Investigating Scaling Behaviour of Monte Carlo Codes for Dense Matrix Inversion

V. Alexandrov (BSC, Spain)
J. Straßburg (BSC,Spain & UoR, UK)

Outline

Motivation

Background on Linear Algebra Problems and Monte Carlo

Scaling experiments

Conclusions

Motivation

Predict behaviour on different system

Find bottlenecks, sweet spot, scaling problems

Easier then running on several machines

Reproducible

Scientific and Engineering Problems

Many scientific problems revolve around:

inverting a real n by n matrix (MI)

- Given A
- Find A^{-1}

solving a system of linear algebraic equations (SLAE)

- Given A and b
- Solve, for x, $Ax = b$
- Or find A^{-1} and calculate $x = A^{-1}b$

Traditional Methods

- Gaussian elimination
- Gauss-Jordan
- Both take $O(n^3)$ steps

Time prohibitive if:

- a large problem
- or a real time solution is required

Idea behind Monte Carlo Methods

Wish to estimate the quantity α

Define a random variable ξ

Where ξ has the mathematical expectation α

Take N independent realisations ξ_i of ξ

- Then $\bar{\xi} = \frac{1}{N} \sum_{i=1}^{N} \xi_i$
- And $\bar{\xi} \approx \alpha$

Reason for Using Monte Carlo

$O(NT)$ steps to find an element of the:

- matrix inverse A^{-1}
- solution vector x

where:

- N number of Markov Chains
- T length of Markov Chains

Independent of n - size of matrix or problem

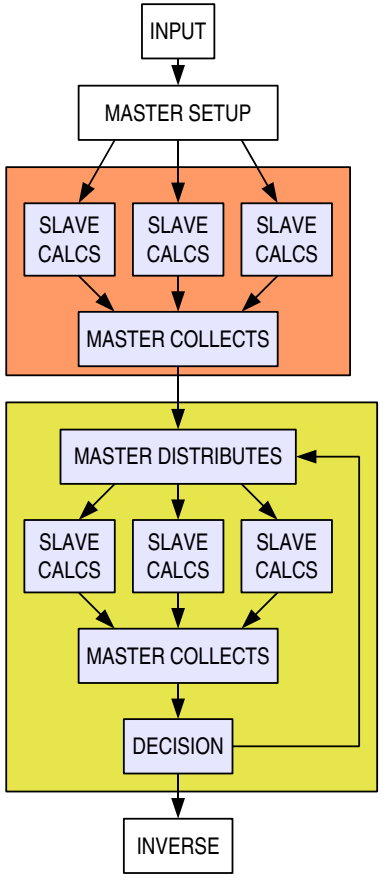
Algorithms can be efficiently parallelised

Parallel Algorithm

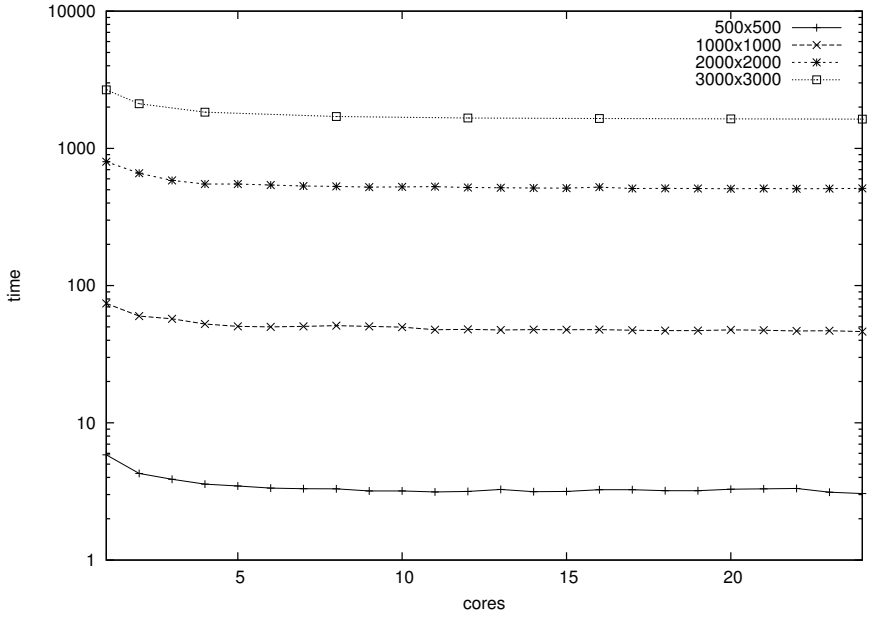
Matrix Setup

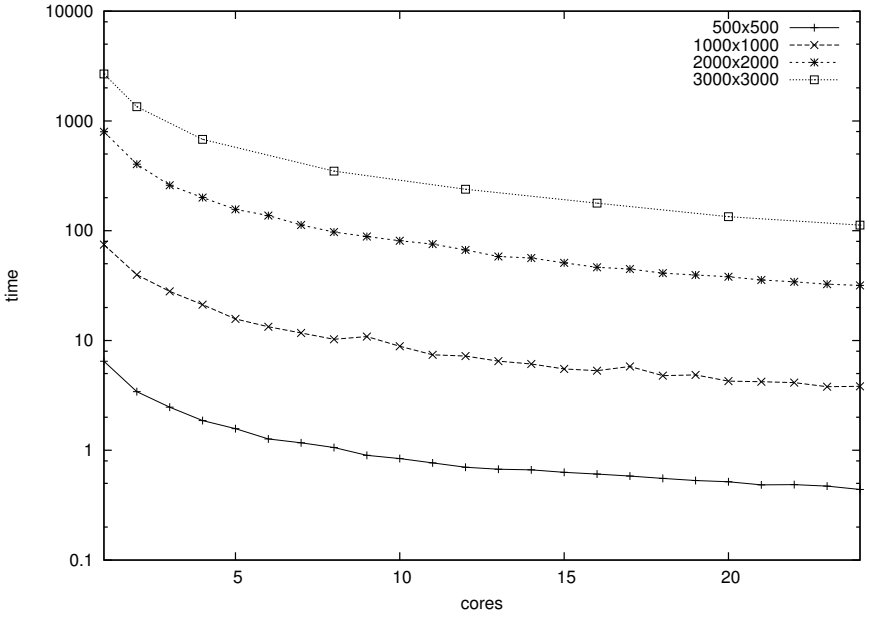
Use parallel Monte Carlo to find B^{-1}

Use parallel iterative refinement to improve accuracy of B^{-1}



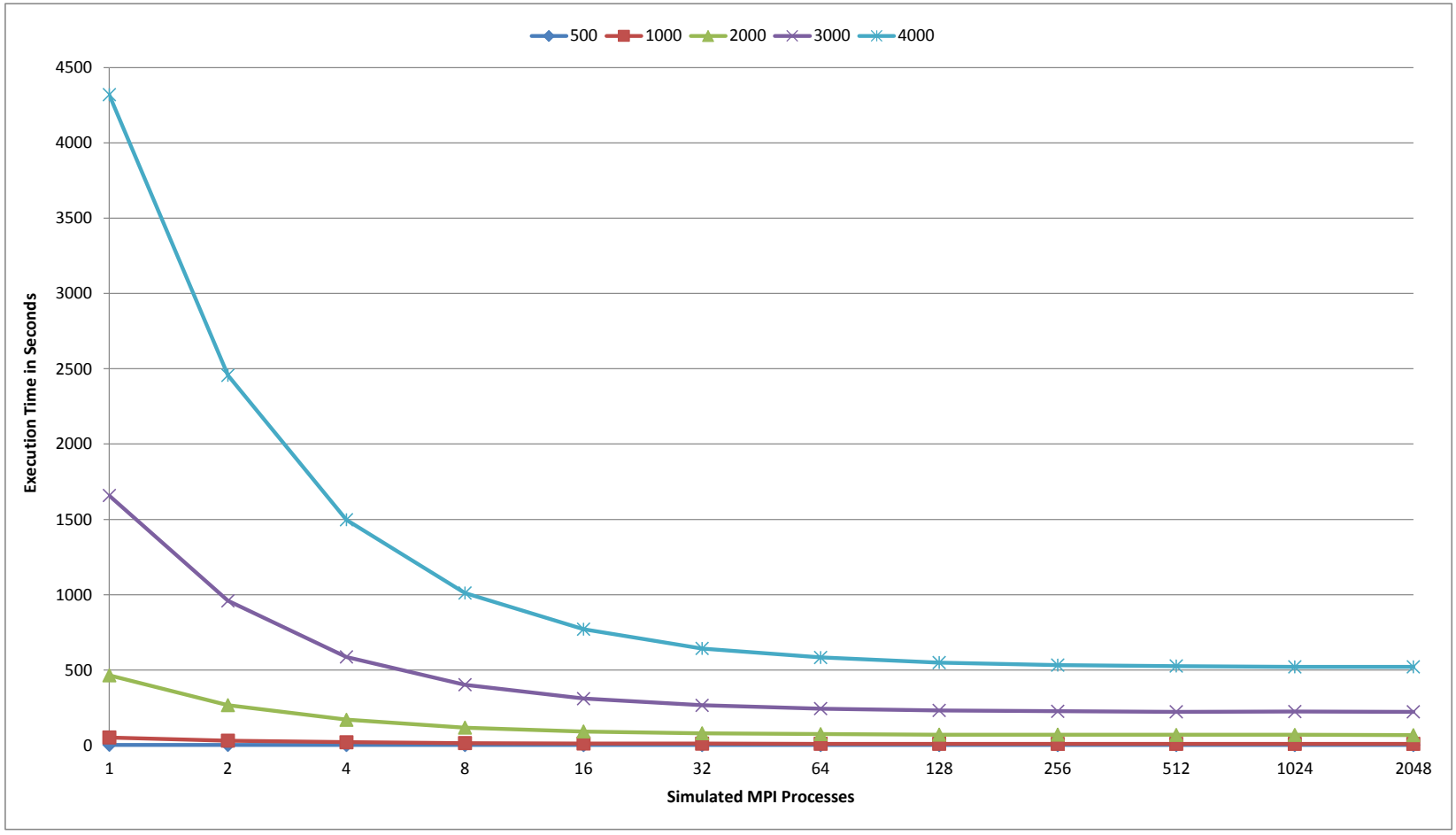
Native vs. simulated scaling





24 core SMP development machine
similar on ORNL cluster

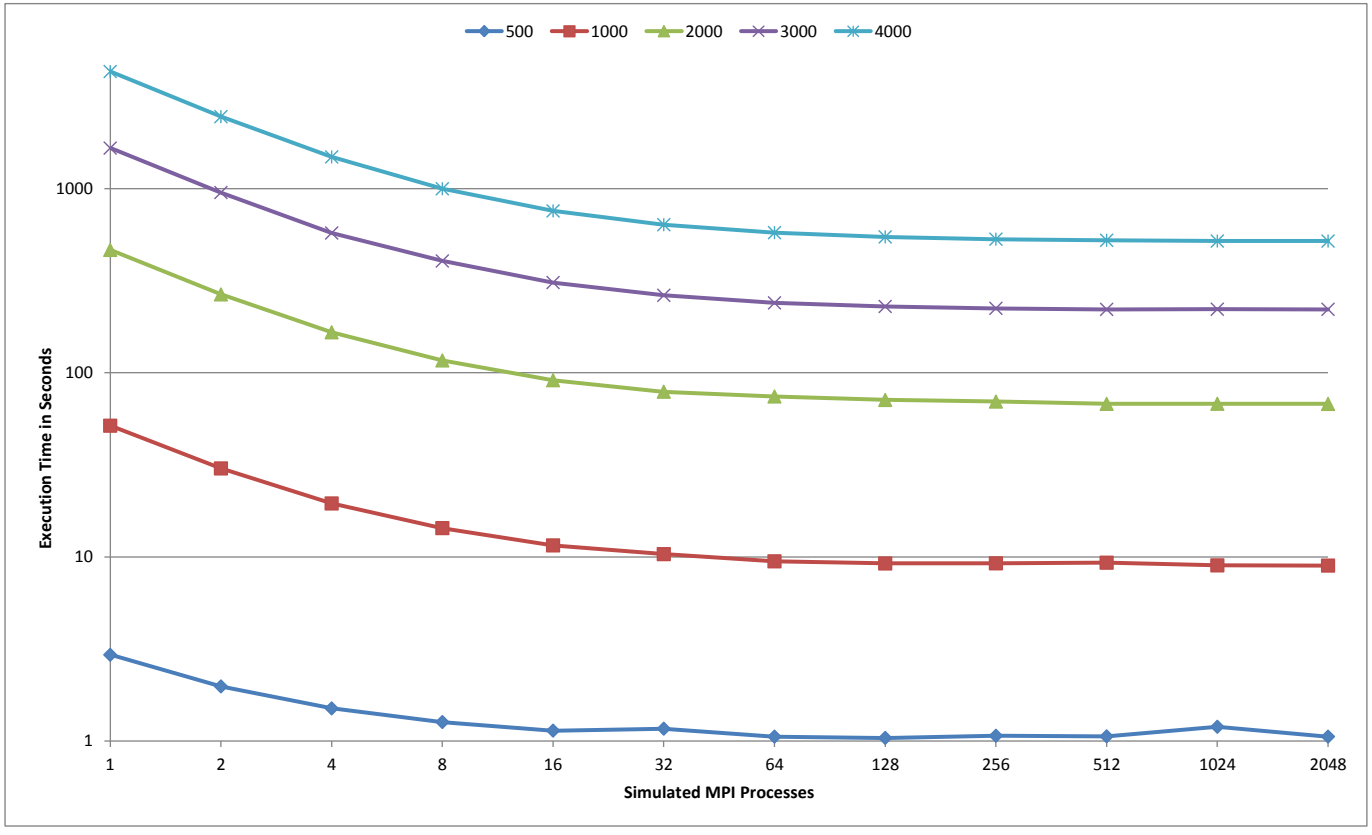
Core scaling



960 core system

240 cores for simulation due to memory bandwidth restrictions

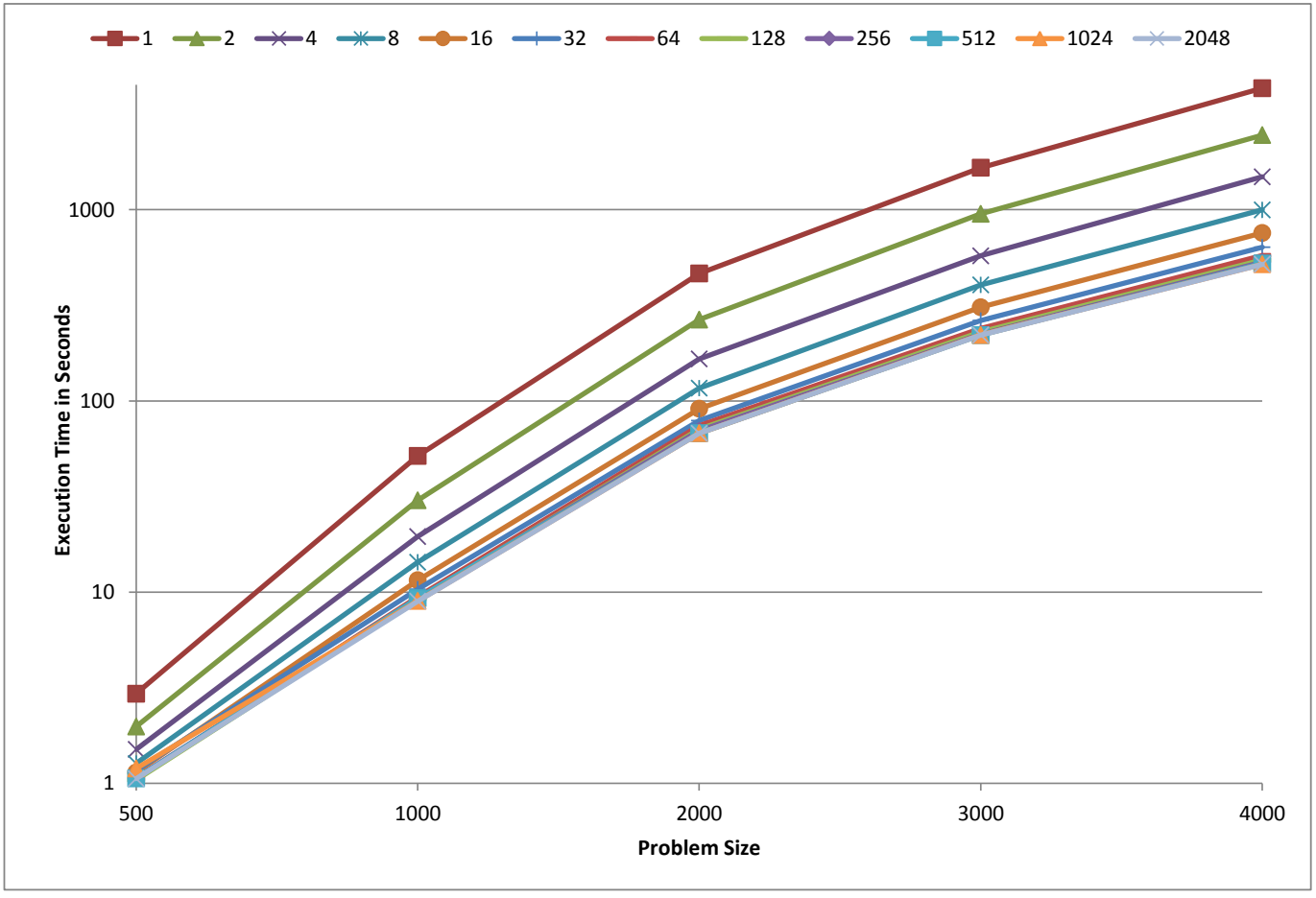
Core scaling



Recurring behaviour for increasing MPI process sizes

Scales well, then plateaus

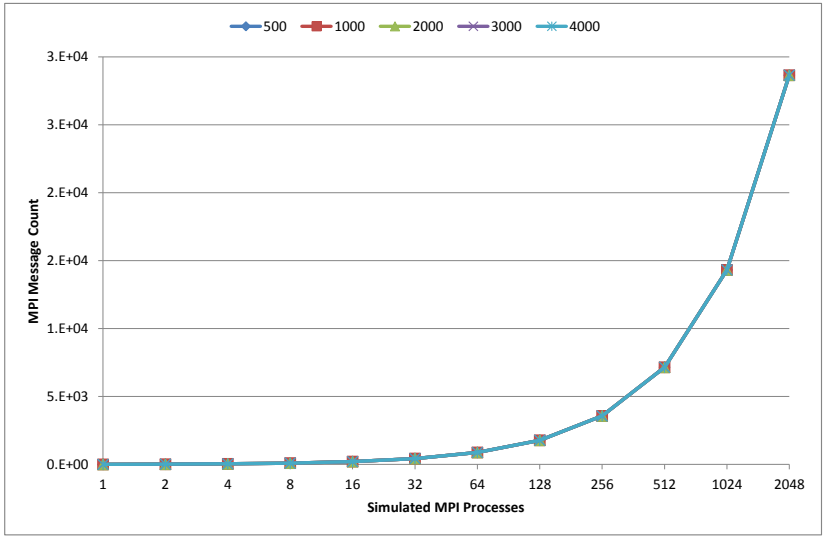
Problem scaling

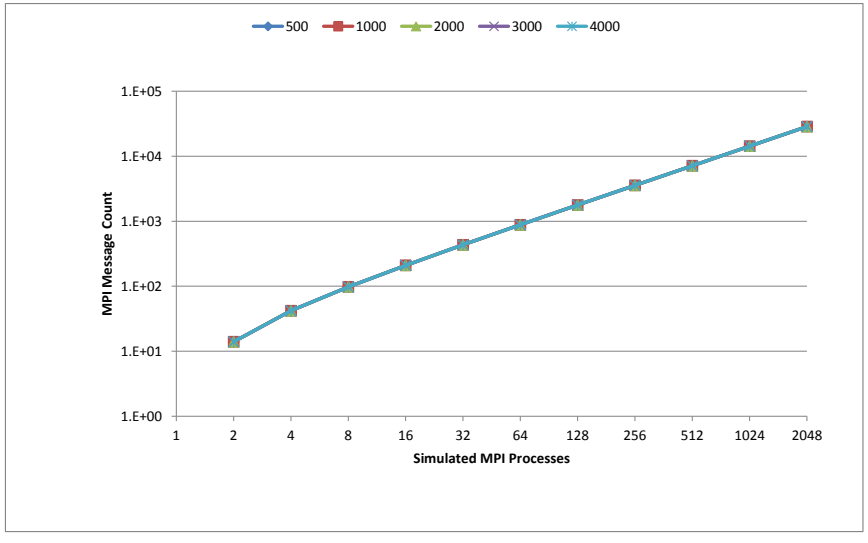


Linear behaviour up to 2000x2000 matrix size

Slight degradation for larger problem sizes

MPI message count scaling





Simulator also gathers MPI statistics

Linear increase of exchanged messages, as expected

Outcome & Conclusions

Behaviour of code is predictable

Simulation provides valuable information

Forecast behaviour on varying systems possible

Time and resource concerns

Future work

Improve the code and retest

Profiling + scaling simulation

Optimize program to be able to handle simulation of larger problem sizes

Runs on larger parallel systems, different networks