Multicore Optimization for Synergia2

Qiming Lu
James Amundson
Fermilab

Draft

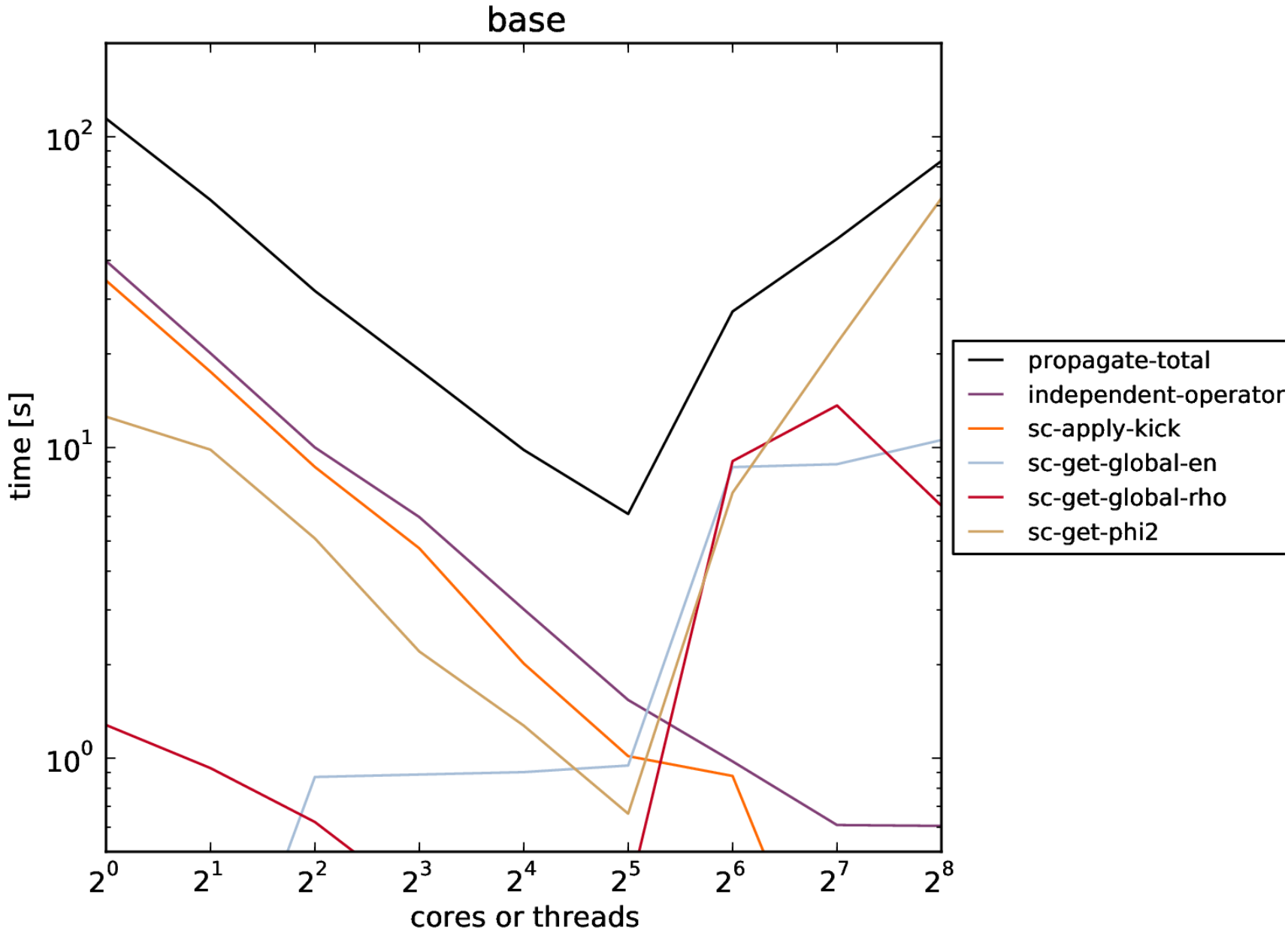
Starting point

- Benchmark
  - Four FODO cells
    - 8 steps/cell
    - 2nd order maps
  - Space charge
    - 32x32x256 grid
  - 2621440 particles
- Difficult scenario
  - Easily scaling portion of calculation smaller than in real-world calculations

Machine

- Fermilab's Wilson cluster
  - 32-core AMD nodes
  - Infiniband

Base component breakdown



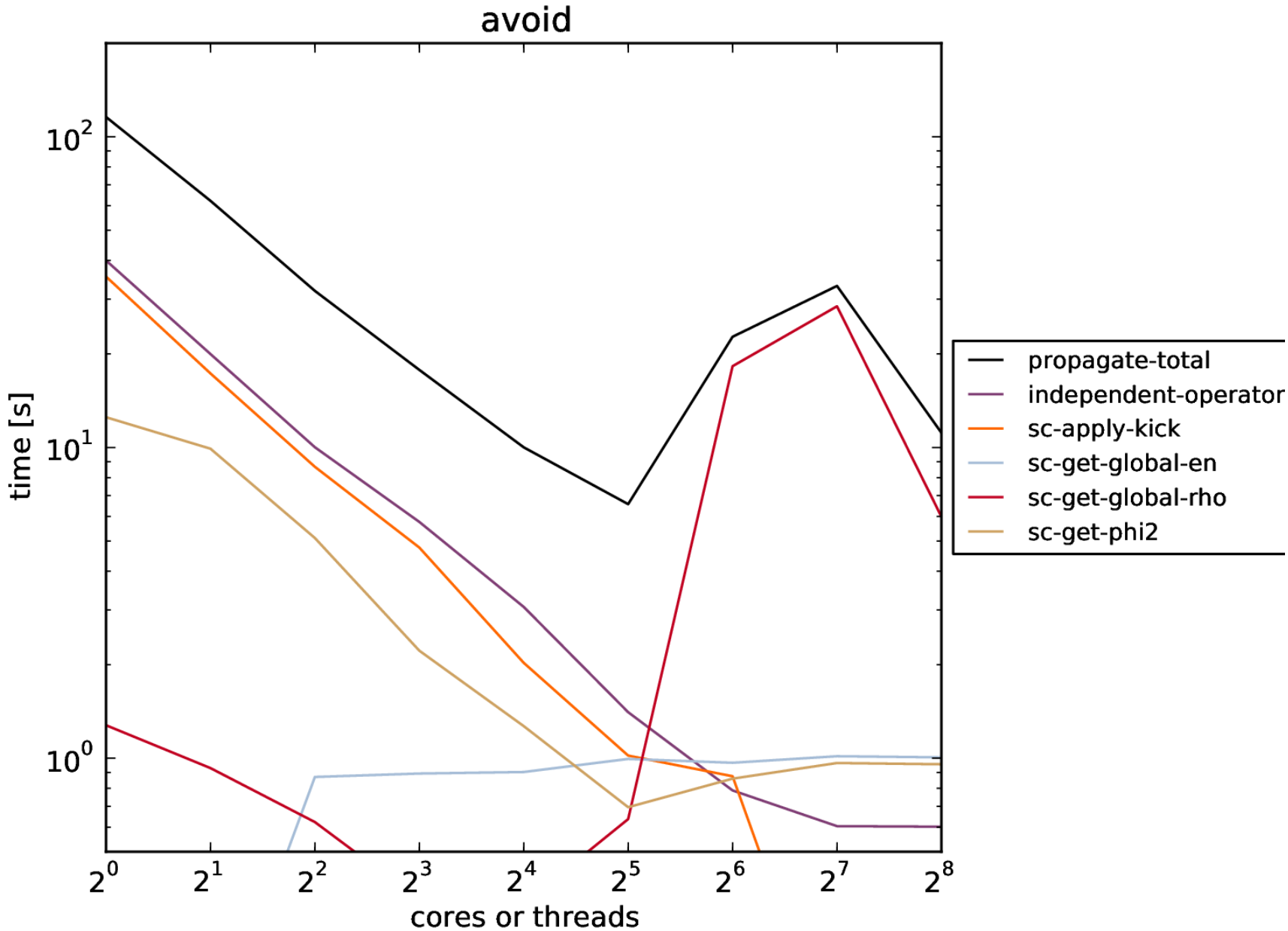
Base

- Labels
  - Sc-get-global-* : communication steps
  - Sc-get-phi2: field solve (dominated by FFTs)
  - Sc-apply-kick, independent-operator-apply: independent-particle operations
- Terrible scaling beyond single node
  - FFT limiting factor
  - Communication steps nearly as bad

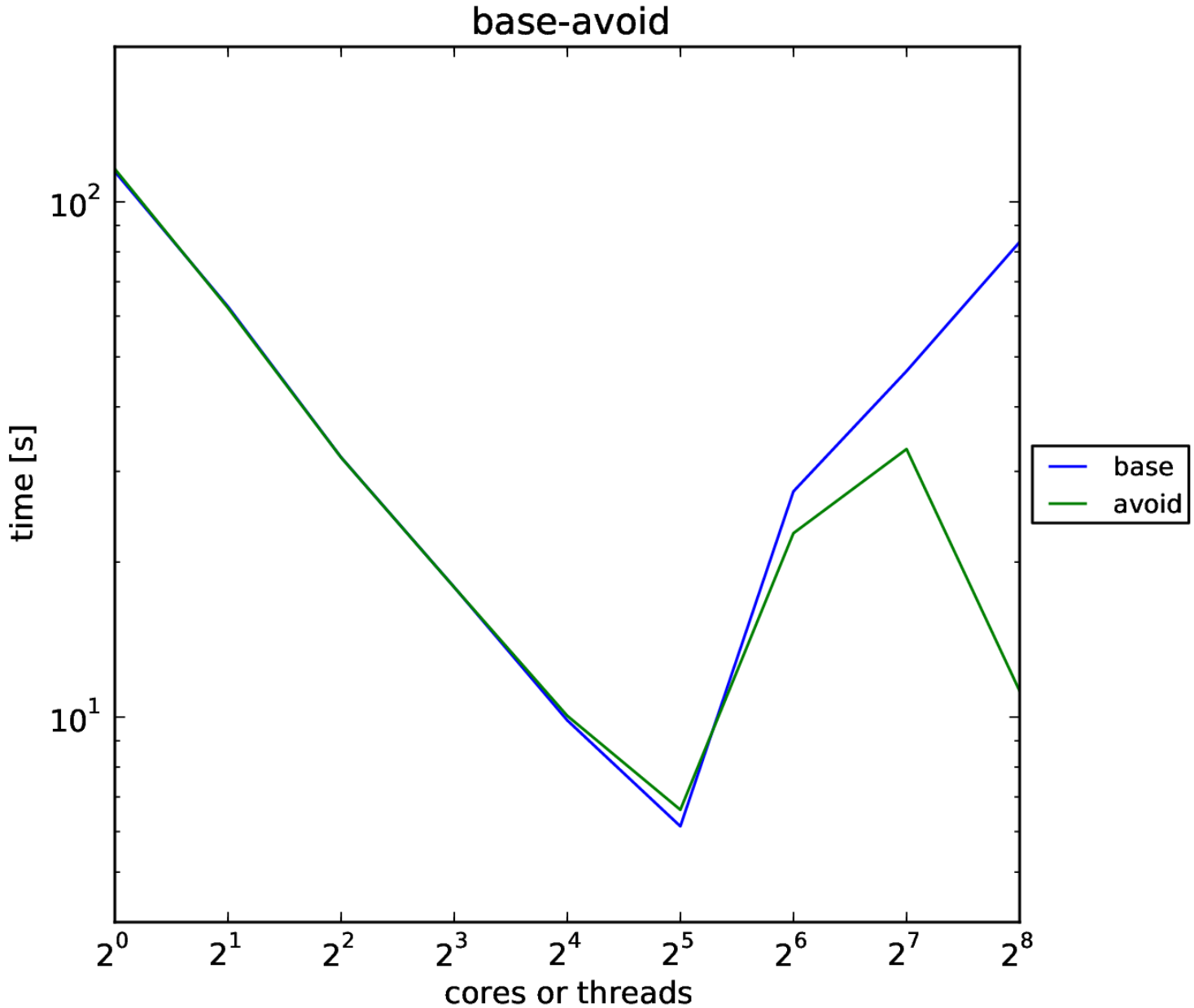
Next step: communication avoidance

- Communication avoidance: perform extra computations in order to avoid communication
- In our case:
  - Perform FFT on each node (32 cores)
    - Avoids having to broadcast fields across nodes (sc-get-global-en)
    - Also avoids poor multi-node scaling of FFT

Avoid component breakdown



Base-avoid comparison



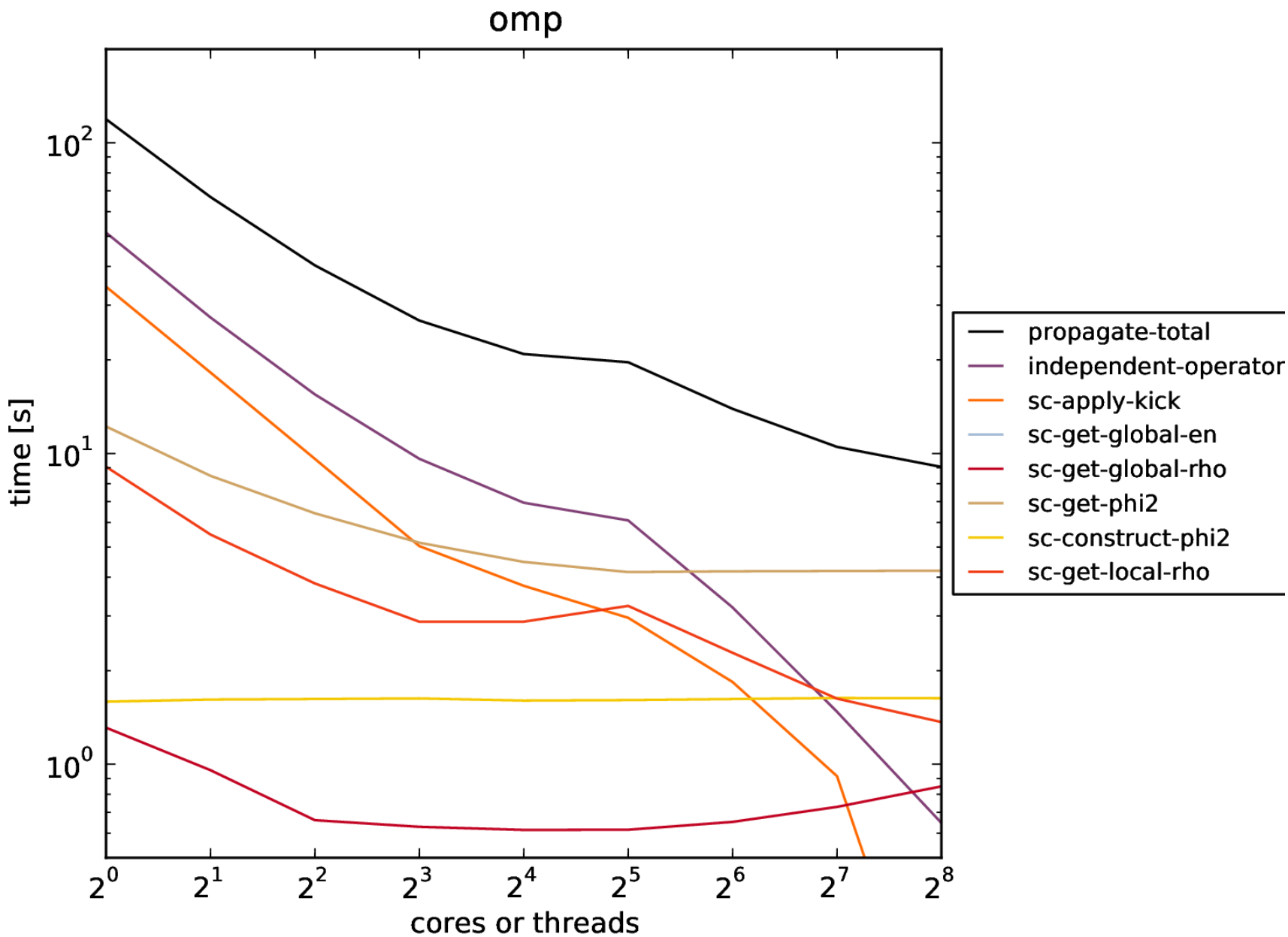
Some progress

- Scaling better, but still terrible
- FFT and field broadcast no longer dominate
- Field collection communication (sc-get-global-rho) now dominates

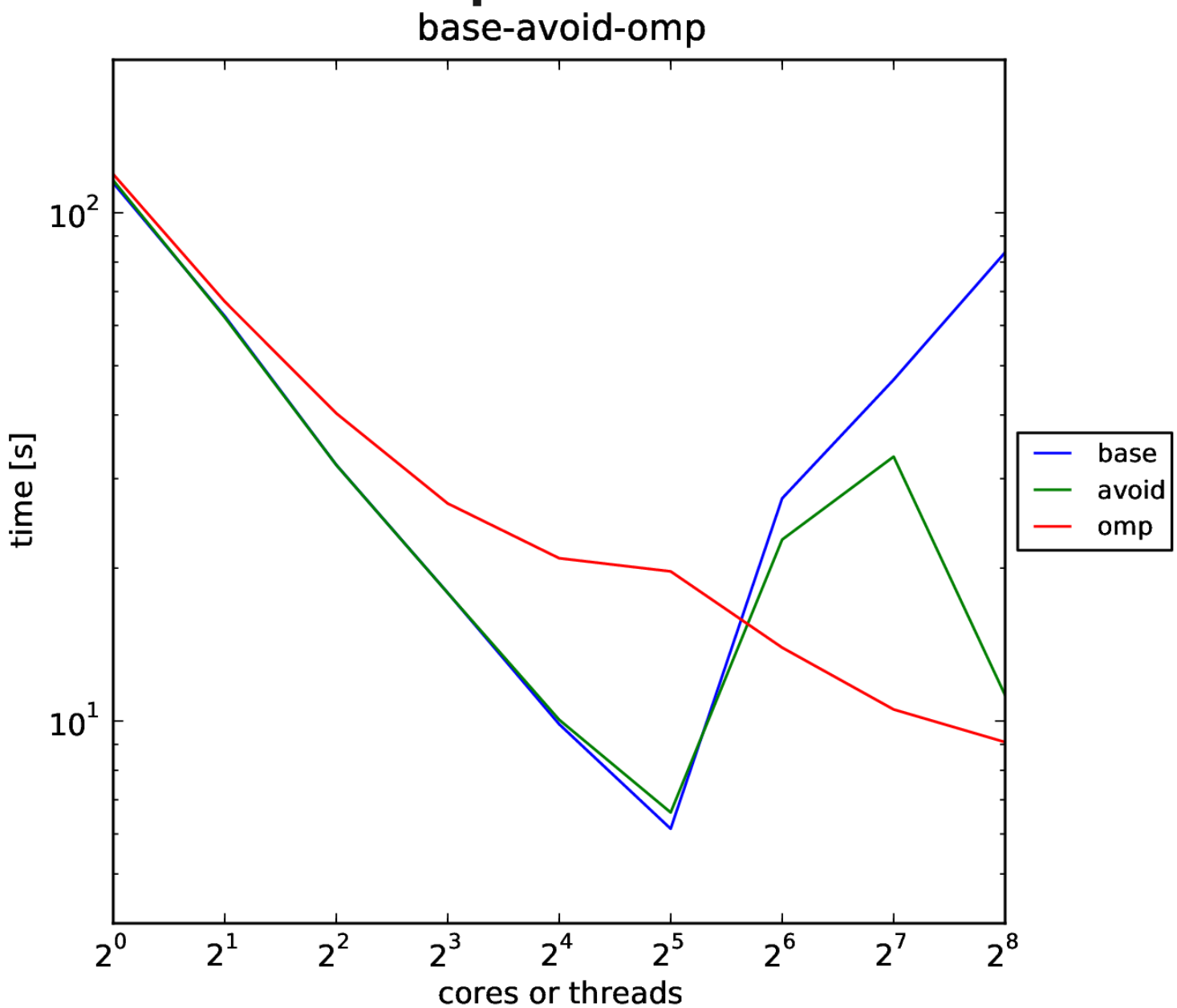
Another step: OpenMP

- details

OpenMP component breakdown



Comparison so far



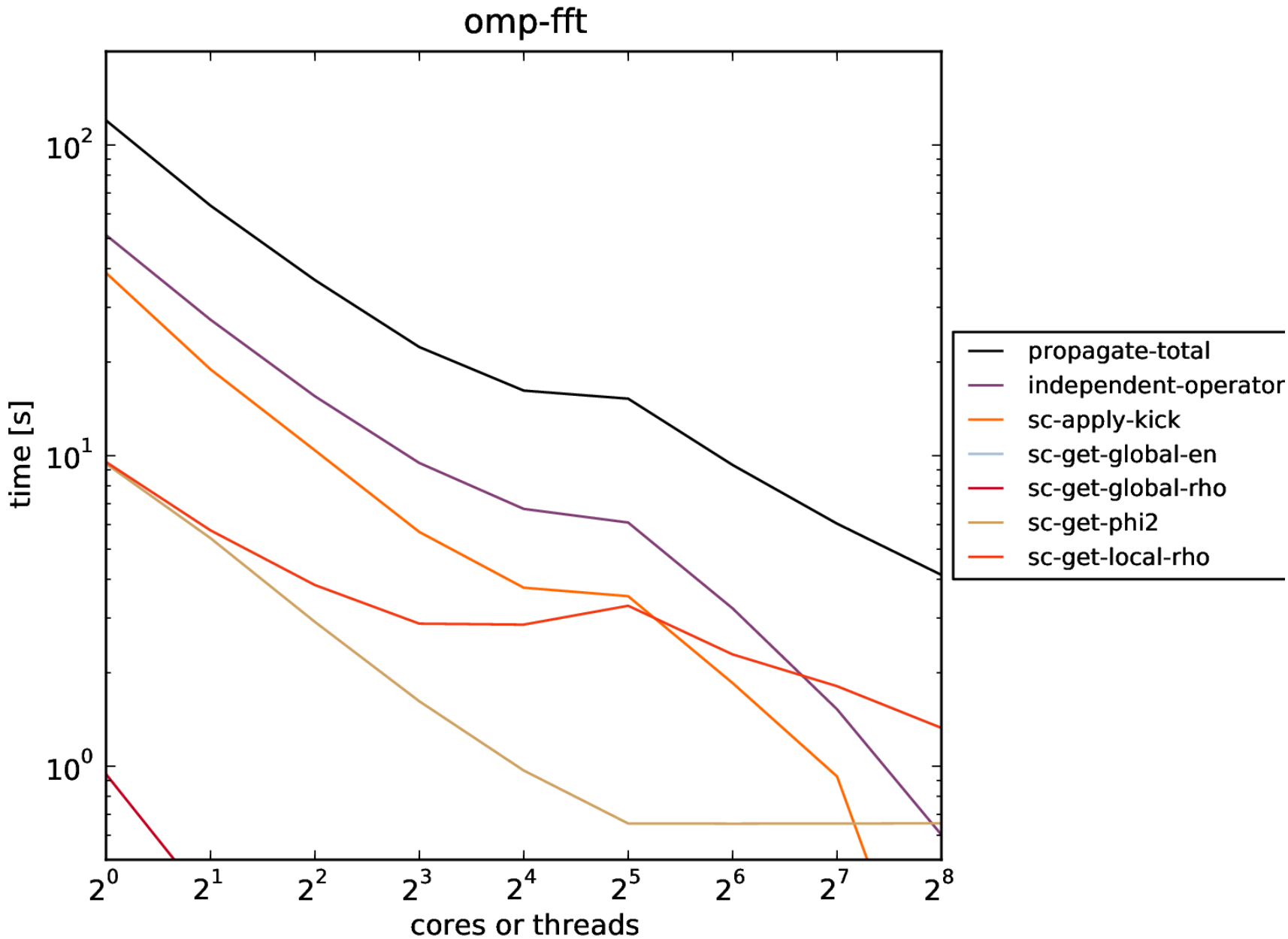
OpenMP results

- Scaling greatly improved
- Peak performance inferior to pure MPI implementation
- Performance dominated by FFT

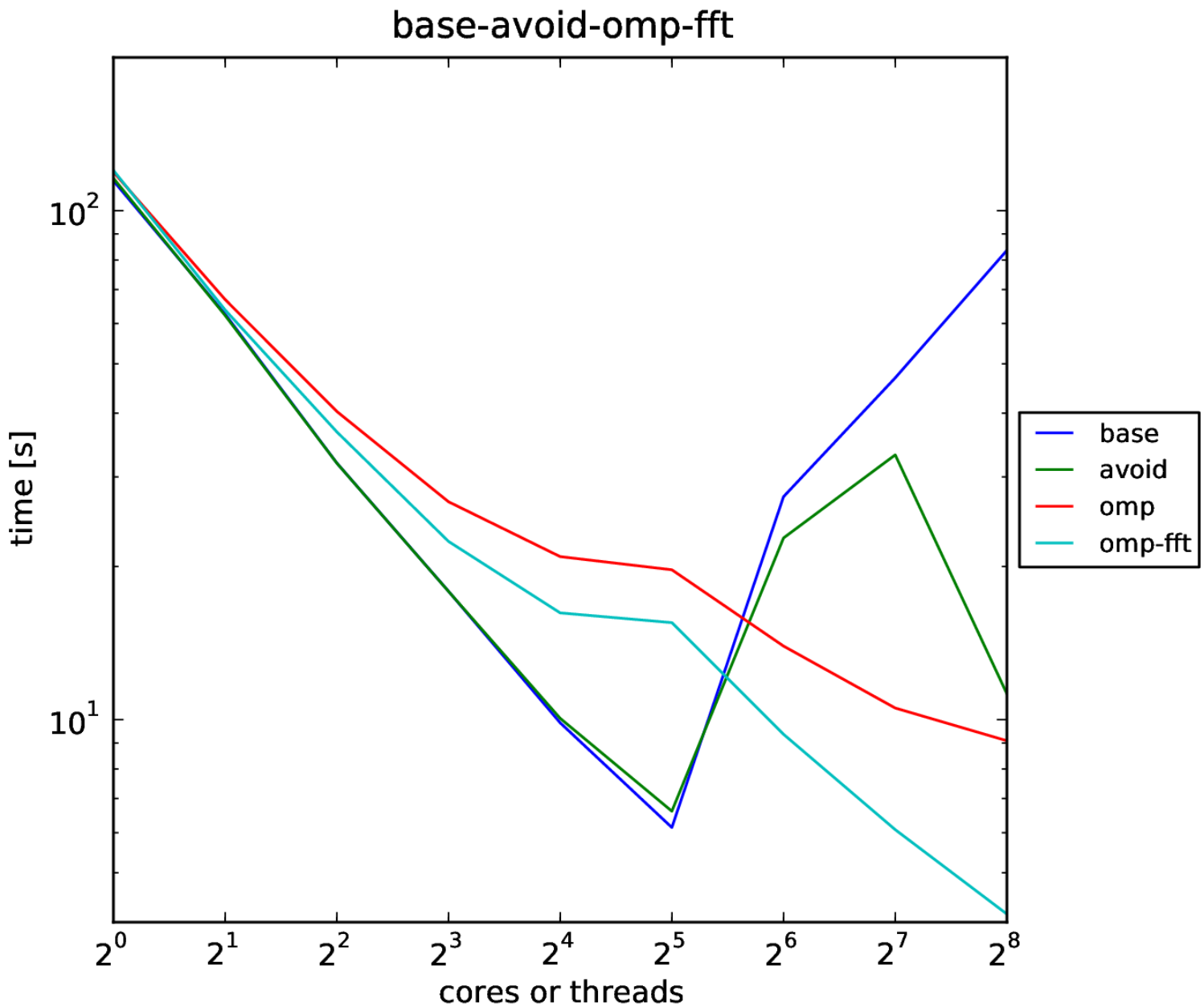
OpenMP-FFT

- details

OpenMP-FFT component breakdown



Comparison


base-avoid-omp-fft
time [s]
10^2
10^1
2^0
2^1
2^2
2^3
2^4
2^5
2^6
2^7
2^8
cores or threads
base
avoid
omp
omp-fft

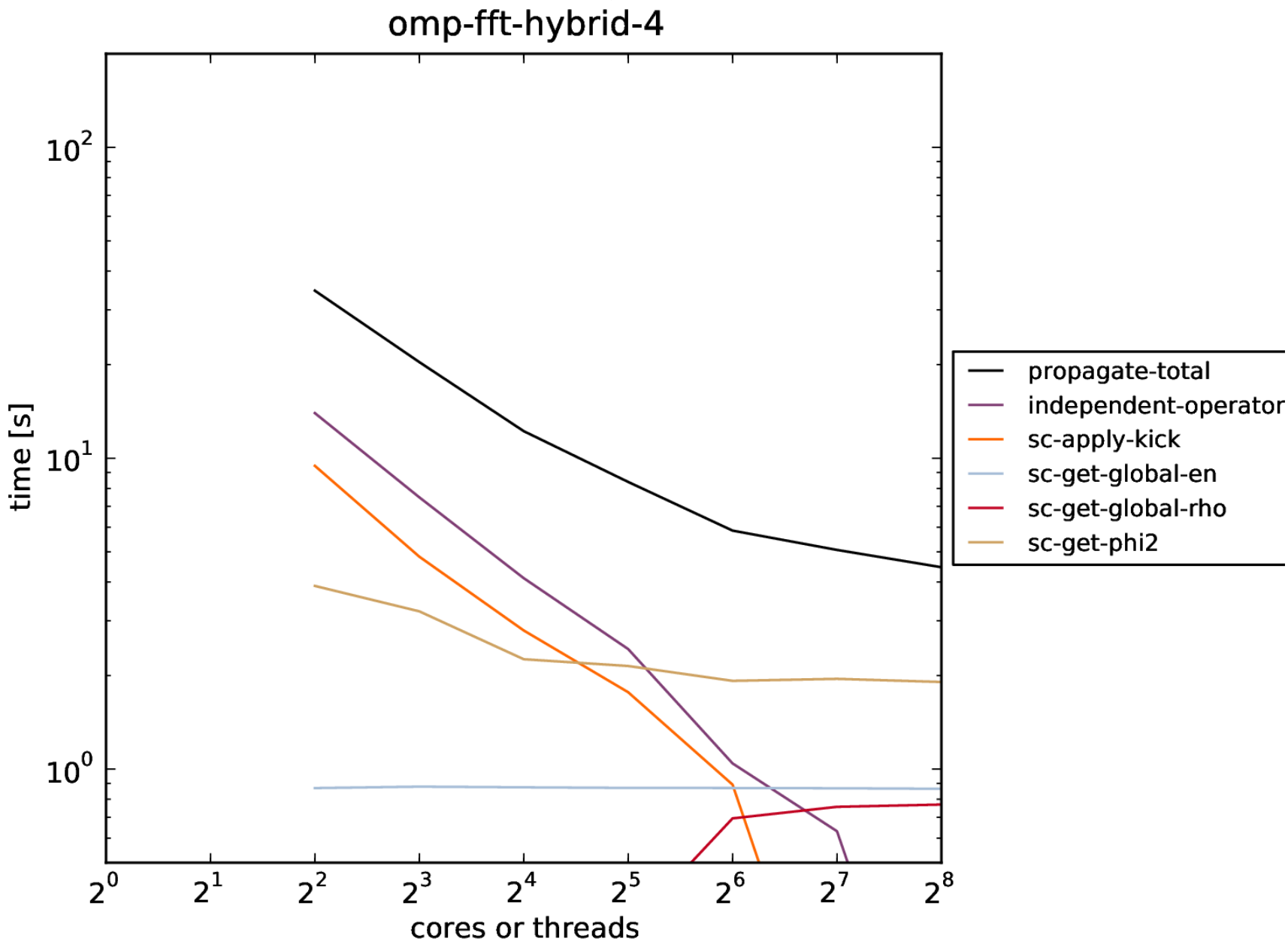
OpenMP-FFT results

- Best scaling so far
- Best peak performance

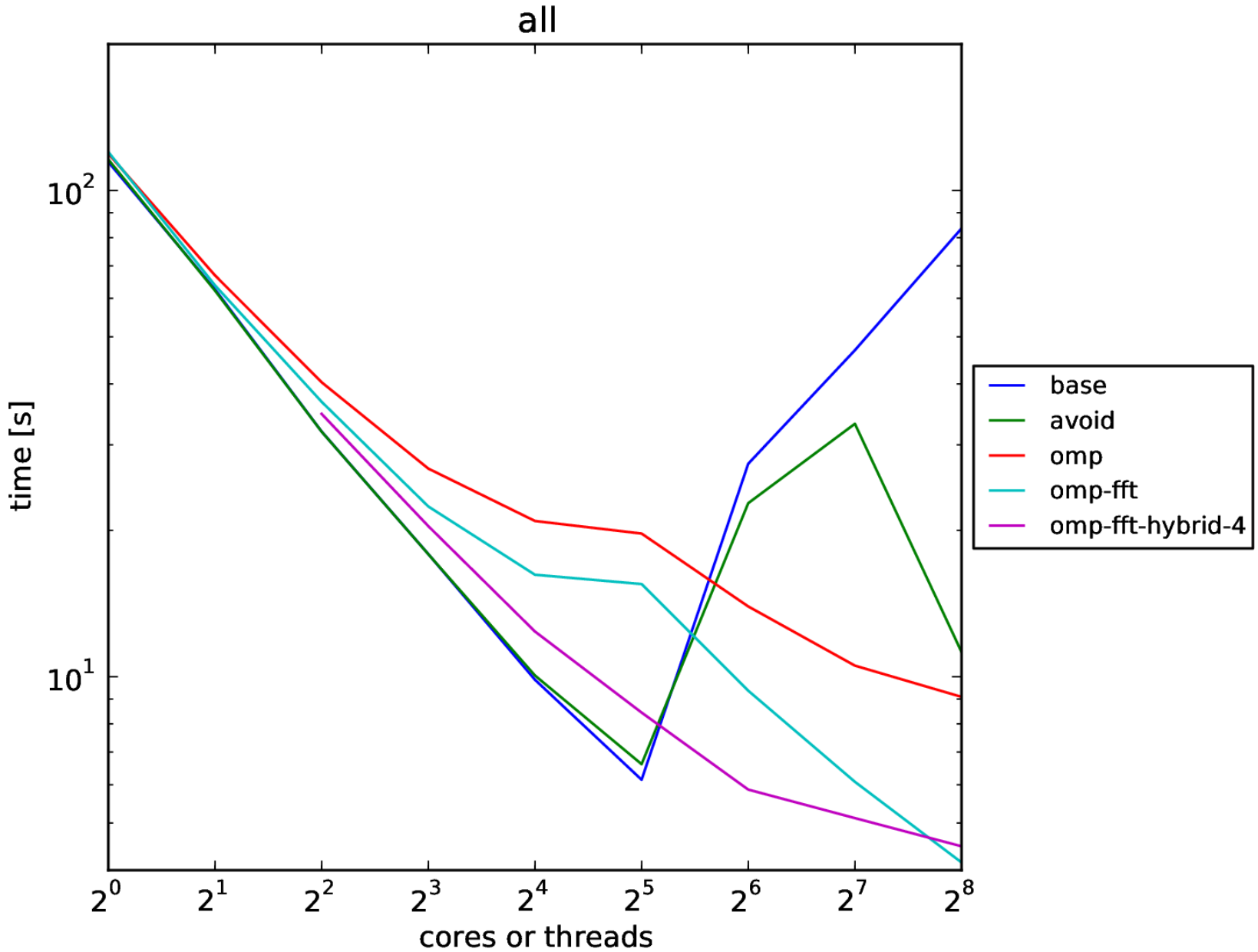
OpenMP-FFT-Hybrid

- Details
- Using four threads in these results

OpenMP-FFT-Hybrid breakdown



Final comparison



Discussion

- OpenMP-FFT-Hybrid has better performance than the non-hybrid case, but somewhat worse scaling behavior
- Expect OpenMP-FFT-Hybrid to have the best overall performance on architectures with fewer cores/node (?)