all the assignments in $\mathcal{E}$.

- Suboptimal solution $\rightarrow$ easy
- Optimality $\rightarrow$ NP complete $\leftarrow$ reduction from Ensemble Computation

Example

$$\boxed{b := (X^T X)^{-1} X^T y}$$

Example

$$\boxed{b := (X^T X)^{-1} X^T y}$$

$$M := X^T X$$

$$\boxed{b := M^{-1} X^T y}$$

Example

$$b := (X^TX)^{-1}X^Ty$$

$$M := X^TX$$

$$b := M^{-1}X^Ty$$

$$b := M^{-1}(X^Ty)$$

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

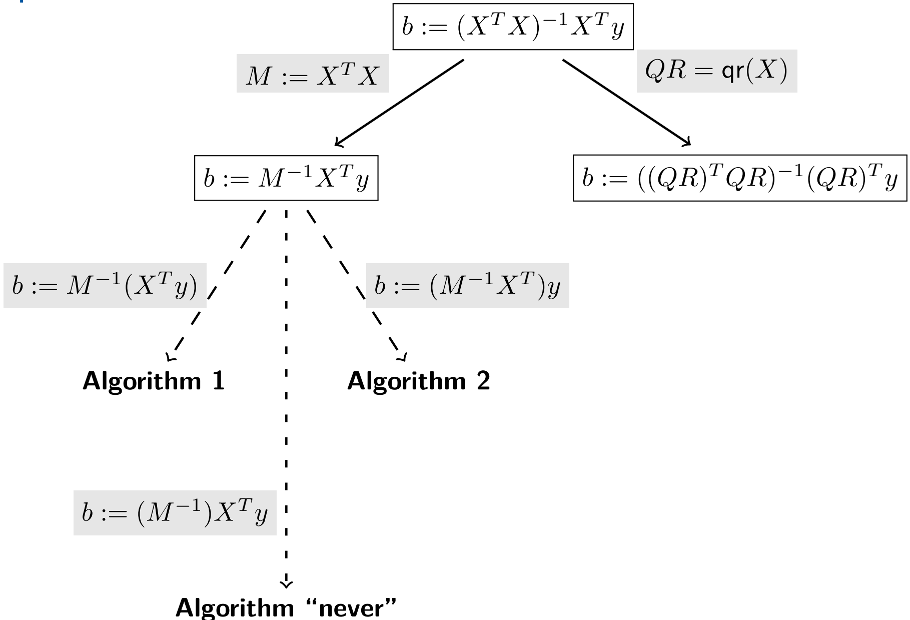
Algorithm 1

Algorithm 2

$$b := (M^{-1})X^Ty$$

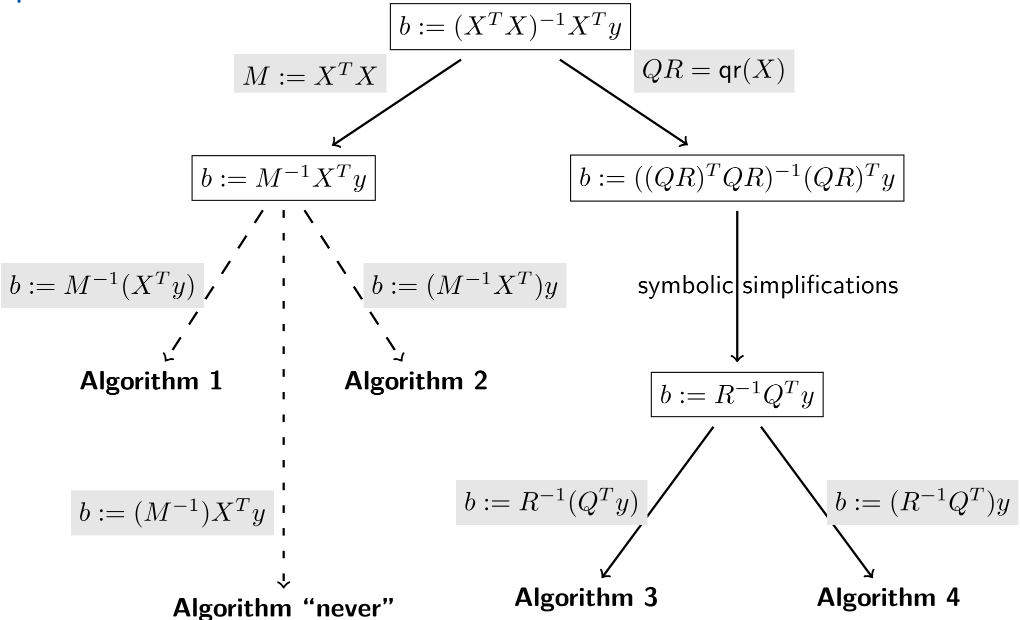
Algorithm “never”

Example

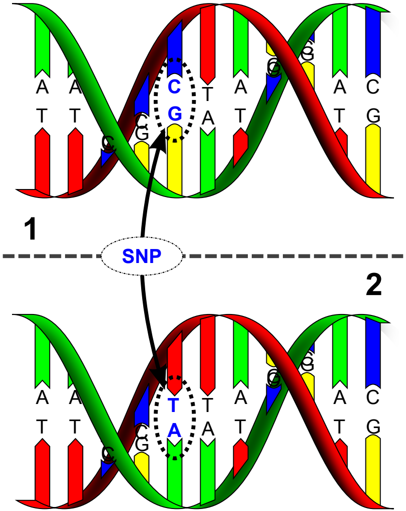


$b := (X^T X)^{-1} X^T y$
$M := X^T X$
$QR = \mathsf{qr}(X)$
$b := M^{-1} X^T y$
$b := ((QR)^T QR)^{-1} (QR)^T y$
$b := M^{-1}(X^T y)$
$b := (M^{-1} X^T) y$
Algorithm 1
Algorithm 2
$b := (M^{-1}) X^T y$
Algorithm “never”

Example


$b := (X^T X)^{-1} X^T y$
$M := X^T X$
$QR = \mathsf{qr}(X)$
$b := M^{-1} X^T y$
$b := ((QR)^T QR)^{-1} (QR)^T y$
$b := M^{-1}(X^T y)$
$b := (M^{-1} X^T) y$
symbolic simplifications
Algorithm 1
Algorithm 2
$b := R^{-1} Q^T y$
$b := (M^{-1}) X^T y$
$b := R^{-1}(Q^T y)$
$b := (R^{-1} Q^T) y$
Algorithm 3
Algorithm 4
Algorithm "never"

GWAS



Source: David Hall

Correlation between a difference in the genome sequence (“SNP”) and a difference in the phenotype (“traits”, observations)

Yurii A. 

 Paolo

Yurii A.    Paolo

“Mixed models” ???

Yurii A.    Paolo

"Mixed models" ???

Linear regression with non-independent outcomes ???

Yurii A.    Paolo

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

Yurii A.

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: $M \in \mathbb{R}^{n \times n}$, $X \in \mathbb{R}^{n \times p}$, $y \in \mathbb{R}^n$, SPD(M)
- Output: $b \in \mathbb{R}^p$

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

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

Mixed Models

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

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

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

Mixed Models

- y: **trait**, phenotype (outcome; vector of observations)
  E.g.: height, blood pressure for a set of people
- X: **SNP**, genome measurements and covariates
  (design matrix; predictors). E.g.: sex and age over height

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

Mixed Models

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

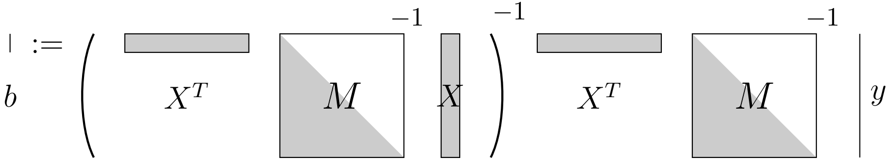
  E.g.: height, blood pressure for a set of people
- X: **SNP**, genome measurements and covariates (design matrix; predictors). E.g.: sex and age over height
- M: **covariance**, dependencies between observations

  E.g.: tall parents have tall children

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

Mixed Models

- y: **trait**, phenotype (outcome; vector of observations)
  E.g.: height, blood pressure for a set of people
- X: **SNP**, genome measurements and covariates
  (design matrix; predictors). E.g.: sex and age over height
- M: **covariance**, dependencies between observations
  E.g.: tall parents have tall children
- b: **effect**, relation between a variation in the outcome (y)
  and a variation in the genome sequence (X)

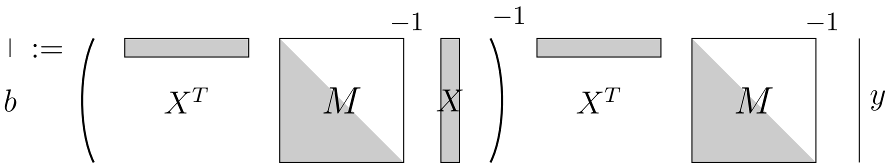


"to be repeated millions of times"

$$\Downarrow$$

for $i = 1, \ldots, m$

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



"to be repeated millions of times"

$$\Downarrow$$

for $i = 1, \ldots, m$

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

$$\Downarrow$$

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
$y_i \in \mathbb{R}^{n}$		8 – 780KBs
$b_i \in \mathbb{R}^{p}$		24 – 160 Bytes
Total	$10^6 \leq m \leq 10^8$	7.5 – 3000 TBs

Streaming problem

- Data does not fit in memory!
  Each instance has to be streamed from and to disk

Streaming problem

- Data does not fit in memory!
  Each instance has to be streamed from and to disk

- Independent problem instances

Streaming problem

- Data does not fit in memory!
  Each instance has to be streamed from and to disk
- Independent problem instances
- Opportunity for overlapping computation and data movement
  "Double buffering"

Streaming problem

```
for( i=0; i < thousands_of_times; i++ )
   {
      /* LOAD  problem_i */
      X = read( "file_X", n * p  , offset, ... );
      y = read( "file_y", n      , offset, ... );
      M = read( "file_M", n * n/2, offset, ... );
      // or maybe:  M = generate_matrix( ... )


      /* SOLVE  problem_i */
      b = compute( X, M, y );


      /* STORE  solution_i */
      write( "file_b", b, p, offset, ... );
   }
```

Language issues

$$\textbf{for } i = 1, \ldots, m \qquad\qquad\qquad m \approx 10^6 - 10^7$$
$$b_i := \left(X_i^T M_i^{-1} X_i\right)^{-1} X_i^T M_i^{-1} y_i$$

Language issues

$$
\begin{array}{ll}
\textbf{for } i = 1, \ldots, m & \qquad m \approx 10^6 - 10^7 \\
\quad b_i := \left(X_i^T M_i^{-1} X_i \right)^{-1} X_i^T M_i^{-1} y_i &
\end{array}
$$

$$\Downarrow$$

$$
\begin{array}{lll}
\textbf{for } i = 1, \ldots, m & & \\
\quad L_i L_i^T = M_i & \texttt{CHOL} & \\
\quad X_i := L_i^{-1} X_i & \texttt{TRSM} & \\
\quad y_i := L_i^{-1} y_i & \texttt{TRSV} & \\
\quad b_i := \texttt{OLS}(X_i, y_i\ ) & & O(n^3 m)
\end{array}
$$

Language issues – wrong problem definition

for $i = 1, \dots, m$ $\qquad m \approx 10^6 - 10^7$

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

$$\Downarrow$$

for $i = 1, \dots, m$

$L_i L_i^T = M_i$ $\qquad$ CHOL

$X_i := L_i^{-1} X_i$ $\qquad$ TRSM

$y_i := L_i^{-1} y_i$ $\qquad$ TRSV

$b_i := \texttt{OLS}(X_i, y_i)$ $\qquad O(n^3 m)$

Problem definition (2nd attempt)

Sequence of GLS problems

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

Problem definition (2nd attempt)

Sequence of GLS problems

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

$$\Downarrow$$

for $i = 1, \dots, m$

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

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

$$\Downarrow$$

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

Different “paradigm”

- **From** loop over indipendent problems
- **To** a sequence of connected problems

Different "paradigm"

- **From** loop over indipendent problems
- **To** a sequence of connected problems

- **From** optimizing the individual iteration
- **To** optimizing the problem as a whole

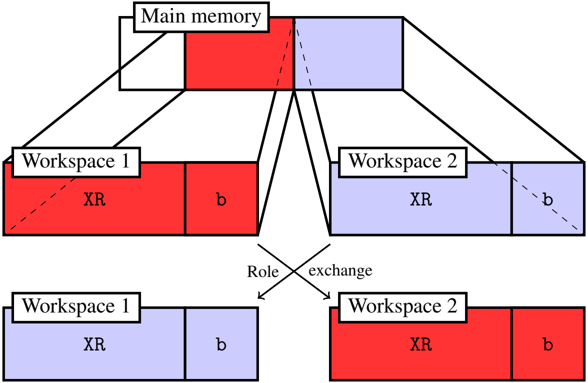
Double buffering

Iteration i

- Store `b[i-1]`
- Load `XR[i+1]`

Iteration $i+1$

- Compute `b[i+1]`



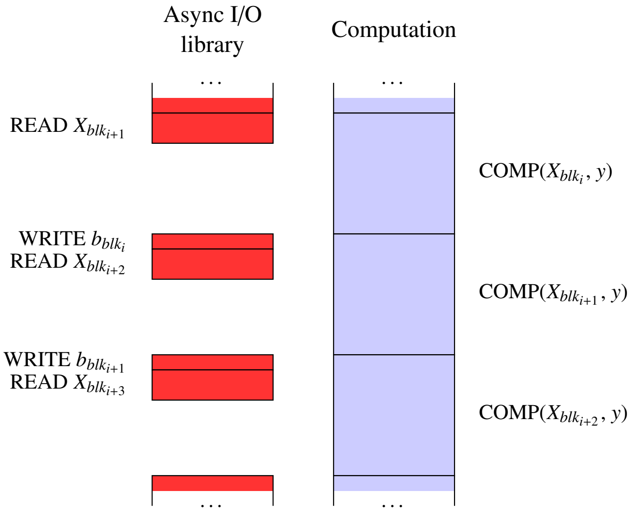
Iteration i

- Compute `b[i]`

Iteration $i+1$

- Store `b[i]`
- Load `XR[i+2]`

Communication-computation overlap



Problem definition – after many attempts!

Two-dimensional grid of **correlated** GLS problems

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

$$\Downarrow$$

for $i = 1, \ldots, m$
  for $j = 1, \ldots, t$
    $b_{ij} := \left(X_i^T M_j^{-1} X_i \right)^{-1} X_i^T M_j^{-1} y_j$

Problem definition – after many attempts!

Two-dimensional grid of **correlated** GLS problems

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

$$\Downarrow$$

for $i = 1, \dots, m$
  for $j = 1, \dots, t$
    $b_{ij} := \left(X_i^T M_j^{-1} X_i \right)^{-1} X_i^T M_j^{-1} y_j$

with

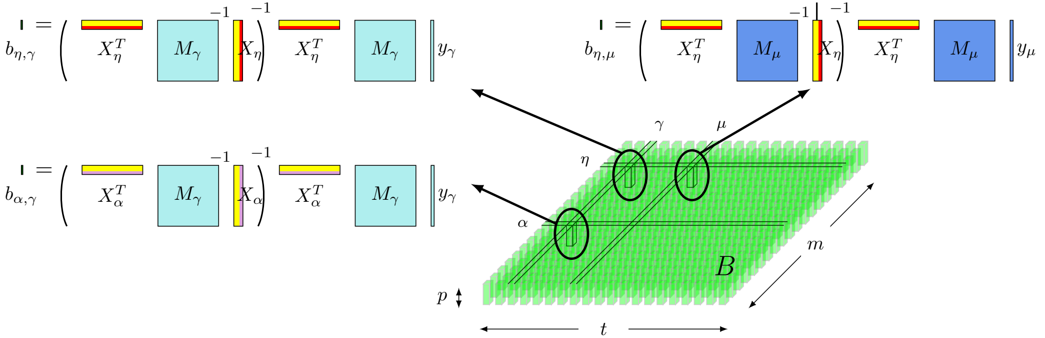
$m \approx 10^6 - 10^7$, $t = 1$ or $\approx 10^3 - 10^5$,

and

$X_i = [X_L | X_{Ri}]$ with $X_{Ri} \in \mathcal{R}^{n \times 1}$,

SPD(M_j) and $M_j = \sigma_j \Phi + h_j I$ $\leftarrow$ **very important!**

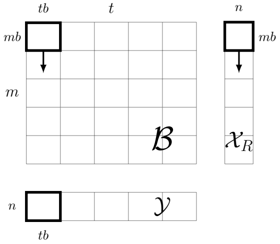
Overview of the (beautiful) problem



Problem size

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

Knowledge from architecture $\rightarrow$ efficiency increased

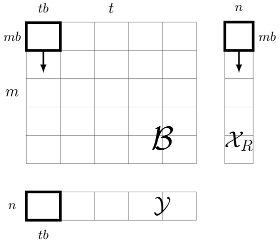


$\mathcal{X}$ and $\mathcal{Y}$ are streamed from disk.

$\mathcal{B}$ is computed by tiles.

- Direction of traversal?
- Size of the tiles?
- How to overlap comp. with comm.?

Knowledge from architecture → efficiency increased



$\mathcal{X}$ and $\mathcal{Y}$ are streamed from disk.

$\mathcal{B}$ is computed by tiles.

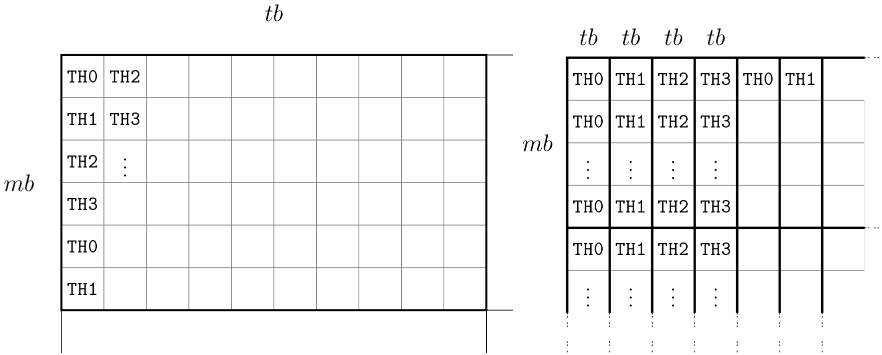
- Direction of traversal?
- Size of the tiles?
- How to overlap comp. with comm.?

Models

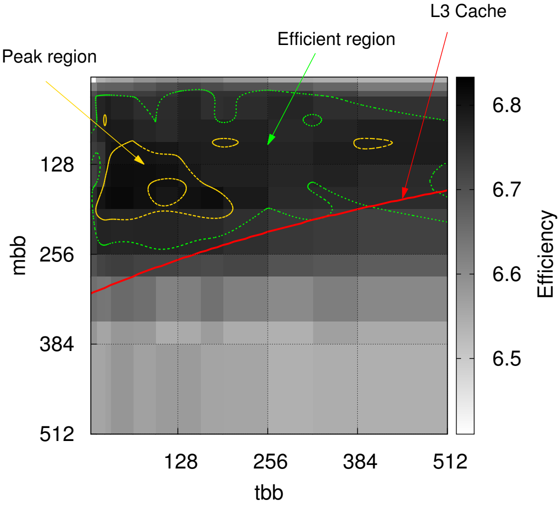
$$\frac{\texttt{\# flops}}{\texttt{\# flops/sec}} > \frac{\texttt{data\_to\_load + data\_to\_store}}{\texttt{IO\_bandwidth}}$$

$$\frac{tb \times (5 + 2(p-1)) \times n}{\texttt{\# flops/sec}} > \frac{(n + tb \times p) \times \texttt{sizeof(datatype)}}{\texttt{IO bandwidth}}$$

- How to assign threads to the tiles?



- Optimal block size?



Knowledge from application $\rightarrow$ complexity lowered

$$\begin{cases} b_{ij} := (X_i^T M_j^{-1} X_i)^{-1} X_i^T M_j^{-1} y_j \\ M_j = \sigma_j \Phi + h_j I \end{cases}$$

Knowledge from application $\rightarrow$ complexity lowered

$$\begin{cases} b_{ij} := (X_i^T M_j^{-1} X_i)^{-1} X_i^T M_j^{-1} y_j \\ M_j = \sigma_j \Phi + h_j I \end{cases}$$

$$Q \Lambda Q^T = \Phi$$

$$\Rightarrow M_j = Q \; (\sigma_j \Lambda + h_j I) \; Q^T$$

$$\Rightarrow \boxed{M_j^{-1} = Q \; (\sigma_j \Lambda + h_j I)^{-1} \; Q^T}$$

Knowledge from application $\rightarrow$ complexity lowered

$$\begin{cases} b_{ij} := (X_i^T M_j^{-1} X_i)^{-1} X_i^T M_j^{-1} y_j \\ M_j = \sigma_j \Phi + h_j I \end{cases}$$

$$Q \Lambda Q^T = \Phi$$

$$\Rightarrow M_j = Q \ (\sigma_j \Lambda + h_j I) \ Q^T$$

$$\Rightarrow \boxed{M_j^{-1} = Q \ (\sigma_j \Lambda + h_j I)^{-1} \ Q^T}$$

$$\begin{aligned} b_{ij} := (X_i^T Q D_j^{-\frac{1}{2}} D_j^{-\frac{1}{2}} Q^T X_i)^{-1} \times \\ X_i^T Q D_j^{-\frac{1}{2}} D_j^{-\frac{1}{2}} Q^T y_j \end{aligned}$$

Knowledge from application → complexity lowered

$$\begin{cases} b_{ij} := (X_i^T M_j^{-1} X_i)^{-1} X_i^T M_j^{-1} y_j \\ M_j = \sigma_j \Phi + h_j I \end{cases}$$

$$Q \Lambda Q^T = \Phi$$

$$\Rightarrow M_j = Q\ (\sigma_j \Lambda + h_j I)\ Q^T$$

$$\Rightarrow \boxed{M_j^{-1} = Q\ (\sigma_j \Lambda + h_j I)^{-1}\ Q^T}$$

$$b_{ij} := (X_i^T Q D_j^{-\frac{1}{2}} D_j^{-\frac{1}{2}} Q^T X_i)^{-1} \times X_i^T Q D_j^{-\frac{1}{2}} D_j^{-\frac{1}{2}} Q^T y_j$$

```
Q Λ Q^T = Φ
for 1 ≤ i ≤ m
    X'_i := Q^T X_i                          GEMM
for 1 ≤ j ≤ t
    y'_j := Q^T y_j                          GEMV
for 1 ≤ j ≤ t
    D_j := σ_j^2 (h_j^2 Λ + (1 − h_j^2) I)
    K_j K_j^T = D_j^(−1/2)
    v_j := K_j^T y'_j
    for 1 ≤ i ≤ m
        W_ij := K_j^T X'_i
        S_ij := W_ij^T W_ij
        b_ij := W_ij^T v_j
        b_ij := S_ij^(−1) b_ij
```

Cost: $O(n^2 m t)$ vs. $O(nmt)$

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?

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

Model-based prediction

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

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, CPU+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, CPU+GPU implementation

- NO $\Rightarrow$ distributed memory + MPI

  partitioning + streaming HD$\leftrightarrow$CPUs, double buffering, data distribution

Algorithm

$$
\begin{aligned}
&LL^T = M \\
&X := L^{-1}X \\
&S := X^T X \\
&GG^T = S \\
&y := L^{-1}y \\
&b := X^T y \\
&b := G^{-1}b \\
&b := G^{-T}b
\end{aligned}
$$

Algorithm – bottleneck?

$$LL^T = M$$

$X := L^{-1}X$ $\rightarrow$ to accelerator (`TRSM`)

$$S := X^T X$$

$$GG^T = S$$

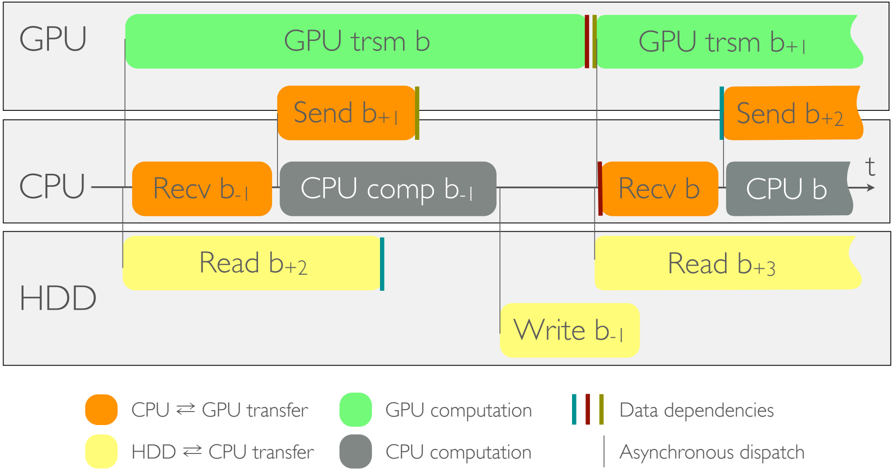
$$y := L^{-1}y$$

$$b := X^T y$$

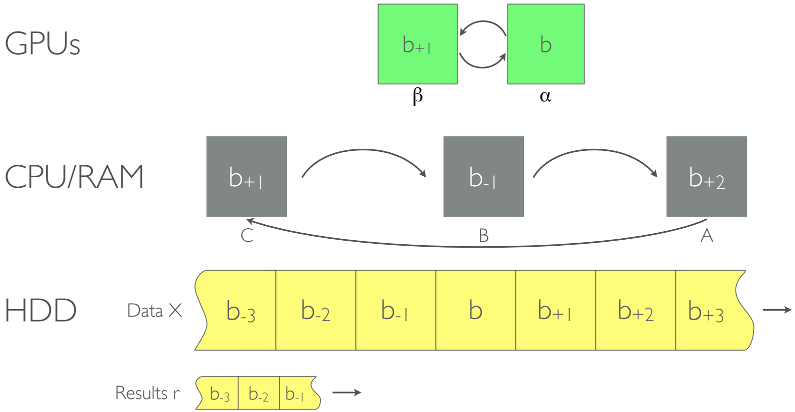
$$b := G^{-1}b$$

$$b := G^{-T}b$$

Double+triple buffering

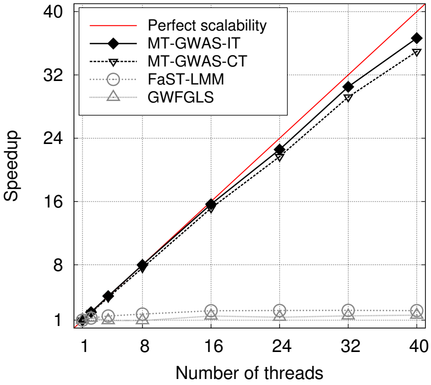


Double+triple buffering



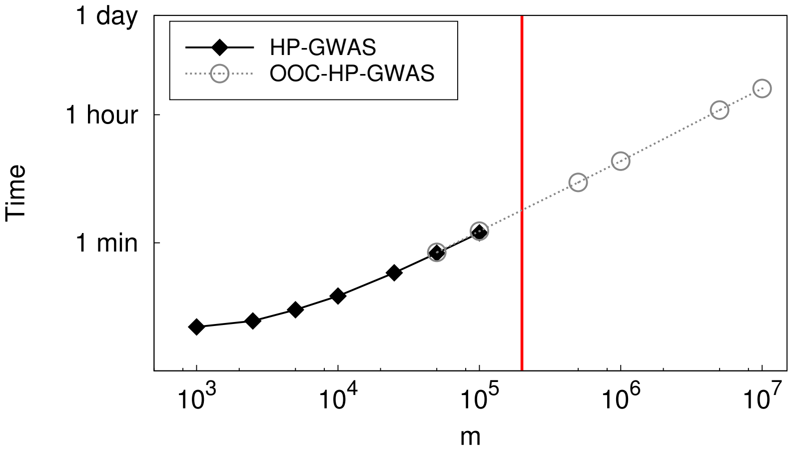
Results: Single node

Scalability   Intel Xeon E7-4850 $n = 1000,\ p = 4,\ m = 100000,\ t = 20000$



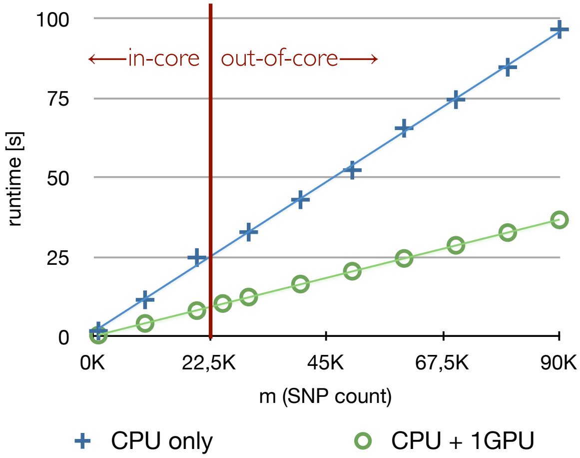
Results: Beyond memory capacity

Intel Xeon E7-4850 Main memory: 32GB $n = 10000,\ p = 4,\ t = 1$



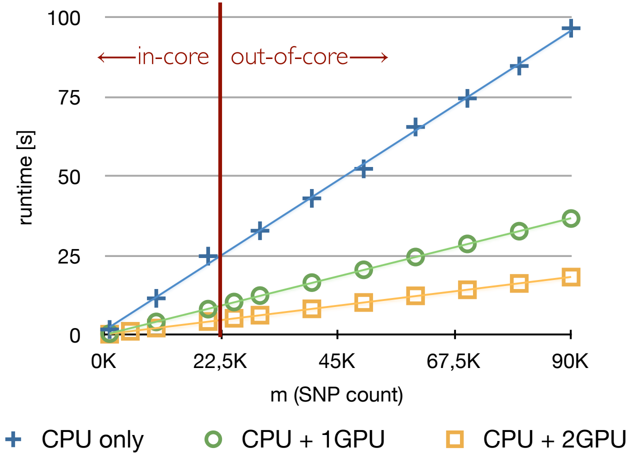
Results: Beyond memory capacity

nVidia Quadro 6000 GPUs $\quad n = 30000,\ p = 4,\ t = 10000$



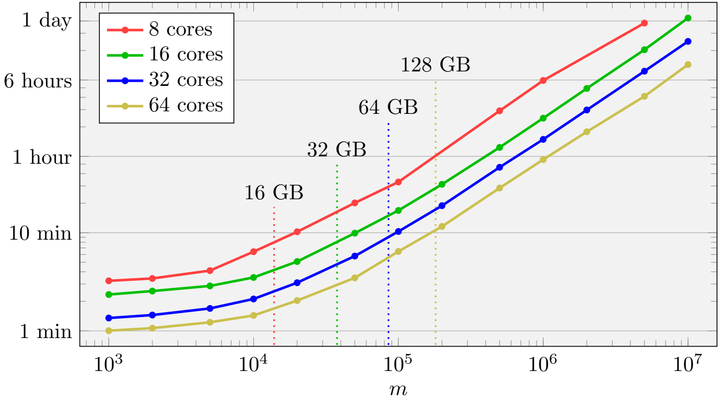
Results: Beyond memory capacity

nVidia Quadro 6000 GPUs $n = 30000,\ p = 4,\ t = 10000$



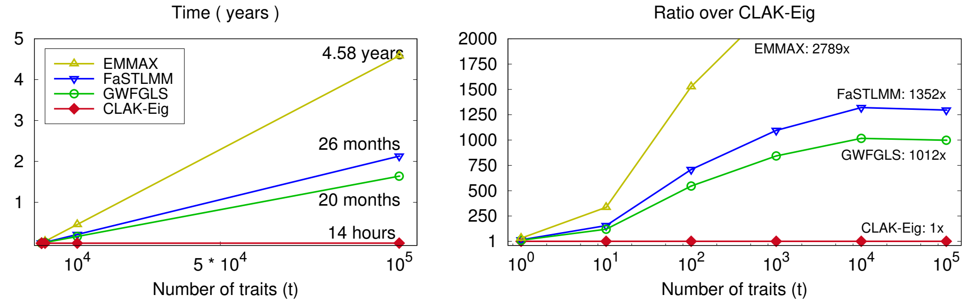
Results: Beyond memory capacity

8 Intel Xeon E7-4850 $\quad n = 30000,\ p = 4,\ t = 10000$



Results: Full GWAS

8 Intel Xeon E7-4850 $n = 1000,\ p = 4,\ m = 1000000$



Not one algorithm to rule them all

- $t > 1$
  "Computing Petaflops over Terabytes of Data: The Case of Genome-Wide Association Studies", D. Fabregat-Traver and P. Bientinesi. ACM Transactions on Mathematical Software (TOMS), Vol.40(4), pp.27:1–27:22, June 2014.

- $t = 1$
  "Solving Sequences of Generalized Least-Squares Problems on Multi-threaded Architectures", D. Fabregat-Traver, Y. S. Aulchenko and P. Bientinesi. Applied Mathematics and Computation, Vol.234, pp.606–617, May 2014.

- GPU
  "Streaming Data from HDD to GPUs for Sustained Peak Performance", L. Beyer and P. Bientinesi. Proceedings of the Euro-Par 2013, Lecture Notes in Computer Science, Vol.8097, pp.788-799, May 2013.

- MPI
  "High Performance Solutions for Big-data GWAS", E. Peise, D. Fabregat-Traver and P. Bientinesi. Parallel Computing, Vol.42, pp.75–87, February 2015.

- OmicABEL: `https://github.com/lucasb-eyer/OmicABEL`
  cuGWAS: `https://github.com/lucasb-eyer/cuGWAS`