Genome-Wide Association Studies: Computing Petaflops over Terabytes of Data

Paolo Bientinesi

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

Blue Gene Active Storage Workshop
January 10th, 2013
JSC, Jülich

Acknowledgments

Yurii Aulchenko

Biologist

Diego Fabregat

Main developer

Lucas Beyer

GPUer

Deutsche Forschungsgemeinschaft
DFG

1. The problem
2. Single trait analysis: $t = 1$
3. The complete multi-trait analysis: $t > 1$
4. Future work

The problem - biology

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

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)

The problem - linear algebra

Linear regression with non-independent outcomes

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

Genome-wide association study

Inputs

- $M \in \mathcal{R}^{n \times n}$, $SPD(M)$, $n \in [10^3, \dots, 10^4]$
- $X \in \mathcal{R}^{n \times p}$, $p \in [3, \dots, 20]$, full rank
- $y \in \mathcal{R}^n$

Output

- $b \in \mathcal{R}^p$

The problem - linear algebra

Linear regression with non-independent outcomes

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

Genome-wide association study

Inputs

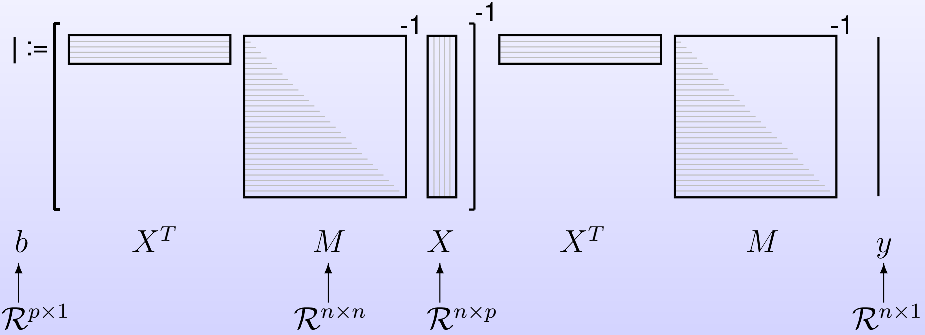
- $M \in \mathcal{R}^{n \times n}$, $SPD(M)$, $n \in [10^3, \dots, 10^4]$
- $X \in \mathcal{R}^{n \times p}$, $p \in [3, \dots, 20]$, full rank
- $y \in \mathcal{R}^n$

Output

- $b \in \mathcal{R}^p$

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

Problem instance



$p \in [4, \ldots, 20] \qquad n \in [10^3, \ldots, 10^4]$

Language issues

"**To be repeated thousands of times**"

Language issues

“**To be repeated thousands of times**”

$\Rightarrow$

```
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

"**To be repeated thousands of times**"

$\Rightarrow$

- independent problem instances
  - computational cost: $O(n^3 + n^2p + np^2) = O(n^3)$

Language issues

"To be repeated thousands of times"

$\Rightarrow$

- independent problem instances
  - computational cost: $O(n^3 + n^2p + np^2) = O(n^3)$
- access pattern irrelevant

Language issues

“To be repeated thousands of times”

$\Rightarrow$

- independent problem instances
  - computational cost: $O(n^3 + n^2p + np^2) = O(n^3)$
- access pattern irrelevant
- opportunity for overlapping computation and data movement
  - double buffering

The problem - the real setup

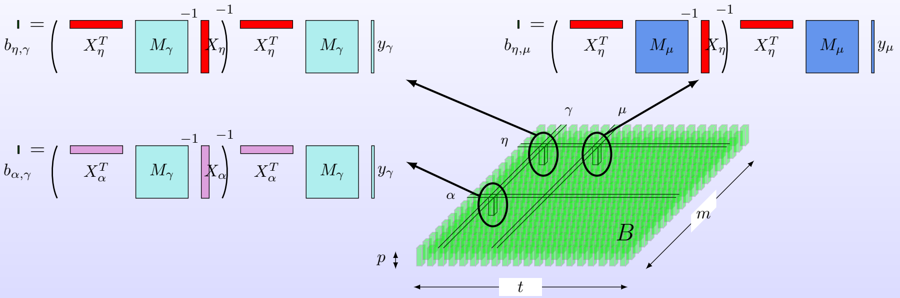
The problem - the real setup

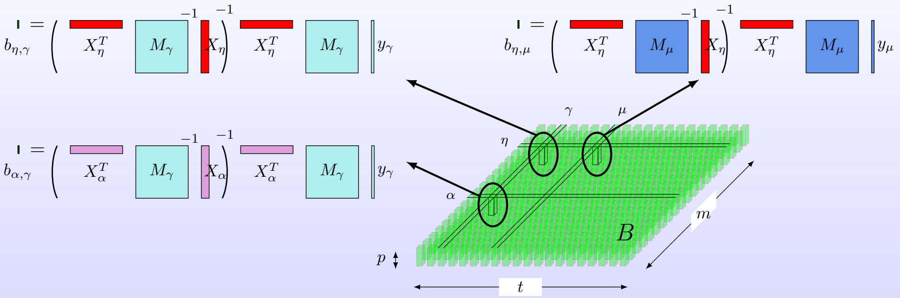
2D sequence

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

$i = 1 \ldots m,$ where $m \approx 10^6 - 10^7$
$j = 1 \ldots t,$ where t is either 1 or $\approx 10^5$

Also:
for each b_{ij}, compute and store $v_{ij} \in \mathcal{R}^{p \times p}$


$b_{\eta,\gamma}$ = (X_η^T M_γ^{-1} X_η)$^{-1}$ X_η^T M_γ^{-1} y_γ
$b_{\eta,\mu}$ = (X_η^T M_μ^{-1} X_η)$^{-1}$ X_η^T M_μ^{-1} y_μ
$b_{\alpha,\gamma}$ = (X_α^T M_γ^{-1} X_α)$^{-1}$ X_α^T M_γ^{-1} y_γ
γ
μ
η
α
B
m
p
t



A completely different problem!

- data re-use
- cost depends on traversal
- HUGE datasets — Terabytes

Complexity

Parameters

- n: population size
- m: number of SNP
- t: number of traits

Complexity

Parameters

- n: population size
- m: number of SNP
- t: number of traits

Scenario #1

$t = 1$
$n = 10.000$
$m = 3.000.000$

Input: 220 GBytes
Output: .2 GBytes

Cost: 3.000 Pflops

Scenario #2

$t = 100.000$
$n = 1.000$
$m = 1.000.000$

Input: 8 GBytes
Output: 9 TBytes

Cost: 100.000 Pflops

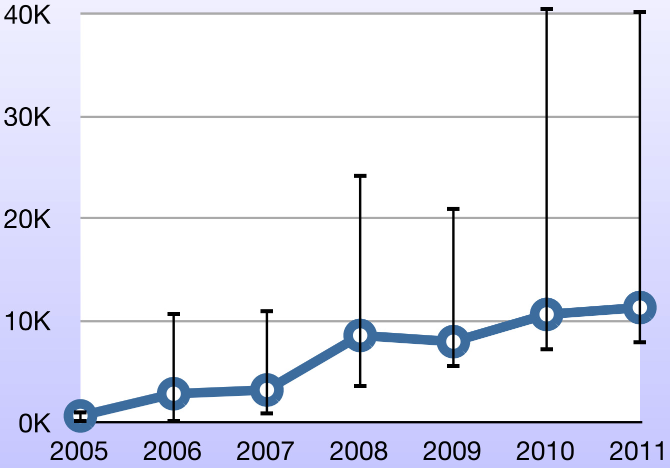
Scenario #3

$t = 100.000$
$n = 10.000$
$m = 10.000.000$

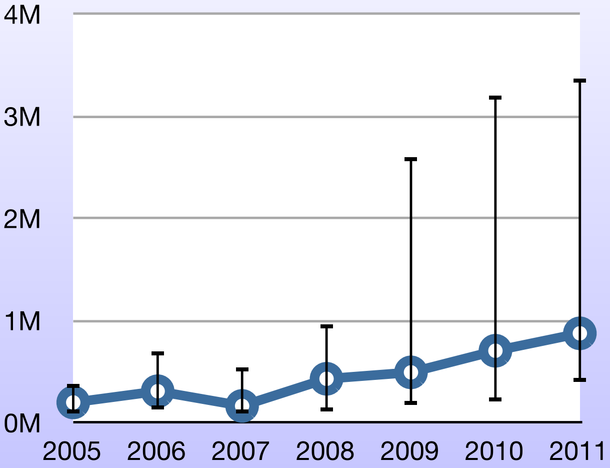
Input: 750 GBytes
Output: 90 TBytes

Cost: 1.000.000 **Eflops**

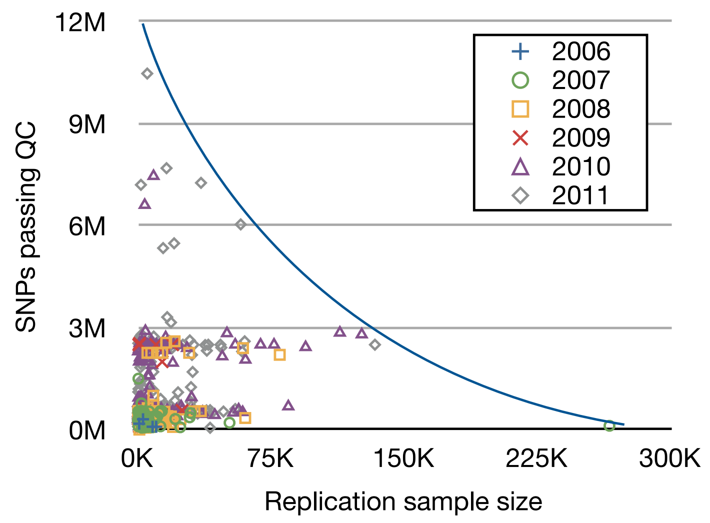
GWAS database: Population size – n



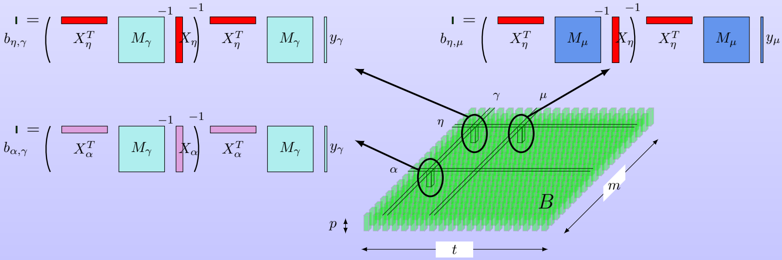
GWAS database: SNPs – m



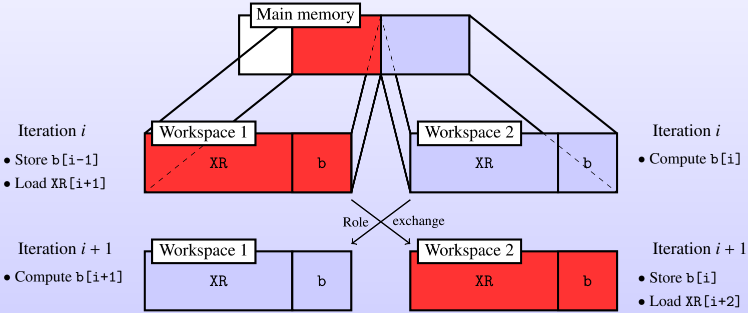
GWAS database: Population vs SNPs



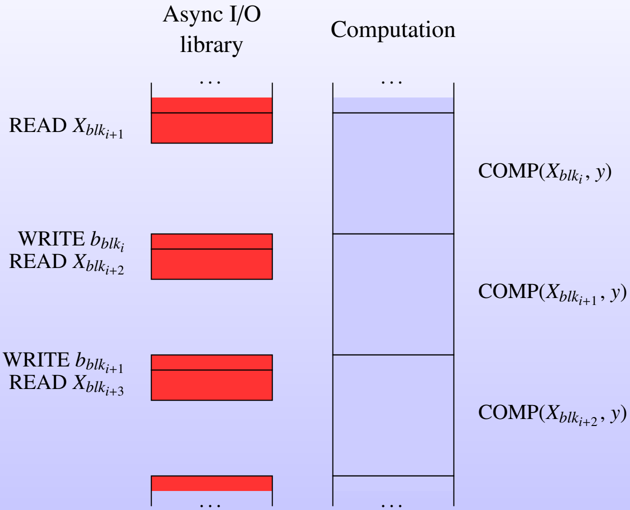
1. The problem
2. Single trait analysis: $t = 1$
3. The complete multi-trait analysis: $t > 1$
4. Future work



Double buffering



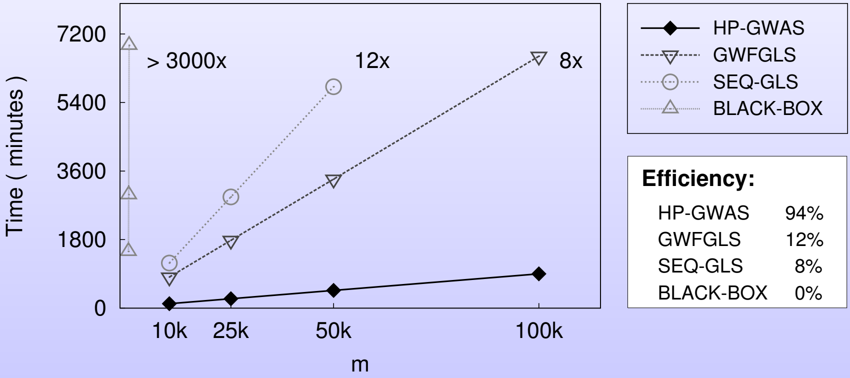
Overlap



Async I/O library
Computation
...
READ $X_{blk_{i+1}}$
COMP(X_{blk_i}, y)
WRITE b_{blk_i}
READ $X_{blk_{i+2}}$
COMP($X_{blk_{i+1}}$, y)
WRITE $b_{blk_{i+1}}$
READ $X_{blk_{i+3}}$
COMP($X_{blk_{i+2}}$, y)
...

Our algorithm: HP-GWAS

Xeon, single thread $n = 10.000, p = 4$

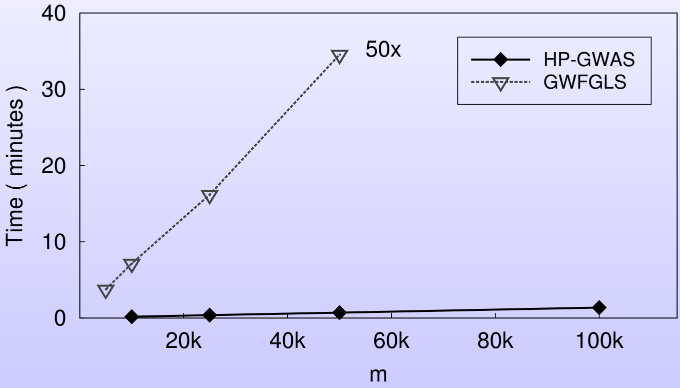


Efficiency:

HP-GWAS	94%
GWFGLS	12%
SEQ-GLS	8%
BLACK-BOX	0%

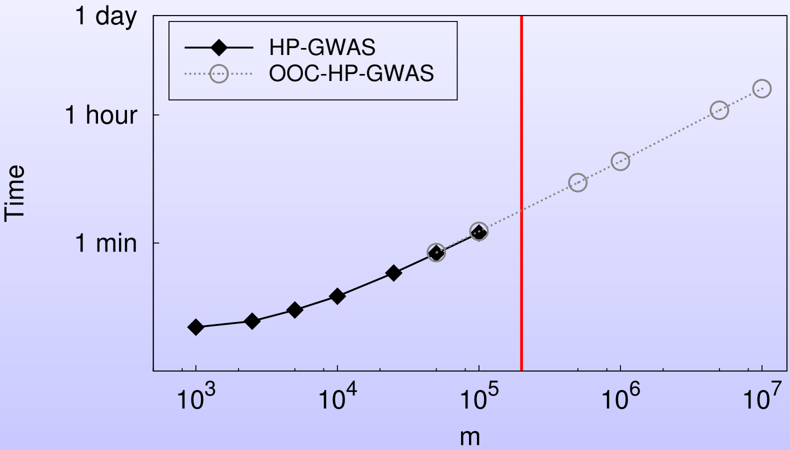
Our algorithm: HP-GWAS

Xeon, 12-cores $n = 10.000, p = 4$



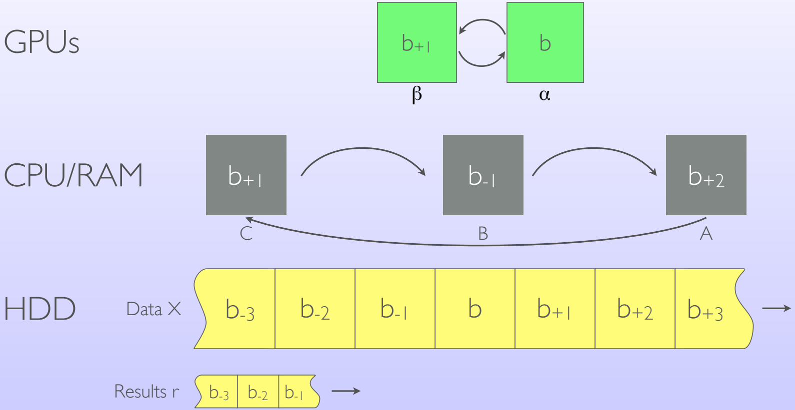
In-core vs Out-of-core

Main memory: 32GB $n = 10.000, p = 4$



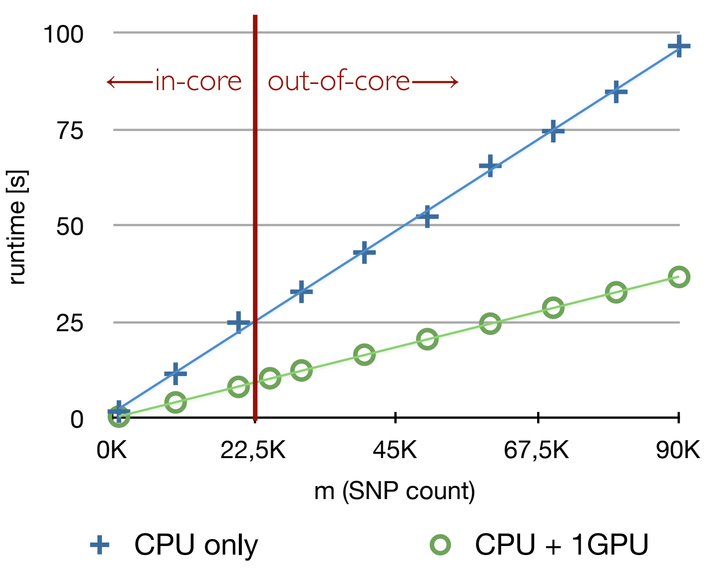
GPUs: Double+Triple buffering

GPUs: Double+Triple buffering

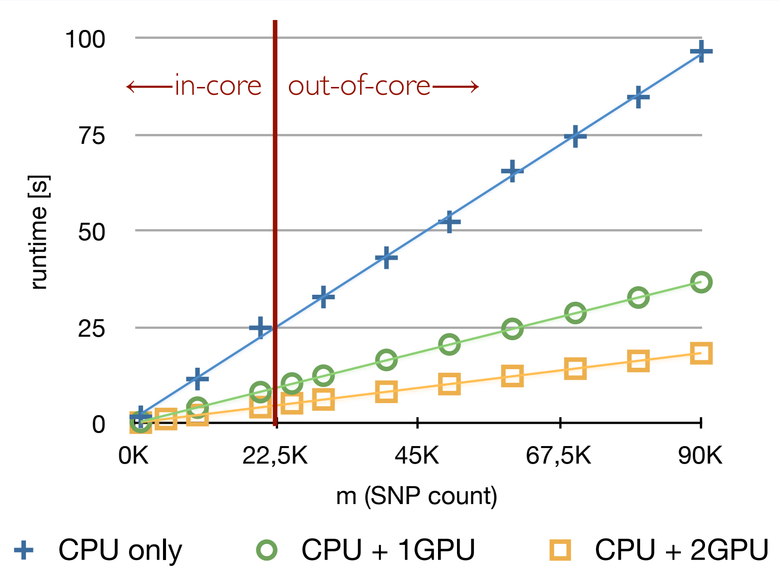


GPU: Results

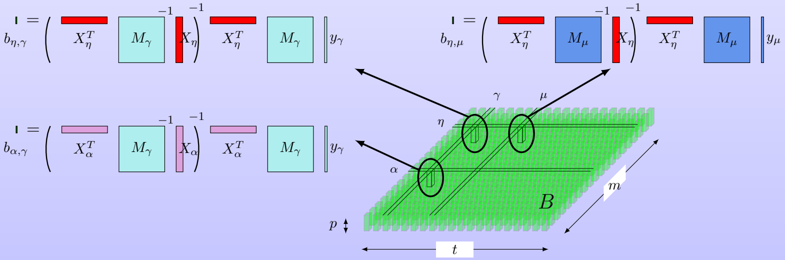
Main memory: 24GB, Fermi: 6GB $n = 10.000, p = 4$



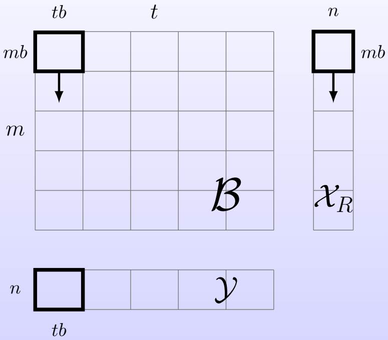
Multiple GPUs



1. The problem
2. Single trait analysis: $t = 1$
3. **The complete multi-trait analysis: $t > 1$**
4. Future work

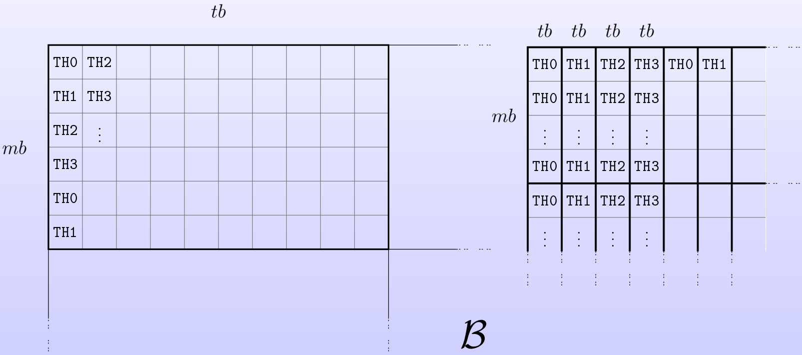


Degrees of freedom



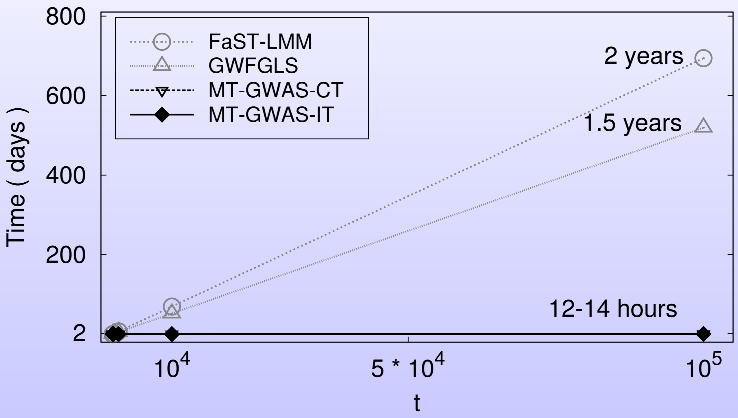
- I/O: How to traverse B? (blocking / shape / direction)
- Single-threaded or collaborative I/O?
- Caching: How to use multi-threading? (blocking / shape)

I/O management



Results

Xeon, 512GB, 40 threads $n = 1.000, p = 4, m = 1.000.000$



1 The problem

2 Single trait analysis: $t = 1$

3 The complete multi-trait analysis: $t > 1$

4 Future work

Future work

GWAS: is it all solved?

- $t = 1$: 40-fold speedups on 12 cores
- $t = 1$: k-fold speedup with k GPUs
- $t > 1$: 1000-fold speedups on 40 cores

Future work

GWAS: is it all solved?

- $t = 1$: 40-fold speedups on 12 cores
- $t = 1$: k-fold speedup with k GPUs
- $t > 1$: 1000-fold speedups on 40 cores
- But ...

Future work

GWAS: is it all solved?

- $t = 1$: 40-fold speedups on 12 cores
- $t = 1$: k-fold speedup with k GPUs
- $t > 1$: 1000-fold speedups on 40 cores
- But ...
  none of our I/O mechanisms will be accepted
- sacrifice overlap for filtering, casting, post-processing, ...

Future work

GWAS: is it all solved?

- $t = 1$: 40-fold speedups on 12 cores
- $t = 1$: k-fold speedup with k GPUs
- $t > 1$: 1000-fold speedups on 40 cores
- But . . .
  none of our I/O mechanisms will be accepted
- sacrifice overlap for filtering, casting, post-processing, . . .

TODO list

- Large n: distributed memory
- GPUs for multi-trait GWAS
- Exploit sparsity