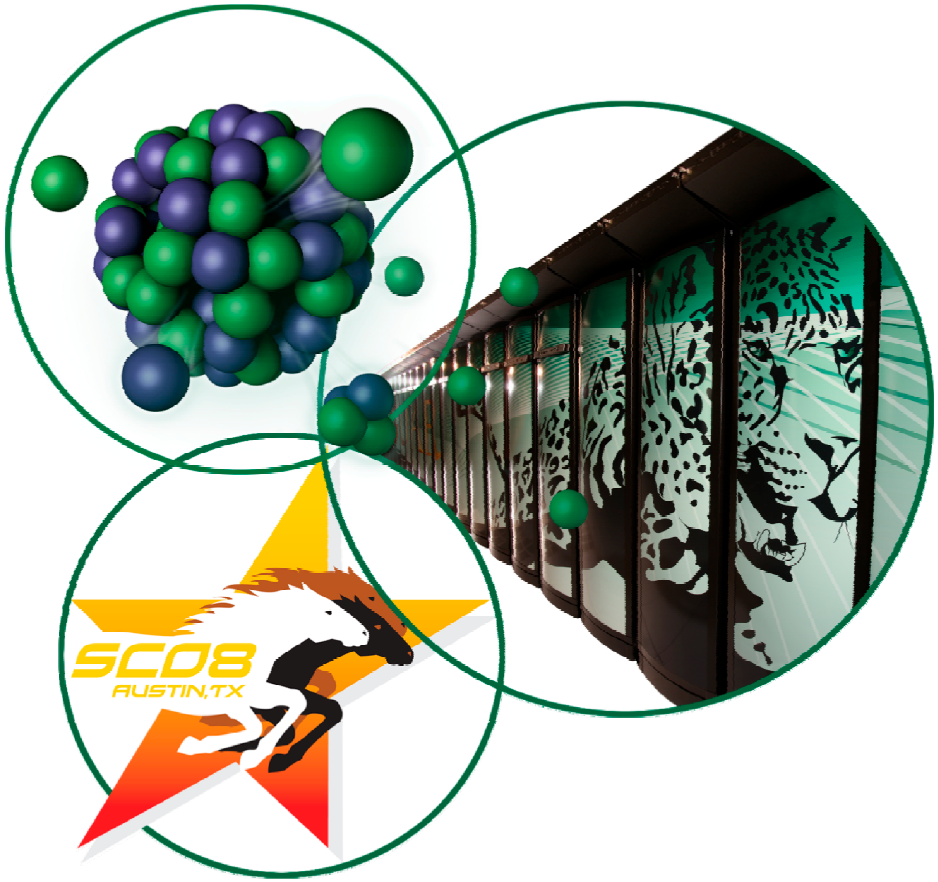
PLASMA
Parallel Linear Algebra Software for Multiprocessor Architectures



Presented by

The Innovative Computing Laboratory

University of Tennessee–Knoxville
Oak Ridge National Laboratory

ICL UT
Innovative Computing Laboratory

The University of Tennessee UT

Oak Ridge National Laboratory

PLASMA objectives

- Parallel performance
  - High utilization of each core
  - Scaling to large numbers of cores
- Any memory model
  - Shared memory
    - Symmetric
    - Nonsymmetric
  - Distributed
  - “Scratchpad” (CELL)
  - GPUs

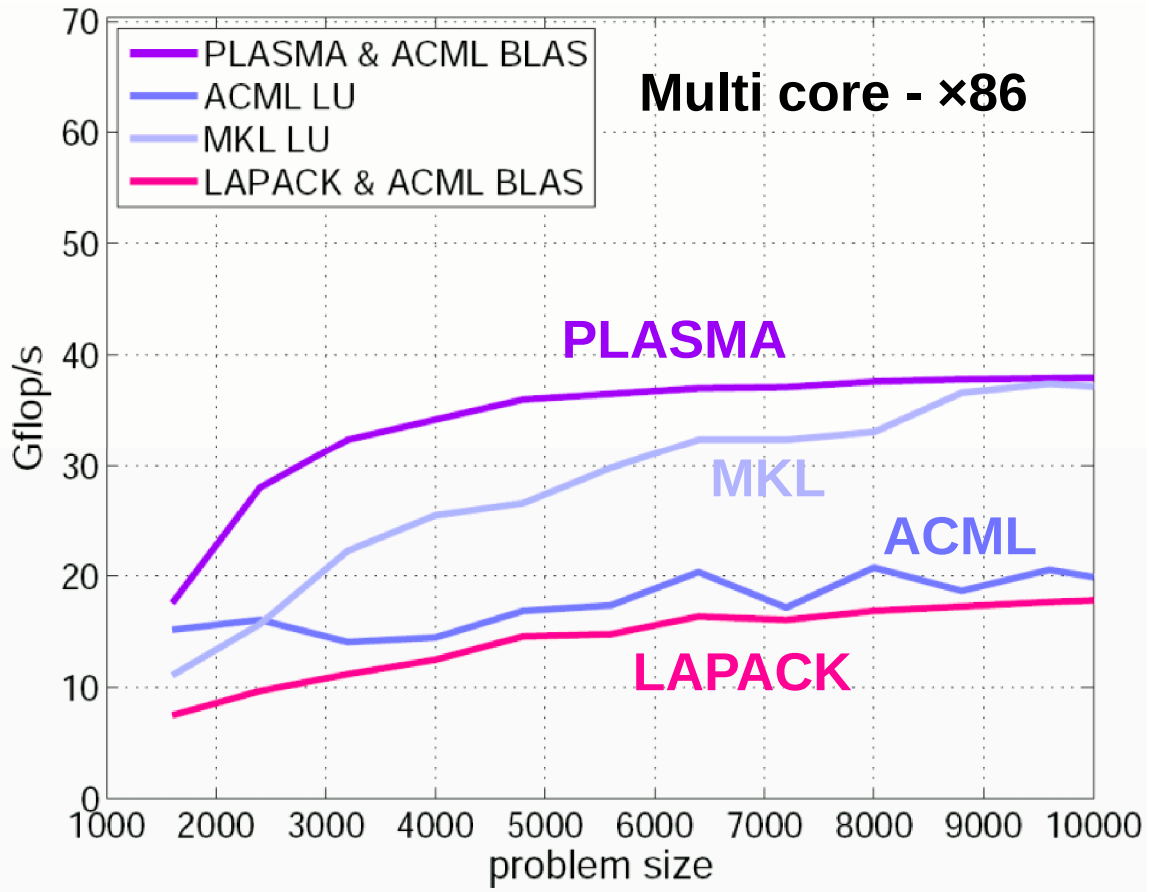
PLASMA methodology

- Data-flow model
- DAG scheduling
- Explicit parallelism
- Implicit communication
- Source-to-source compilation + run time
- Data-flow visualization
- DAG visualization
- Profiling
- Tracing

ICL UT Innovative Computing Laboratory
The University of Tennessee UT
Oak Ridge National Laboratory

Provide highest performance

LU—octa-socket, dual-core Opteron



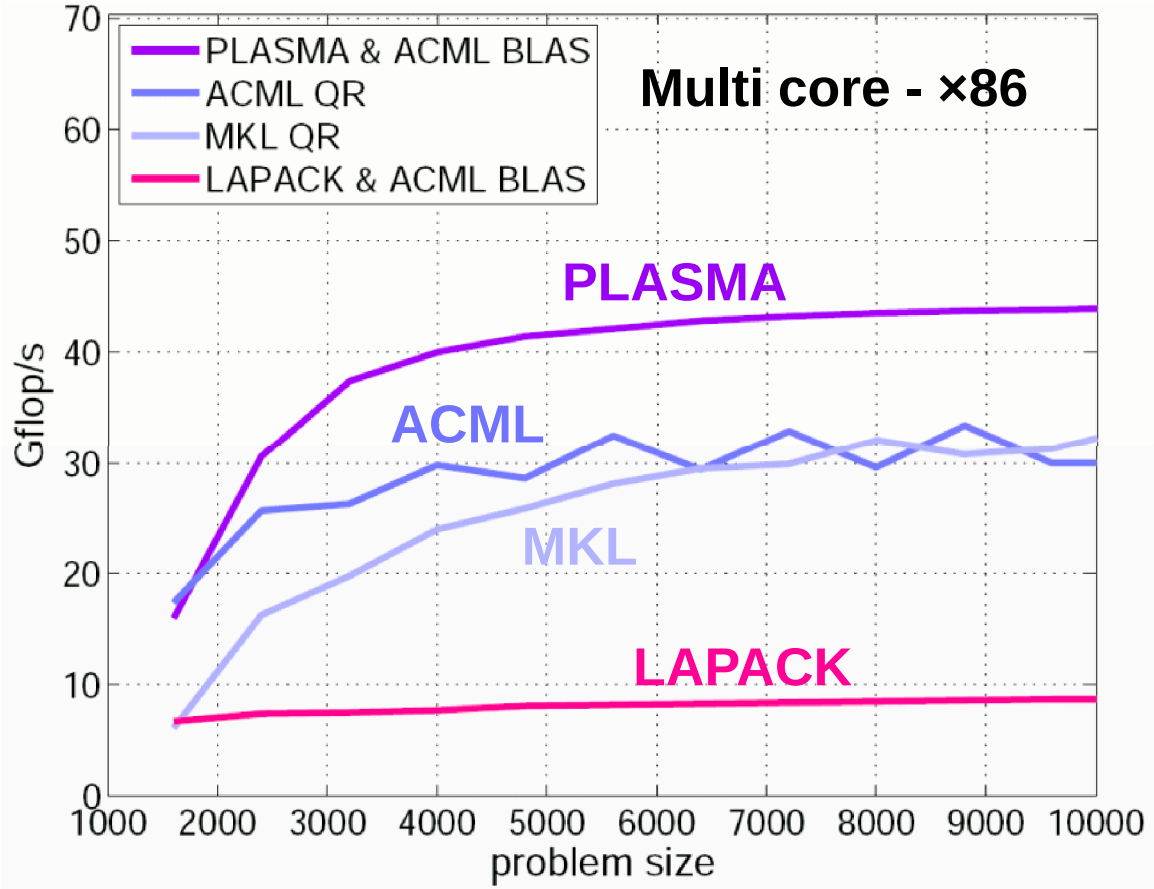
ICL UT Innovative Computing Laboratory

The University of Tennessee UT

Oak Ridge National Laboratory

Provide highest performance

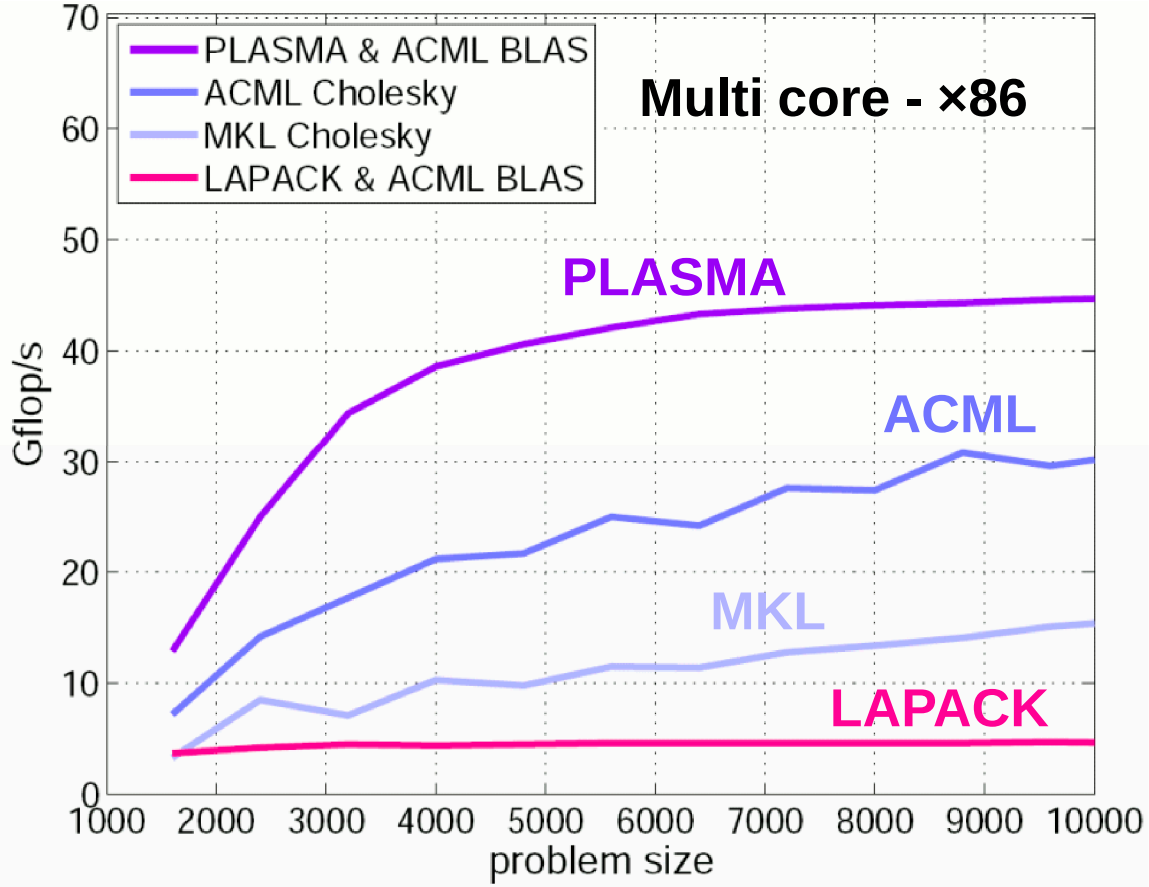
QR—octa-socket, dual-core Opteron



ICL UT Innovative Computing Laboratory
The University of Tennessee UT
Oak Ridge National Laboratory

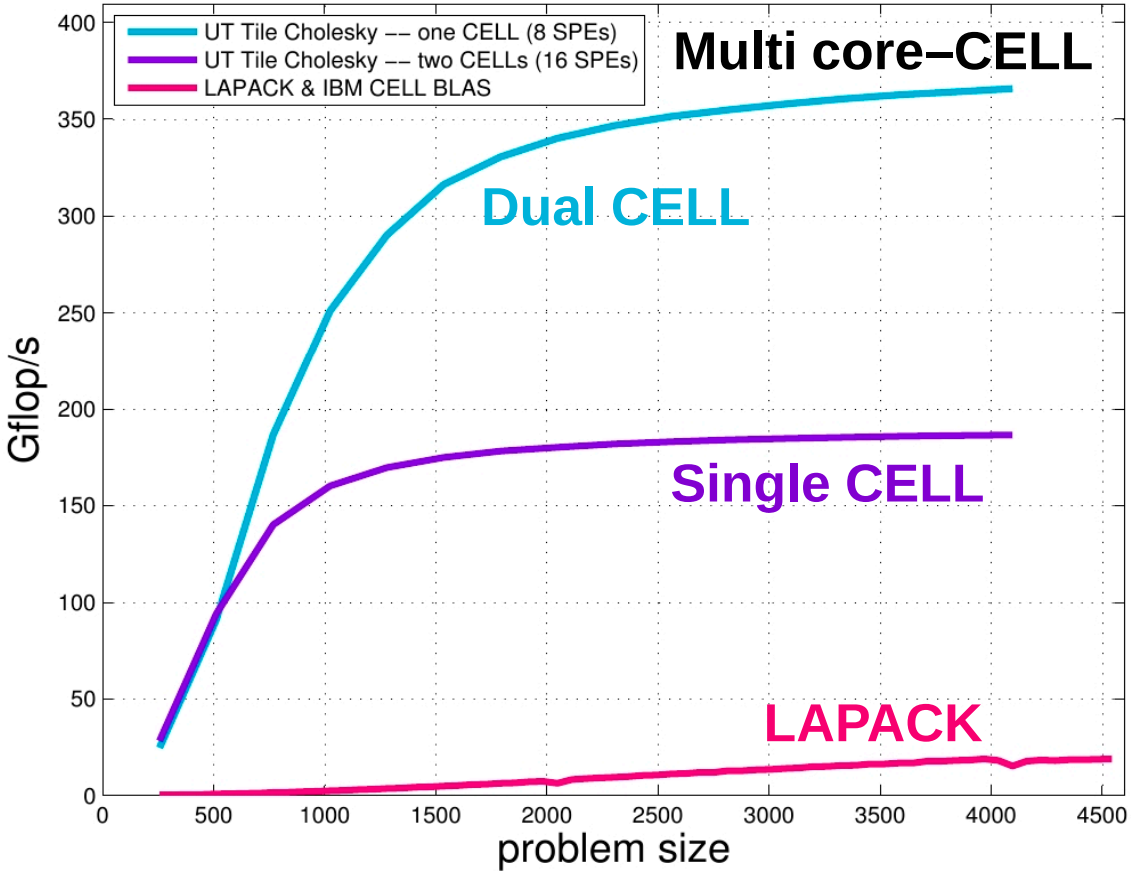
Provide highest performance

Cholesky—octa-socket, dual-core Opteron



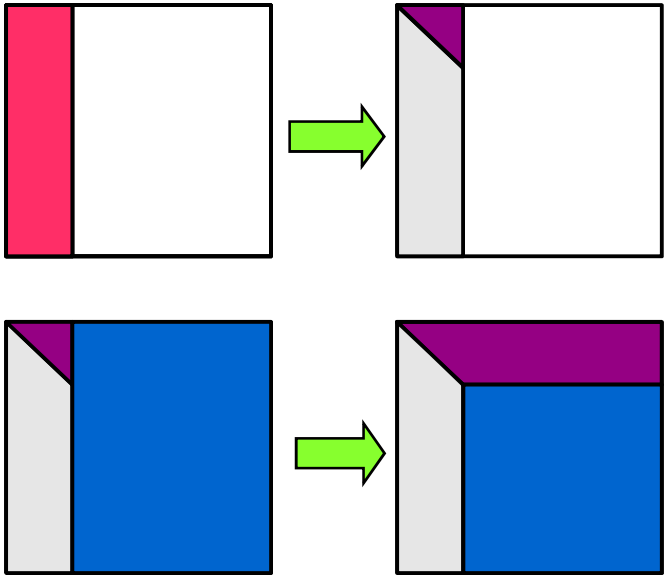
Provide highest performance

Cholesky—CELL Processor

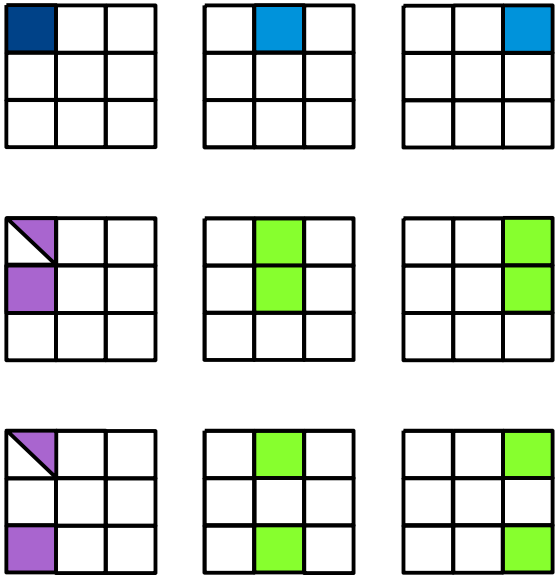


Utilize novel algorithms

Block algorithms – LAPACK

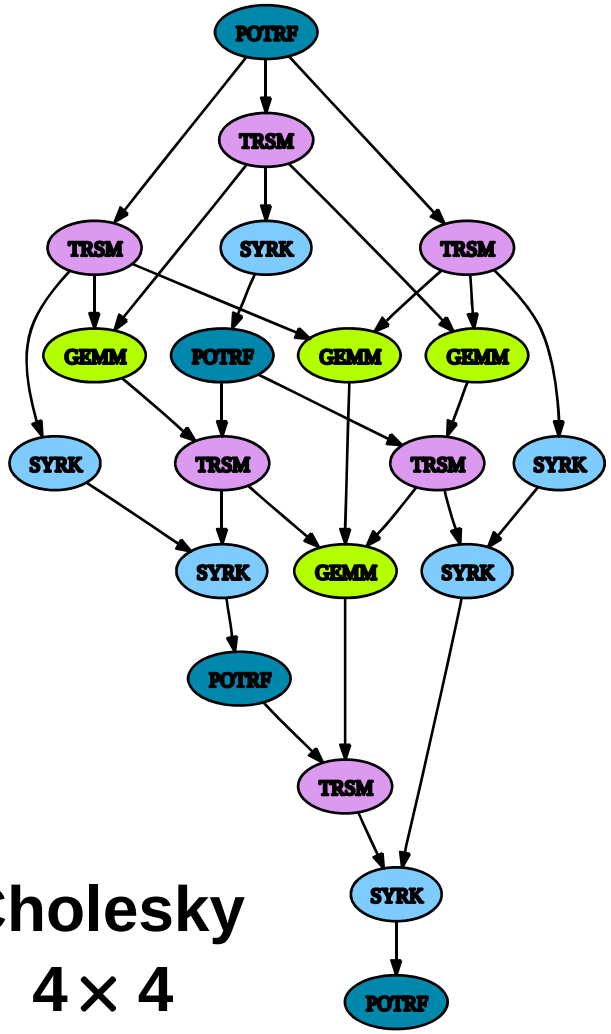


Tile algorithms – PLASMA

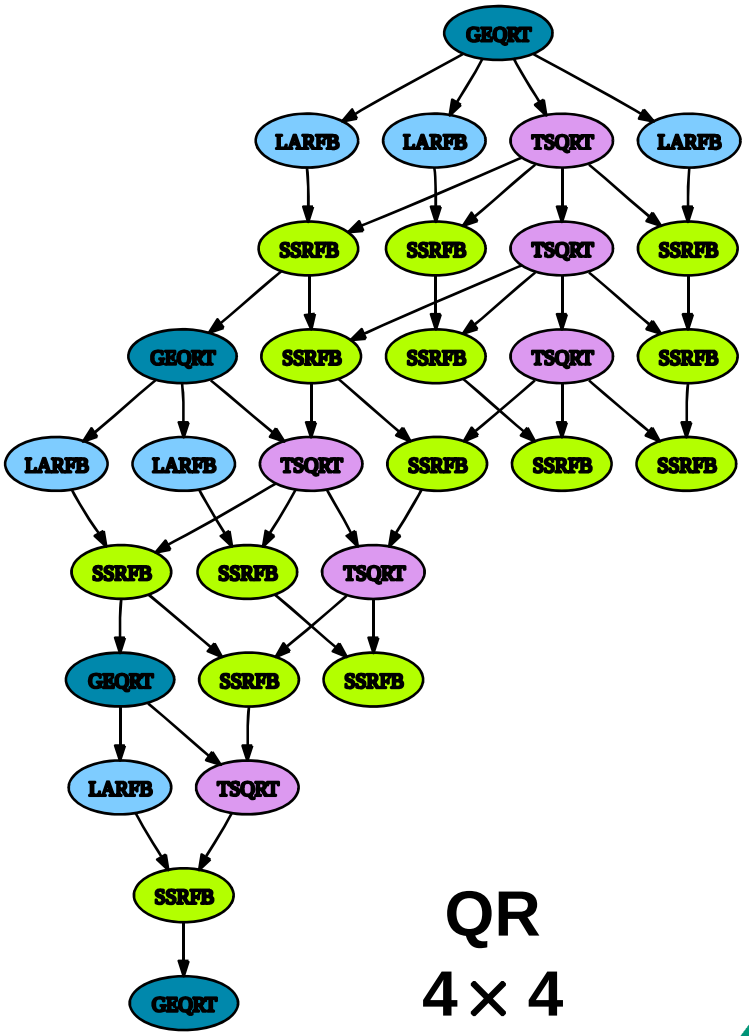


ICL UT Innovative Computing Laboratory
The University of Tennessee UT
Oak Ridge National Laboratory

Express computation as a DAG

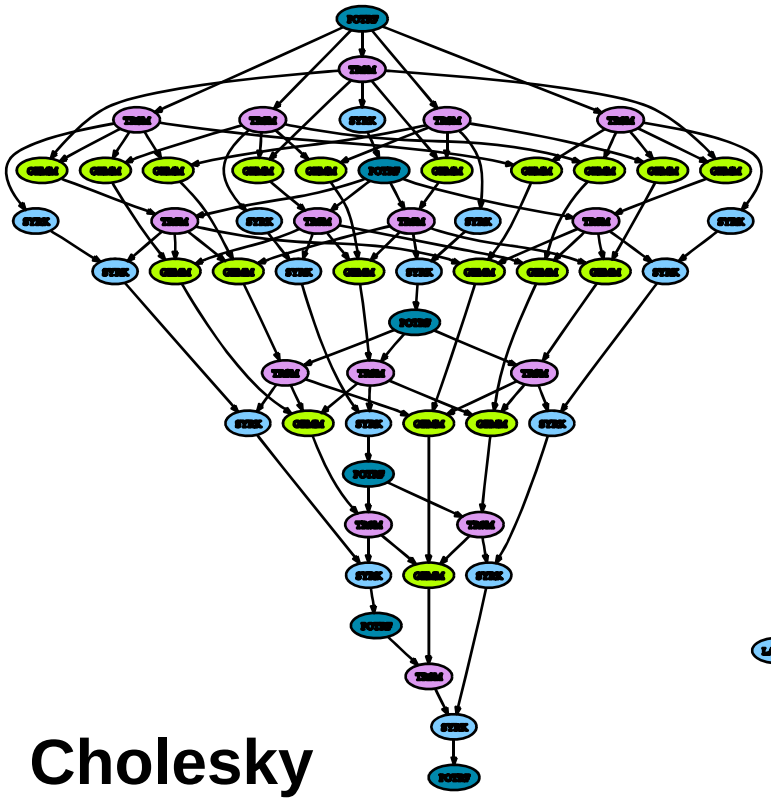


Cholesky
4 × 4

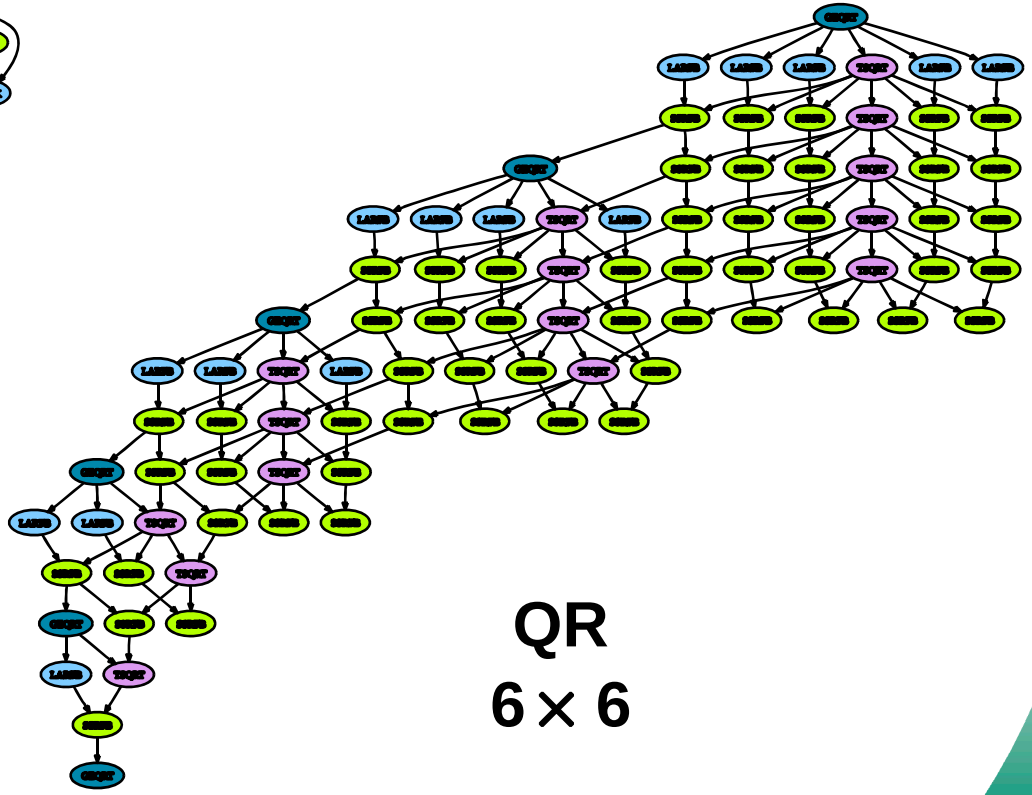


QR
4 × 4

Express computation as a DAG



Cholesky
6 × 6

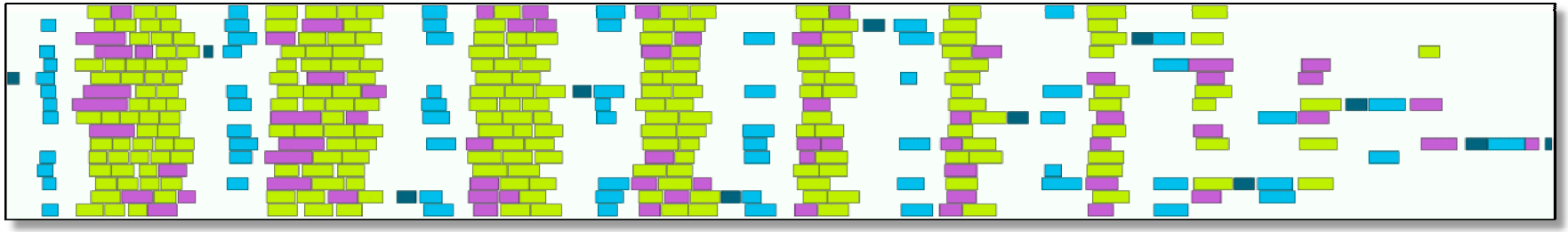


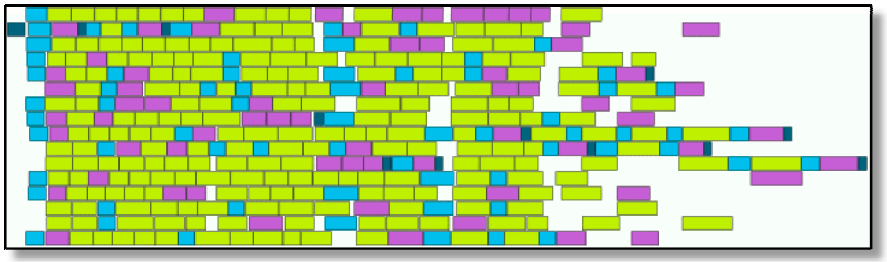
QR
6 × 6

ICL UT Innovative Computing Laboratory
The University of Tennessee UT
Oak Ridge National Laboratory

Dynamically schedule tasks – Cholesky

Nested parallelism (e.g., Cilk, TBB)





PLASMA
Arbitrary DAG
Fully dynamic scheduling

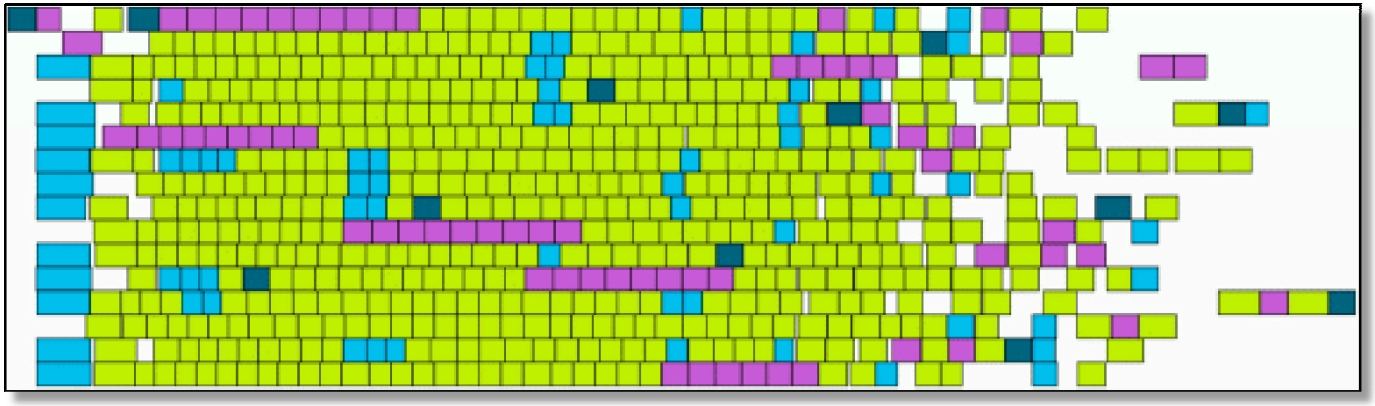
ICL UT Innovative Computing Laboratory
The University of Tennessee UT
Oak Ridge National Laboratory

Dynamically schedule tasks – QR

PLASMA

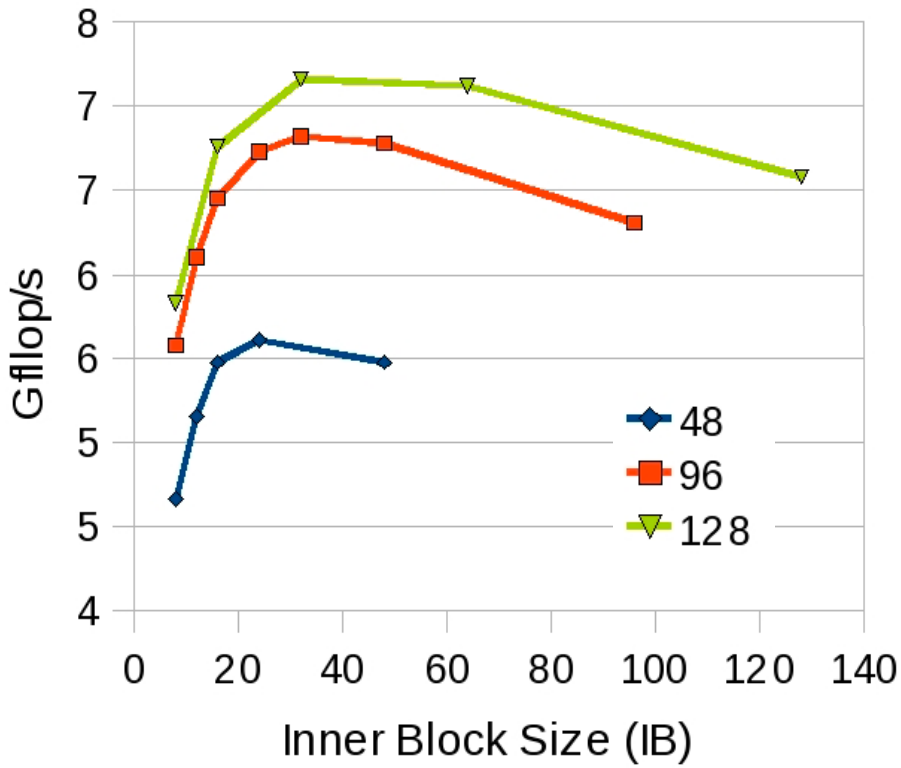
Arbitrary DAG

Fully dynamic scheduling



Model kernel performance – DSSRFB

DSSRFB kernel performance
Intel Xeon 2.4 GHz—single core

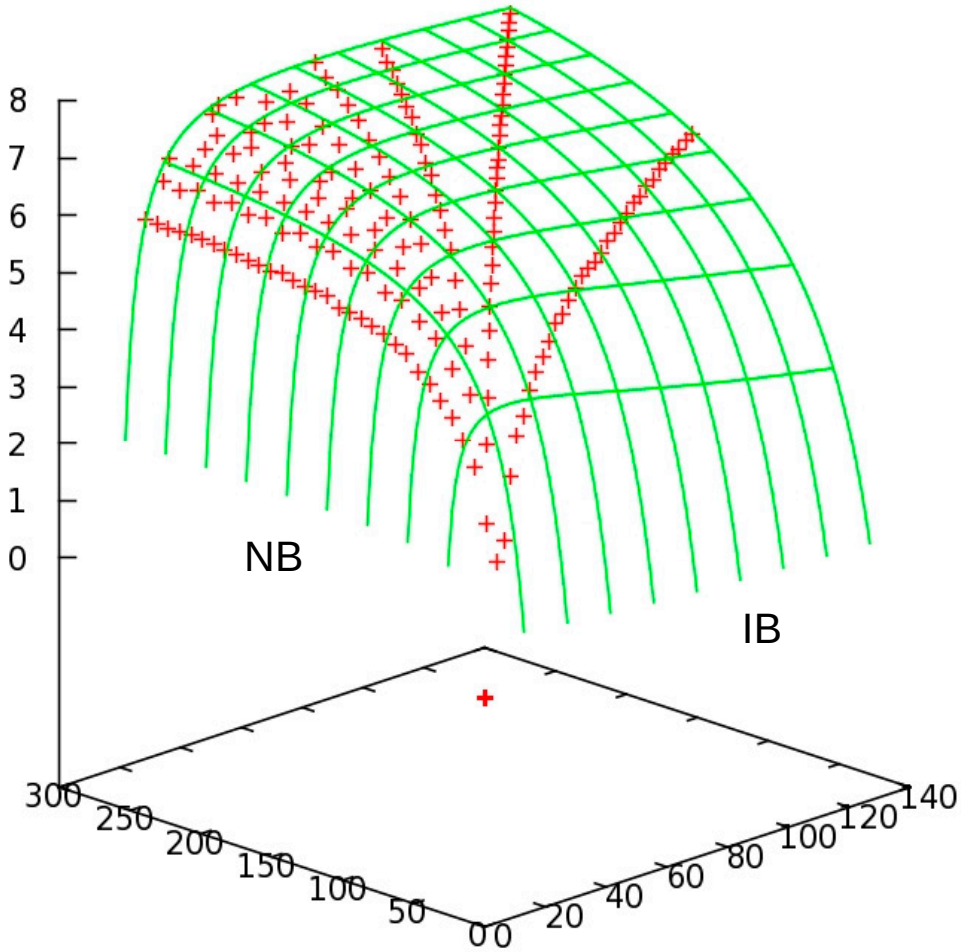


- **NB – Constant**
- **IB - ???**

Modeling and Tuning

ICL UT Innovative Computing Laboratory
The University of Tennessee UT
Oak Ridge National Laboratory

Model kernel performance – DSSRFB



Peak

9.6 (Gflop/s)

Model estimate

9.59 (Gflop/s)

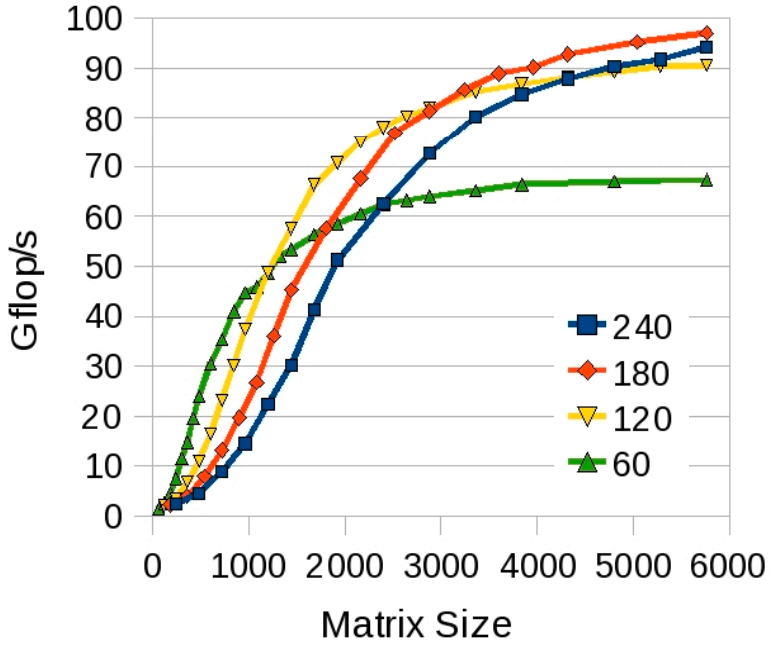
Error = 0.18%

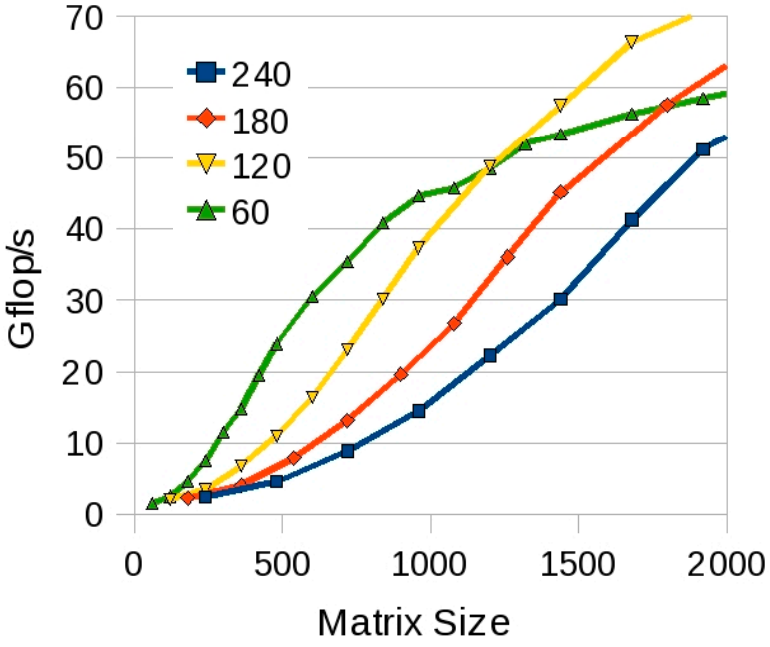
(of peak estimate)

Model algorithm performance – QR

Asymptotically the best != the best

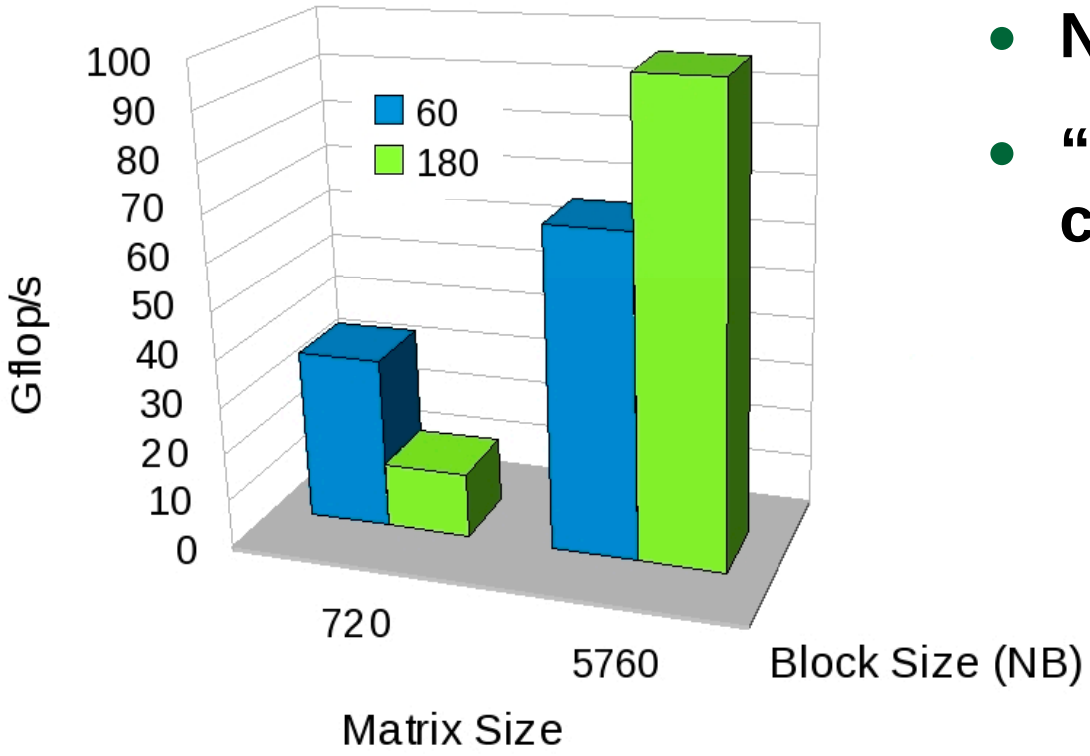
Tile QR—Intel Xeon 2.4 GHz
Quad-socket quad-core (16 cores)





Tune algorithm performance – QR

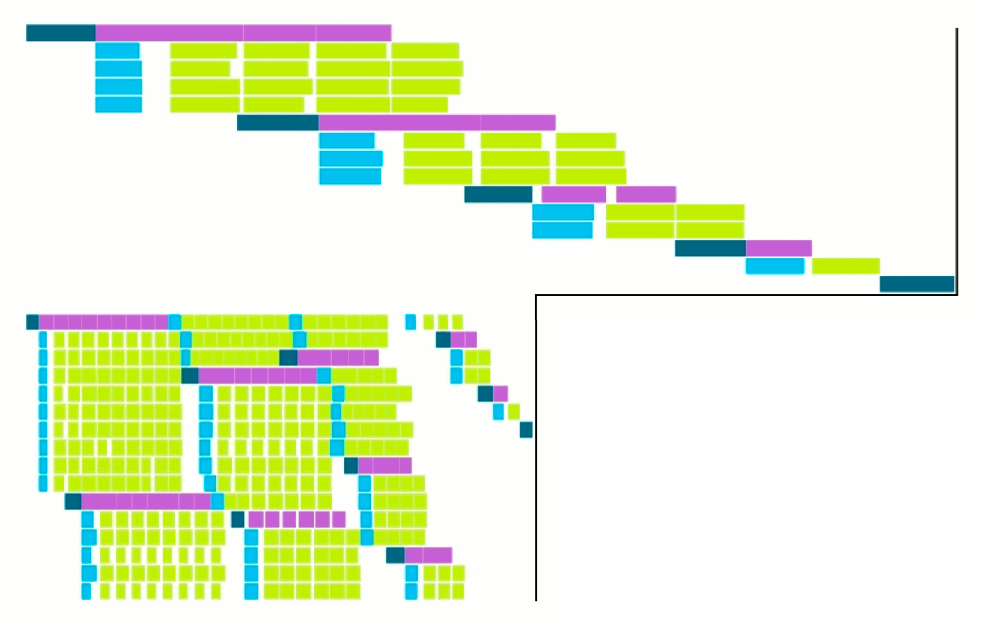
Tile QR—Intel Xeon 2.4 GHz
Quad-socket quad-core (16 cores)

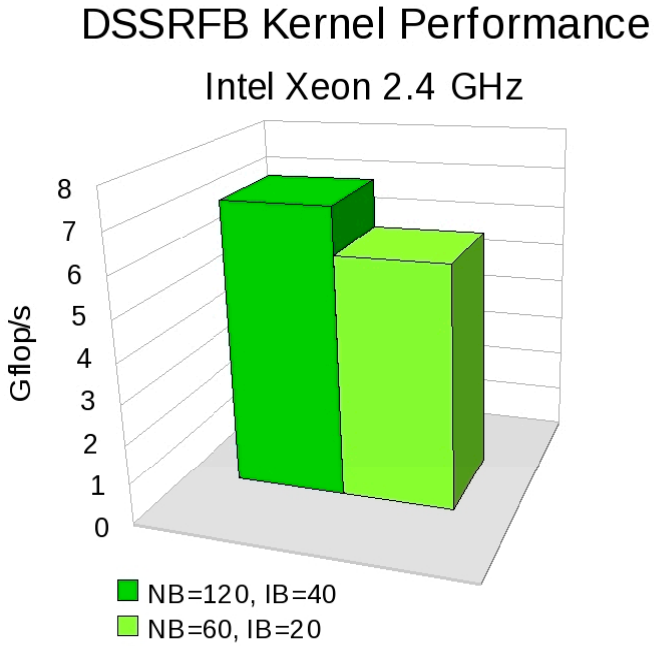


- **No "one size fits all"**
- **"Blind" choice can give catastrophic effects**

ICL UT Innovative Computing Laboratory
The University of Tennessee UT
Oak Ridge National Laboratory

Tune algorithm performance – QR





For smaller problems finer granularity allows exploitation of a higher degree of parallelism at the price of small per-core performance drop

ICL UT Innovative Computing Laboratory
The University of Tennessee UT
Oak Ridge National Laboratory

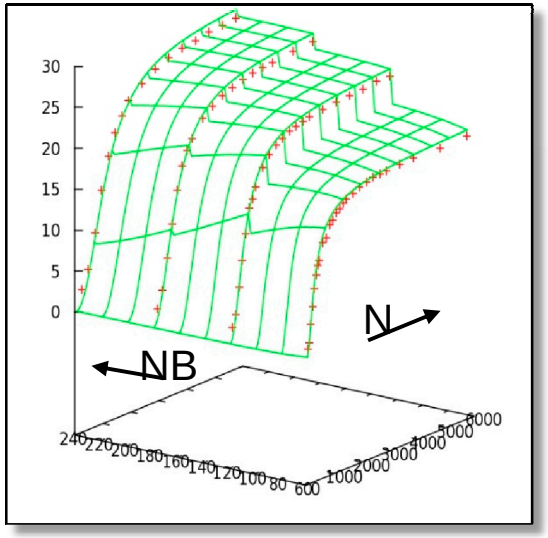
Tune algorithm performance – QR

Gflop/s = F (NB, N, P)

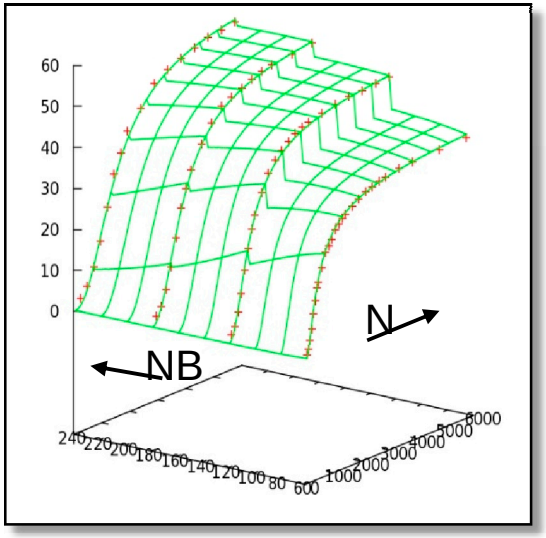
Processors/cores

Problem size

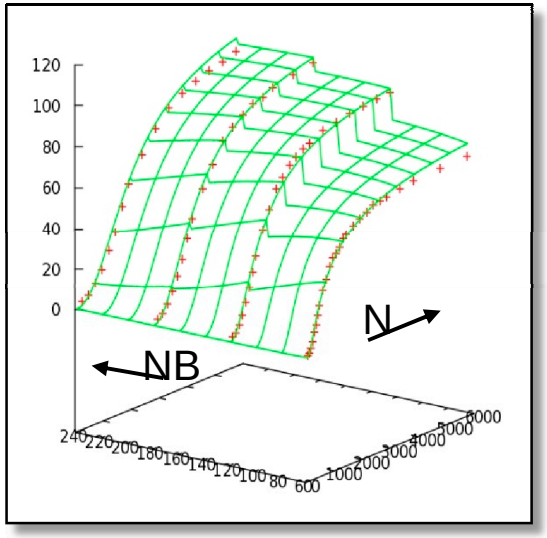
Tunable parameter



4 Cores



8 Cores



16 Cores

ICL UT Innovative Computing Laboratory
The University of Tennessee UT
Oak Ridge National Laboratory

Contact

Jack Dongarra
dongarra@cs.utk.edu