Parallel Conjugate Gradient Solver

Dmitry K. Kolmogorov, Niels N. Sørensen
Wen Z. Shen and Jens N. Sørensen

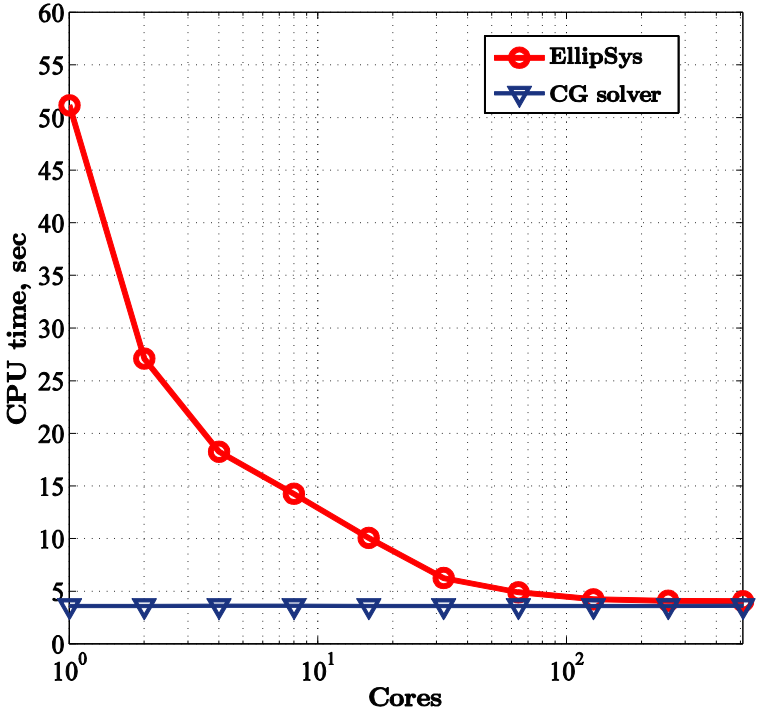
DTU, Wind Energy

August 26, 2014

Outline

- Problem statement
- Parallel conjugate gradient solver
- Optimization techniques
- EllipSys performance enhancement
- Conclusions

Speed Up



- Lid-driven cavity flow, Re=100
- Grid ~ **1 million cells** with
- **32x32** blocks

  32x32 cells per block

EllipSys flow solver

Momentum equations

PC equation

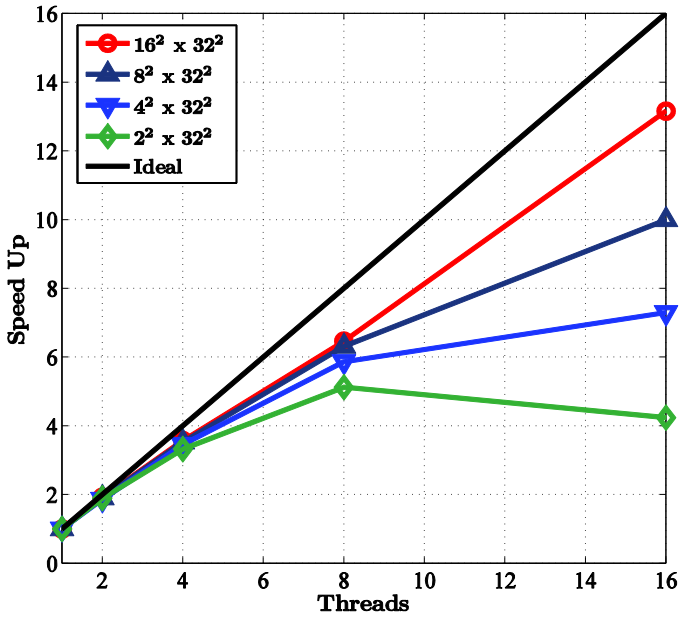
MG solver

Relaxation

Restriction/Prolongation

Coarse Grid solver

Parallel CG solver



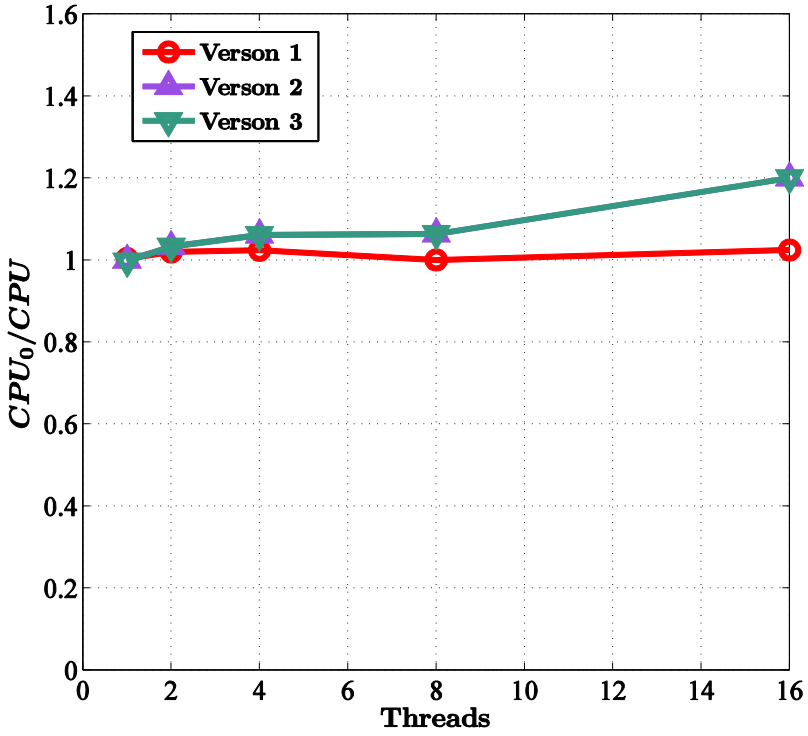
Speed Up of the CG solver

CG Solver enabled on different grid levels of MG solver:

Level	Cells per block
2	16x16
3	8x8
4	4x4
5	2x2

Grid **32x32 blocks** with **32x32** cells per block

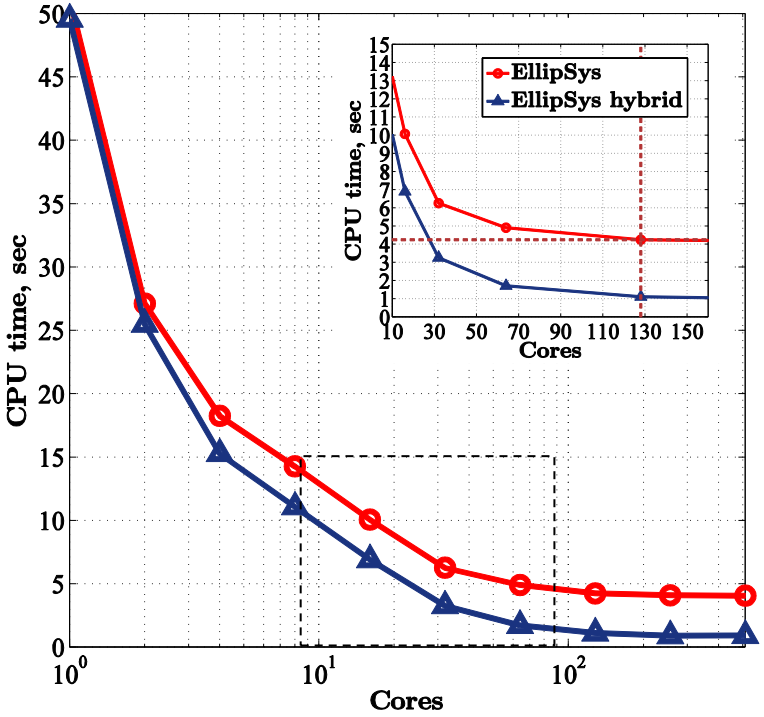
Optimization of the Parallel CG solver

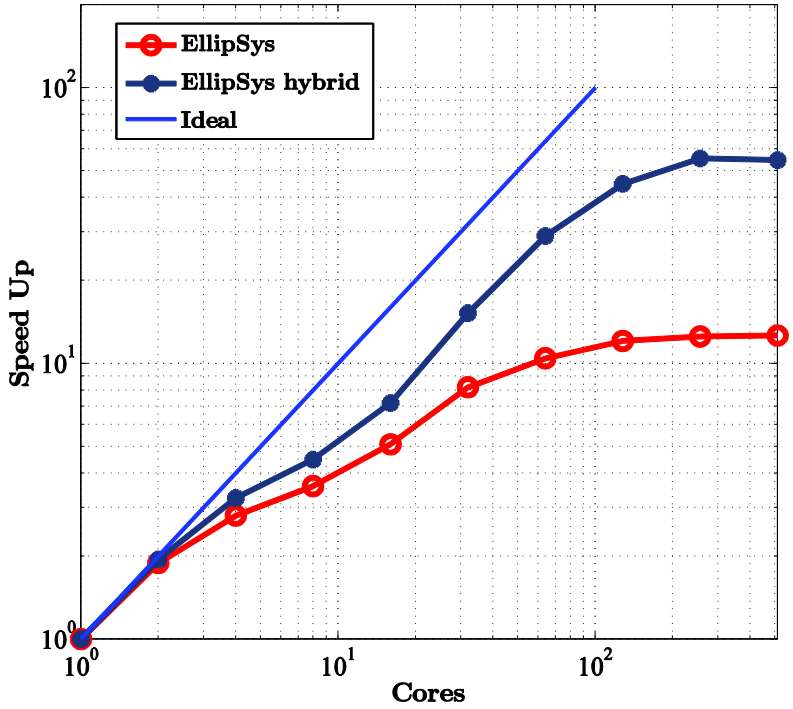


- V0: Fine grain parallelism
- V1: Coarse grain parallelism (threads created outside of loops)
- V2: With decreased # of barriers (some variables transformed to private)
- V3: Explicit loop partitioning + reduction replaced by atomic call

20 % increase of efficiency compared to straightforward implementation

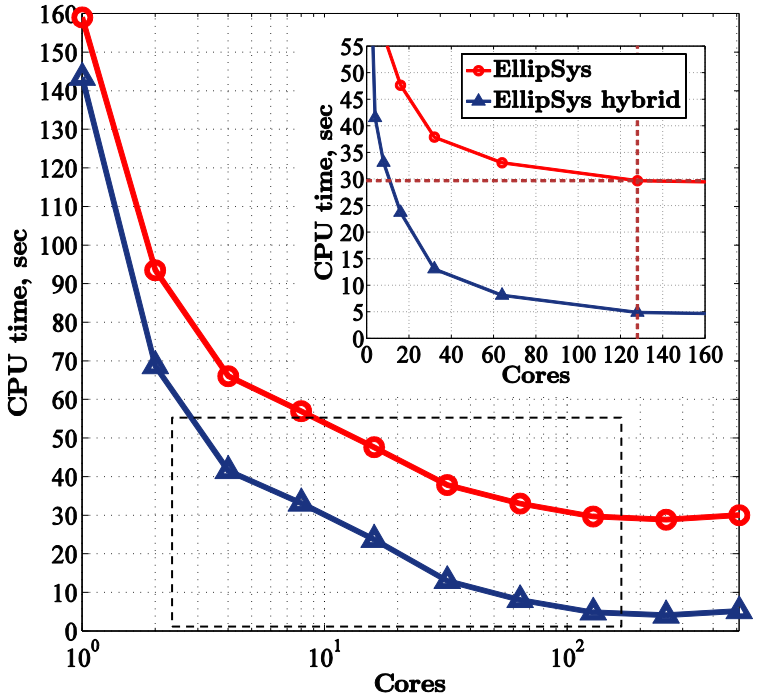
EllipSys performance

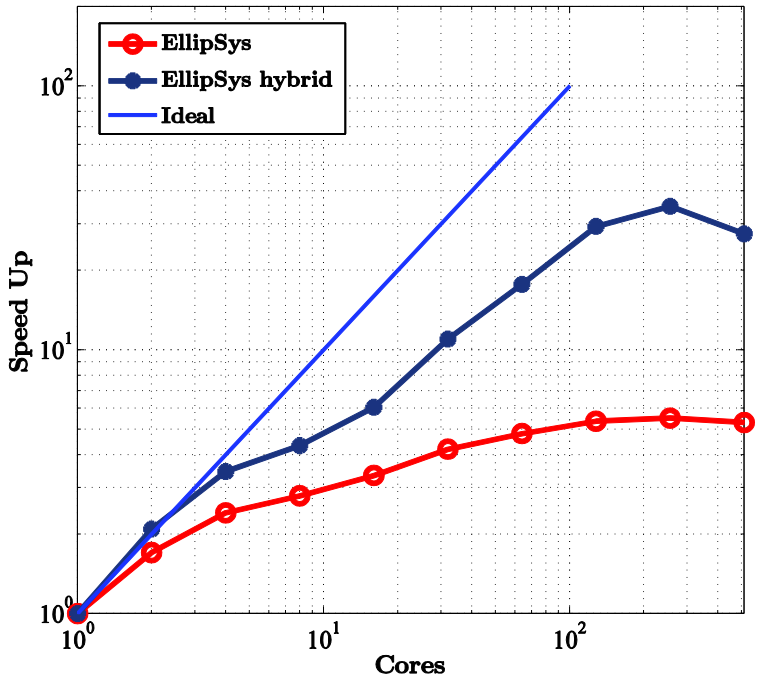




Grid ~ **1 million cells: 32x32** blocks with **32x32** cells per block

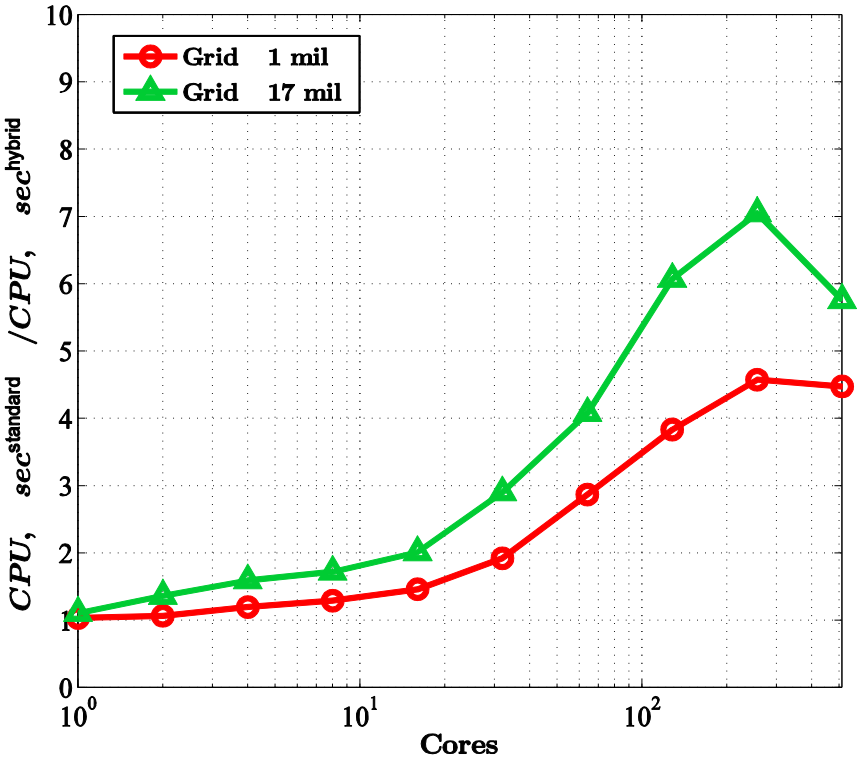
EllipSys performance





Grid ~ **17 million cells: 64x64** blocks with **64x64** cells per block

EllipSys performance



- Grid ~ **1 million cells** with
- **32x32** blocks

  32x32 cells per block

- Grid ~ **17 million cells** with
- **64x64** blocks

  64x64 cells per block

Conclusions

- EllipSys flow solver is known to be originally parallelized on machines with distributed memory architecture, where it has recommended it's high efficiency.
- Hybrid distributed/shared machine memory architecture with multiple number of cores per node becomes widespread today.
- Conjugate gradient solver of EllipSys flow solver was parallelized using shared memory processing based on OpenMP directives.
- The implemented parallel solver extends EllipSys code to state of the art architectures and substantially decreases the required CPU time and the required computational resources.