

SYSC5906 - Directed Studies
(Distributed Sparse Matrices)

Performance Testing

Introduction

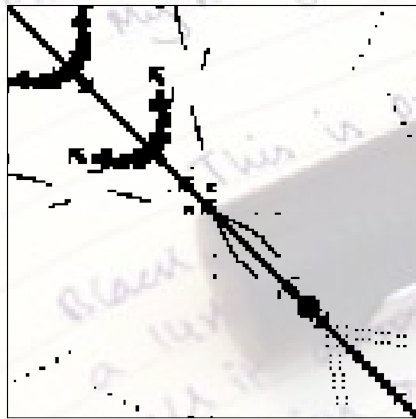
As promised

- a demo,
- an overview of the code,
- maybe some results if there's anything interesting to look at, and/or
- a bit of a retrospective, the good, bad, and ugly.



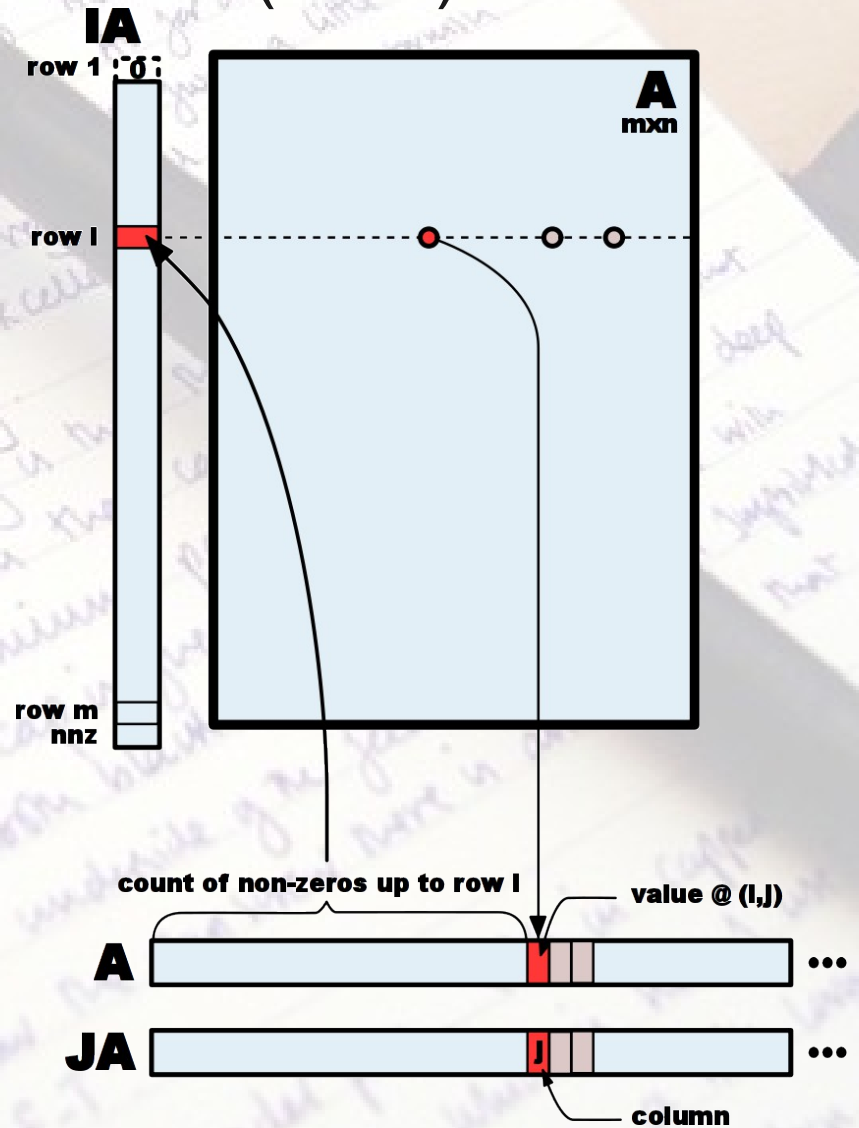
Review

A “sparse” matrix



(Few non-zero entries)

Compressed Sparse Row (CSR) format



File formats


- MatrixMarket
- Rutherford-Boeing
- Harwell Boeing

Direct Solvers

- UMFPACK
 - Single-threaded
 - MatLab's sparse solver
 - UMFPACK v4.0 used in MatLab 6.5
 - UMFPACK v4.3 used in MatLab 7.1
- MUMPS
 - MPI-based multi-frontal sparse solver

A photograph showing the interior of a large, modern warehouse. The space is filled with industrial equipment, including several forklifts and stacks of boxes. Workers in various attire are visible, some standing and others sitting on boxes. The ceiling is high with a complex network of steel beams and numerous bright, industrial lights. The overall atmosphere is one of a busy, organized industrial environment.

Distributed

A photograph of a vast, flat, arid landscape. The ground is dry and dusty, with scattered, sparse vegetation consisting of small, scrubby bushes and a few bare, dead trees. In the far distance, a range of low mountains or hills is visible under a clear, pale sky. The overall scene conveys a sense of isolation and sparse habitation.

Sparse

A project name?

DEL MAR

meagre-crowd

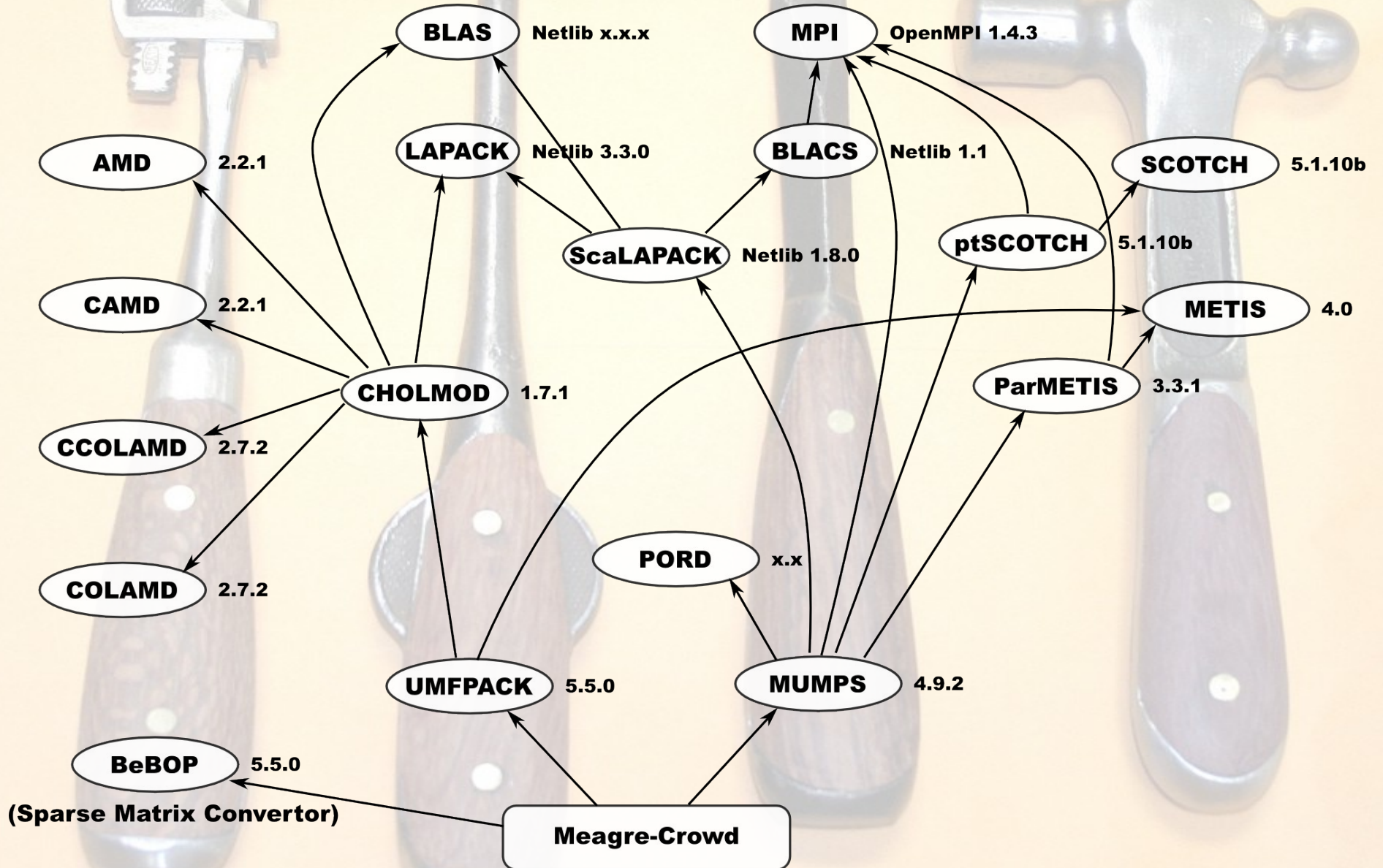
Under the Hood

- Analyze
- Factorize
- Solve

Overhead:

- Initialize: MPI, file I/O, solvers
- Load matrix file

Dependencies



50% of effort here!

side note: Packaging

- Autotools: autoconf, automake, etc.
- Regression tests
- Memory leaks: valgrind/memcheck
- Test coverage: gcov

Method

- Start at bare remote system
- Install dependencies
- Compile, test and install test-harness
- Launch jobs, wait to complete & generate summary of results (.csv)
 - Compare results for various matrices, cores, and solvers

Systems

- Local laptop
 - localhost: 2 cores
 - 1 x 1 socket x 2 cores, Core2 Duo @ 2.66GHz, 3GB memory
- Sharcnet
 - mako.sharcnet.ca: 240 cores
 - 14 x 2 sockets x 4 cores, Xeon @ 3.0GHz, 8GB memory
 - 16 x 2 sockets x 4 cores, Nehalem @ 2.1 GHz, 16GB memory
 - 1h per job run time limit

Example Results

summary.csv:

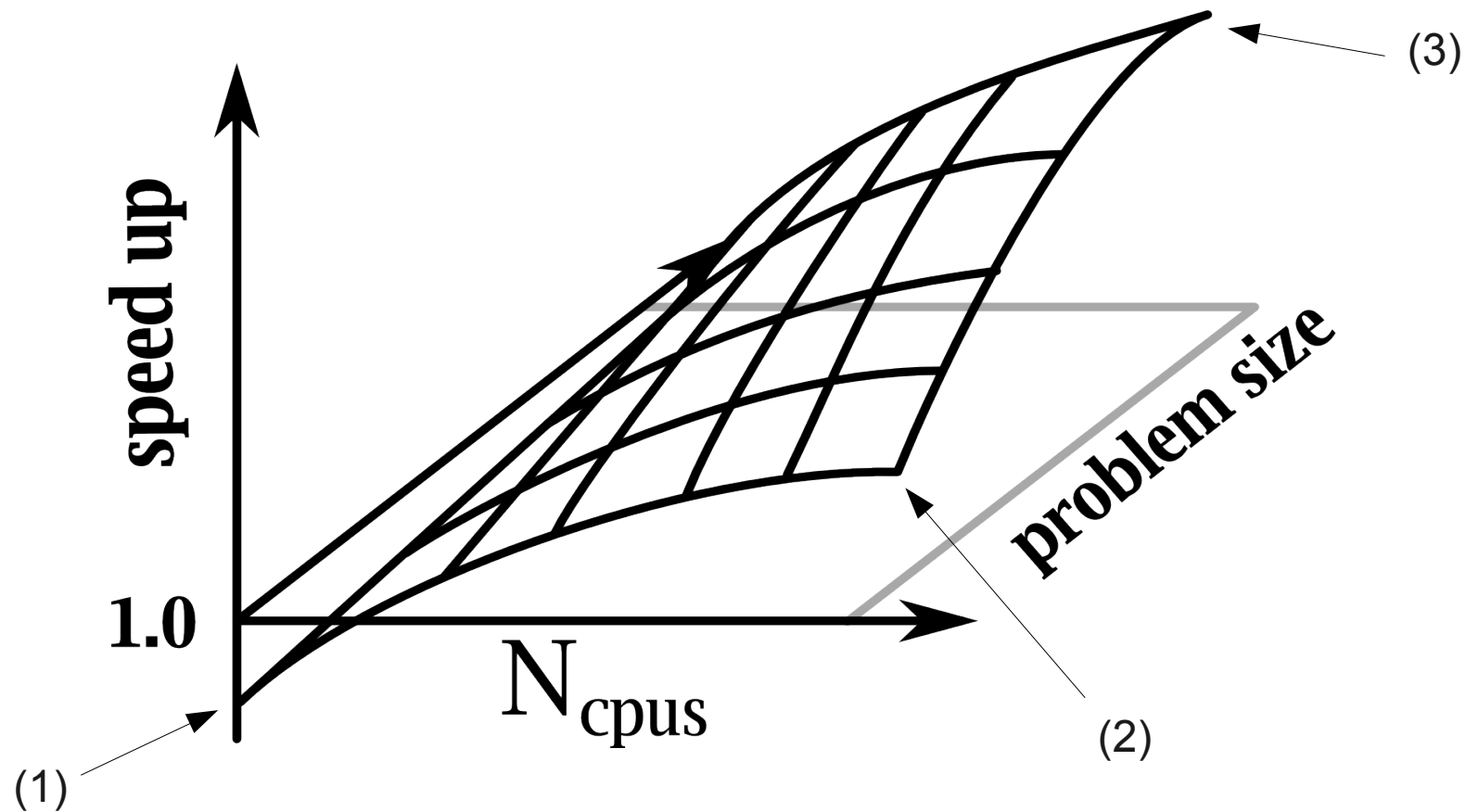
```
solver, cores, threads, matrix, symmetry, type, m, n, nz, analyze, factorize, solve, total (ms)
umfpack, 1, 0, ATandT/onetone2, unsymmetric, real, 36057, 36057, 227628, 46.246, 187.259, 24.384, 257.889
umfpack, 1, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 12350.008, 794.808, 100.267, 13245.083
mumps, 1, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 389.836, 1046.105, 196.017, 1631.958
mumps, 2, 0, ATandT/onetone2, unsymmetric, real, 36057, 36057, 227628, 376.045, 195.486, 20.111, 591.642
mumps, 2, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 458.362, 607.423, 153.085, 1218.870
mumps, 3, 0, ATandT/onetone2, unsymmetric, real, 36057, 36057, 227628, 378.637, 160.970, 17.732, 557.339
mumps, 3, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 505.757, 794.173, 288.290, 1588.220
mumps, 4, 0, ATandT/onetone2, unsymmetric, real, 36057, 36057, 227628, 406.408, 219.242, 41.284, 666.934
mumps, 4, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 559.350, 696.596, 233.179, 1489.125
mumps, 5, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 440.115, 499.976, 123.479, 1063.570
mumps, 6, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 5757.103, 578.390, 170.805, 6506.298
mumps, 7, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 6048.197, 600.729, 172.852, 6821.778
mumps, 8, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 5396.500, 536.353, 173.956, 6106.809
mumps, 16, 0, ATandT/onetone2, unsymmetric, real, 36057, 36057, 227628, 424.216, 208.312, 31.774, 664.302
mumps, 16, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 5345.562, 604.350, 195.516, 6145.428
mumps, 24, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 5333.293, 586.916, 262.442, 6182.651
mumps, 32, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 5331.116, 608.359, 550.088, 6489.563
mumps, 64, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 5365.019, 3656.561, 3282.073, 12303.653
mumps, 128, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 3815.333, 628.596, 27365.360, 31809.289
mumps, 240, 0, ATandT/onetone2, unsymmetric, real, 36057, 36057, 227628, 4838.315, 494.256, 82661.824, 87994.395
mumps, 240, 0, Freescale/transient, unsymmetric, real, 178866, 178866, 961790, 4141.450, 17643.739, 93299.403, 115084.5
```

from a run on mako.sharcnet.ca



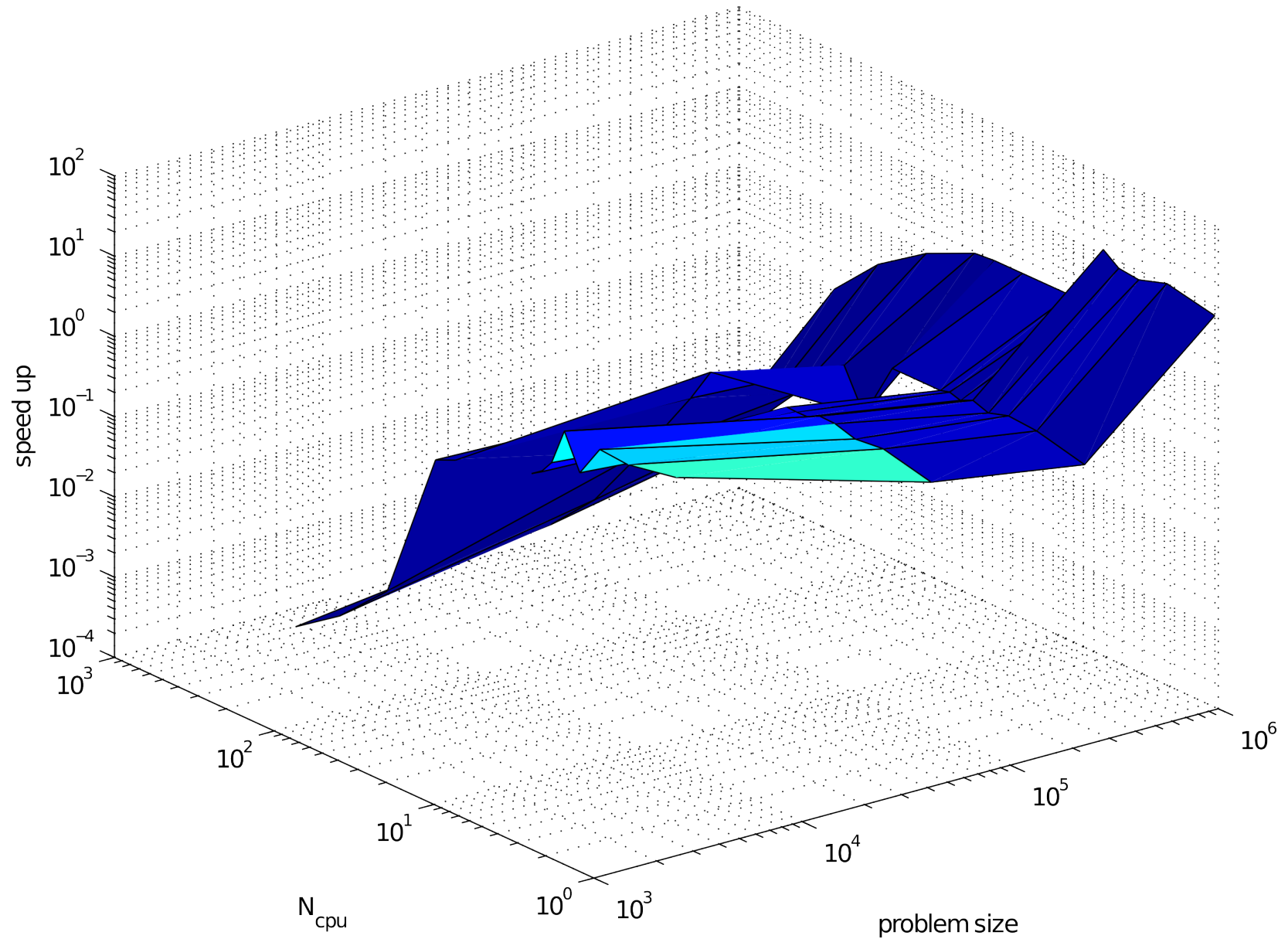
Demo

Wanted to show...

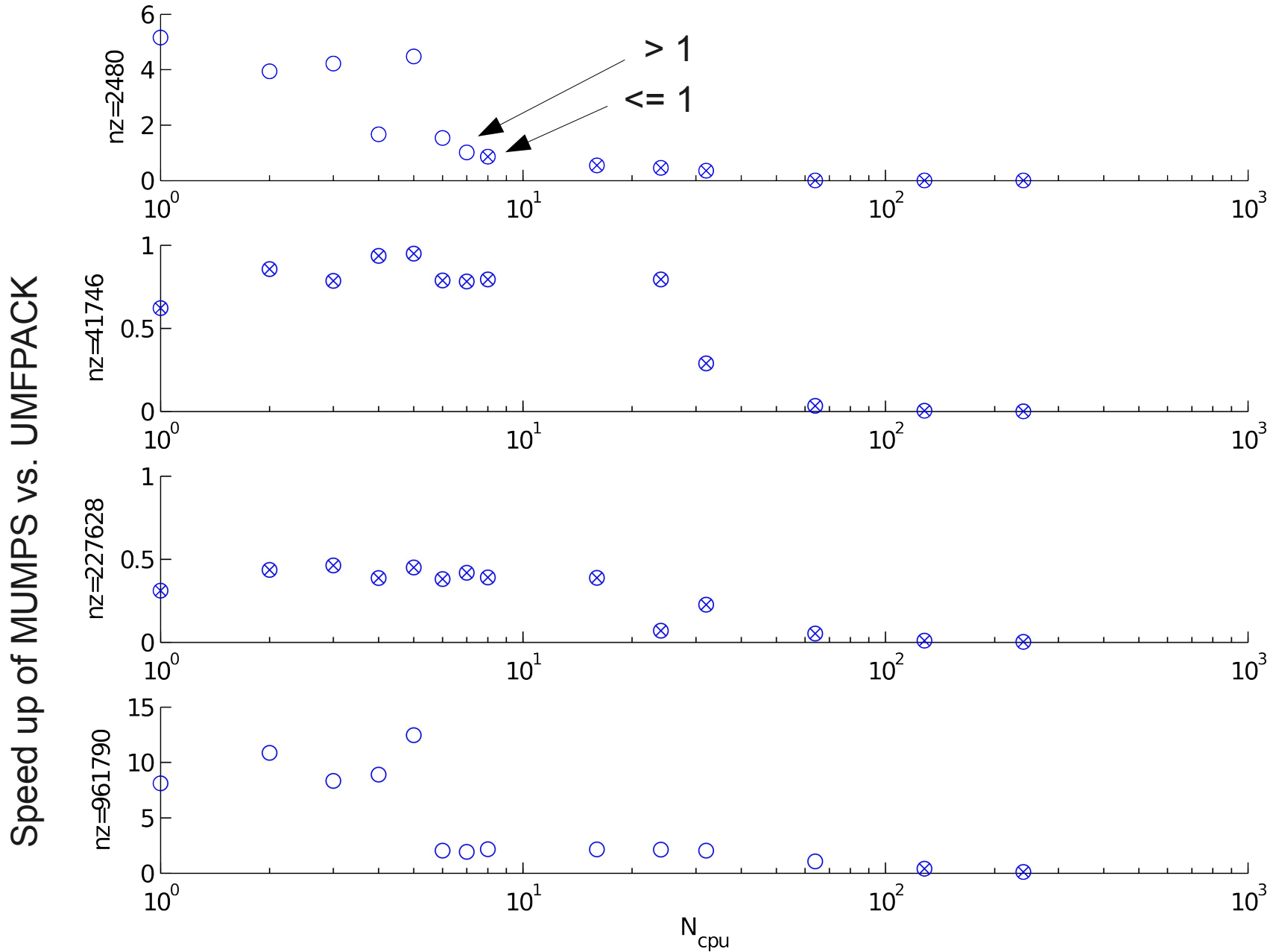


- (1) Communication costs for small problems
- (2) Diminishing returns from additional CPUs
- (3) Increasing pay-off for larger problems

Actual Results



Actual Results



the Good, the Bad, the Ugly

A silhouette of a person in a dynamic, athletic pose, possibly a dancer or performer, against a bright, glowing background. The person is in a wide stance with arms raised, one hand holding a long, thin object like a stick or a sword. The background is a gradient of warm colors, from yellow to orange, with a bright light source behind the person, creating a halo effect.

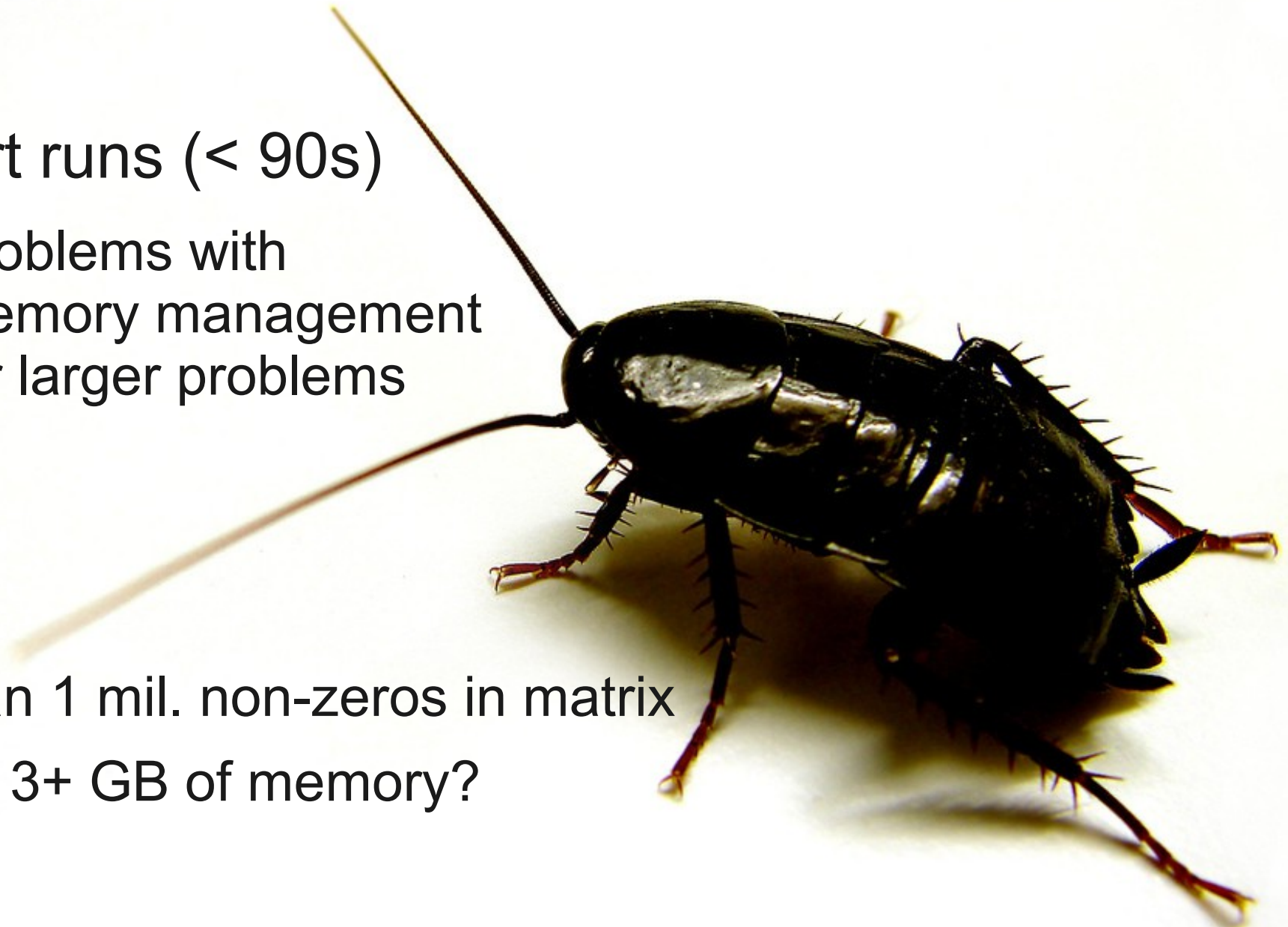
- Good
 - Works, got results
 - Solid foundation
- Bad
 - Poor packaging of dependencies
 - System limitations
- Ugly
 - Memory management in dependencies
 - file loader, format conversion and Metis

Problems

Short runs (< 90s)

- Problems with memory management for larger problems

More than 1 mil. non-zeros in matrix
... 3+ GB of memory?



Carrying On

- Other solvers
 - Heterogeneous (MPI), homogeneous (OpenMP), and single threaded solvers
- Handle other matrix types
 - Complex, symmetric, Hermitian
- Improve file handling
 - Load MatLab, Harwell-Boeing, Rutherford-Boeing
- Try other compute resources
(Thanks sharcnet!)

Questions?



github
SOCIAL CODING

Source code available at
<http://github.com/boyle/meagre-crowd>