Speeding up the AWImager

Sebastiaan van der Tol, Cyril Tasse,
Joris van Zwieten, Ger van Diepen

Current developments AWImager

- W-Stack algorithm
  - Implemented by Cyril Tasse
- Parallelize over nodes
  - Distribute the (de)gridding over nodes
- Port code to GPU
  - Standard libraries (FFT)
  - Port code to GPU
  - Optimize code
  - John Romein wrote optimized GPU gridder

Porting AWImager to python

- Development of high level algorithms is faster in Python
- Porting to GPU might require algorithmic changes
- Basic imaging functionality has been ported to python (Joris van Zwieten)
- New functionality: distribution of (de)gridding over nodes. Implementation in only 200 lines of python code.

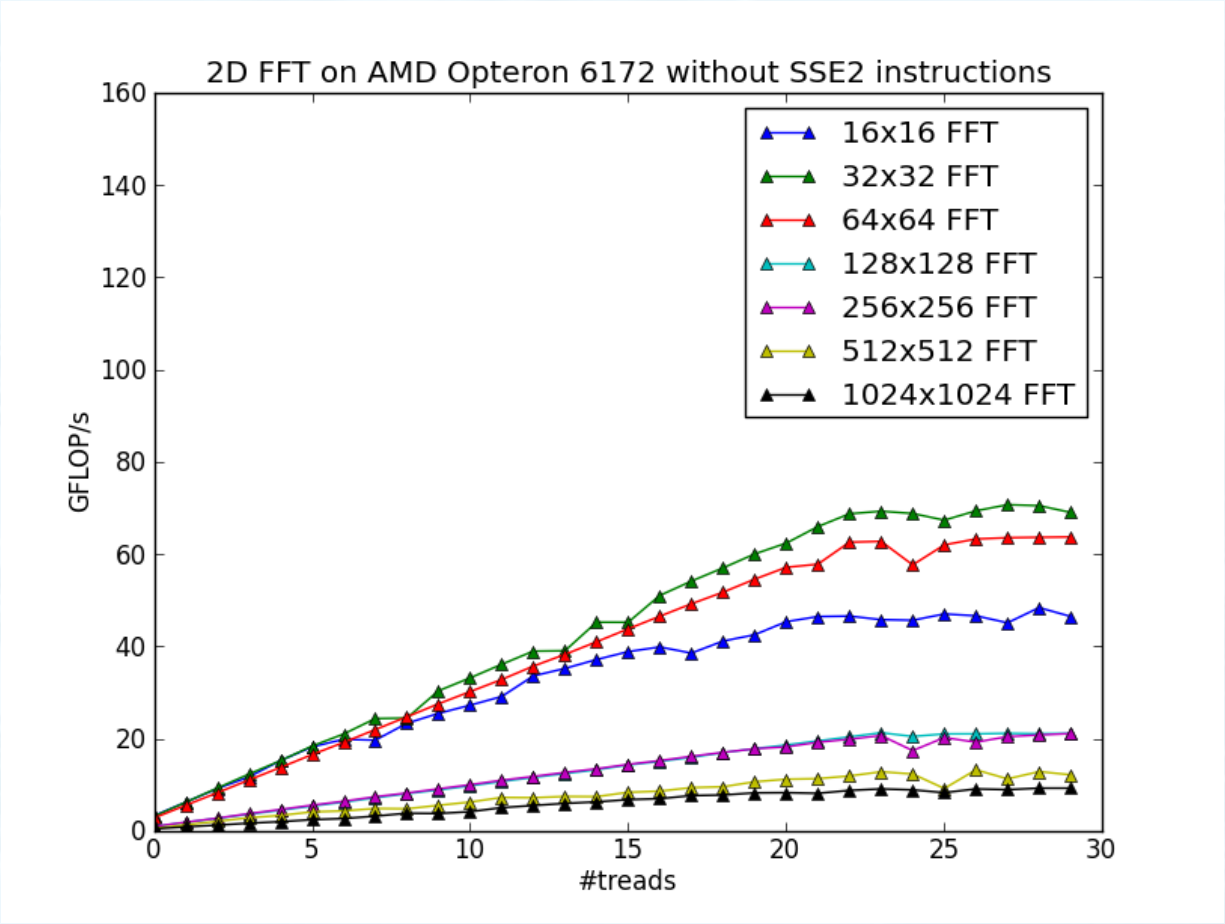
GPU tests

- NVS 4200M
  - 48 cores
  - 1.48 GHz
  - 71 GFLOP/s
- GTX NVIDIA GeForce 680
  - Costs about € 500
  - 1536 cores
  - 1.058 GHz
  - 1545 GFLOP/s

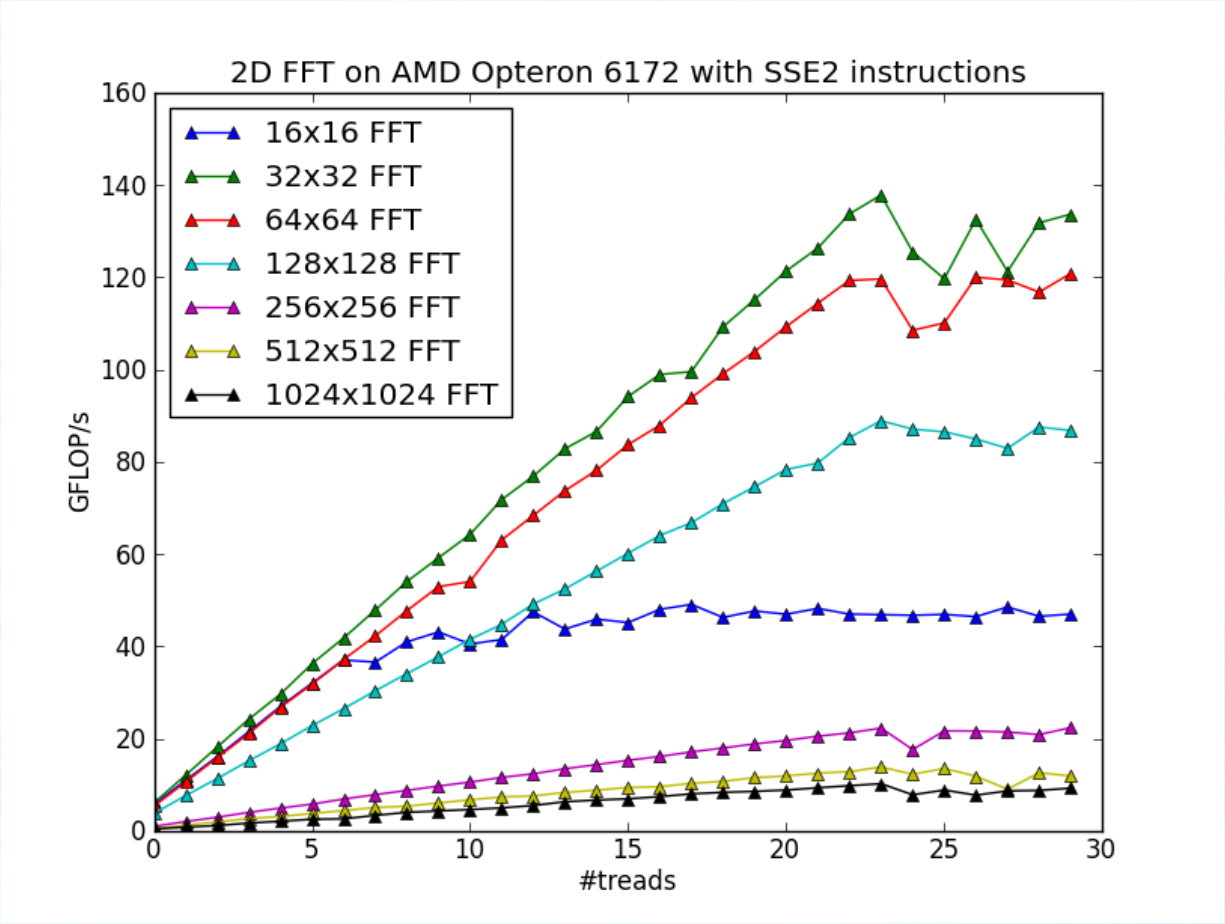
CPU

- CEP II : per node 2 x AMD Opteron 6172
  - 12 cores @ 2.1 GHz
  - 3 instruction per cycle
  - SSE2: 4 single or 2 double precision floating point operations per instruction
  - 4 x 12 x 3 x 2 x 2.1 GHz = 604 GFLOP/s

CPU test on CEP II (1)



CPU test on CEP II (2)



Conclusions

- High level code of AWImager will be in Python
- Distribution over nodes will be supported
- Rough estimate: a €500 GPU can be as fast as 2 CEP II nodes
- AWImager will be able to use GPU