NLAFET
European Commission
HORIZON 2020
Parallel Numerical Linear Algebra
for Future Extreme Scale Systems

Parallel Numerical Linear Algebra for Future Extreme-Scale Systems

Bo Kågström and the NLAFET Consortium Team

Dept. of Computing Science and HPC2N, Umeå University, Sweden
bokg@cs.umu.se, info@nlafet.eu, www.nlafet.eu

EXDCI Workshop, Praha, May 9–10, 2016

Umeå Universitet
Inria informatics mathematics
Science & Technology Facilities Council
MANCHESTER 1824 The University of Manchester

Members of the NLAFET Consortium

- Umeå University, Sweden (UMU; *Coordinator* Bo Kågström; Lennart Edblom)
- The University of Manchester, UK (UNIMAN; Jack Dongarra)
- Institute National de Recherche en Informatique et en Automatique, France (INRIA; Laura Grigori)
- Science and Technology Facilities Council, UK (STFC; Iain Duff)

NLAFET
Parallel Numerical Linear Algebra
for Future Extreme Scale Systems

Key European players with recognized leadership, proven expertise, experience, and skills across the scientific areas of NLAFET!

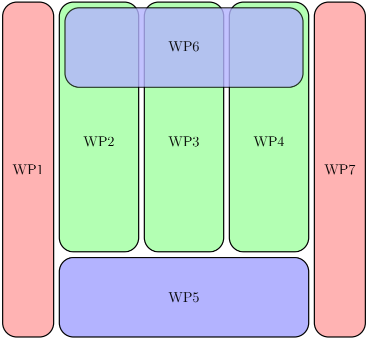
Vast experience contributing to open source projects!

NLAFET—Aim and Main Research Objectives

Aim: *Enable a radical improvement in the performance and scalability of a wide range of real-world applications relying on linear algebra software for future extreme-scale systems.*

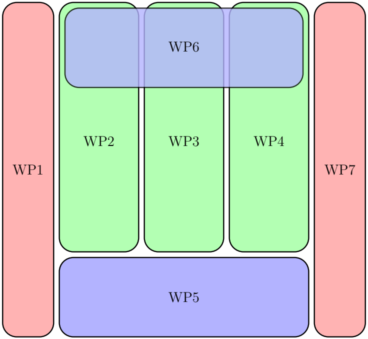
- Development of novel *architecture-aware algorithms* that expose as much parallelism as possible, exploit heterogeneity, avoid communication bottlenecks, respond to escalating fault rates, and help meet emerging power constraints
- Exploration of *advanced scheduling strategies and runtime systems* focusing on the extreme scale and strong scalability in multi/many-core and hybrid environments
- Design and evaluation of novel strategies and software support for both *offline and online auto-tuning*
- Results will appear in the open source *NLAFET software library*

NLAFET Work Package Overview



- WP1: *Management and coordination*
- WP5: *Challenging applications—a selection*
  Materials science, power systems, study of energy solutions, and data analysis in astrophysics
- WP7: *Dissemination and community outreach*
  Research and validation results; stakeholder communities

Research focus—Critical set of NLA operations



- WP2: *Dense linear systems and eigenvalue problem solvers*
- WP3: *Direct solution of sparse linear systems*
- WP4: *Communication-optimal algorithms for iterative methods*
- WP6: *Cross-cutting issues*

WP2, WP3 and WP4: research into extreme-scale parallel algorithms
WP6: research into methods for solving common cross-cutting issues

WP2, WP3 and WP4 at a glance!

- Linear Systems Solvers
- Hybrid (Batched) BLAS
- Eigenvalue Problem Solvers
- Singular Value Decomposition Algorithms

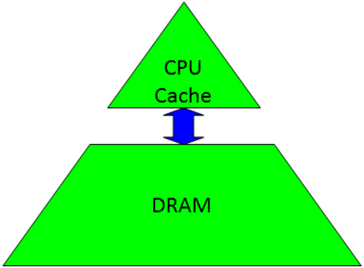
- Lower Bounds on Communication for Sparse Matrices
- Direct Methods for (Near–)Symmetric Systems
- Direct Methods for Highly Unsymmetric Systems
- Hybrid Direct–Iterative Methods

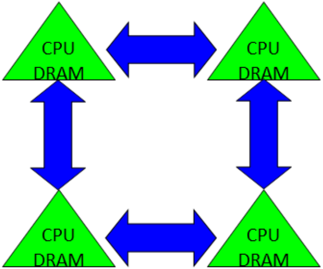
- Computational Kernels for Preconditioned Iterative Methods
- Iterative Methods: use p vectors per it, nearest-neighbor comm
- Preconditioners: multi-level, comm. reducing

Why avoid communication?

Algorithms have two costs (measured in time or energy):

1. Arithmetic (FLOPS)
2. Communication: moving data between
    - levels of a memory hierarchy (sequential case)
    - processors over a network (parallel case).





Extreme scale systems accentuate the need to avoid communication!

Why avoid communication?

Running time of an algorithm involve three terms:

- # Flops $*$ Time_per_flop
- # Words moved / Bandwidth
- # Messages $*$ Latency

Time_per_flop $\ll$ 1 / Bandwidth $\ll$ Latency

Gaps growing exponential with time [FOSC]

Annual improvements			
Time per flop		*Bandwidth*	*Latency*
59%	Network	26%	15%
	DRAM	23%	5%

Goal: Redesign algorithms (or invent new) to avoid communication!
Attain lower bounds on communication if possible!

Batched BLAS motivation

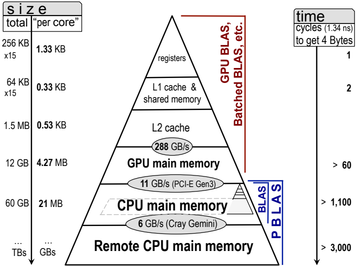
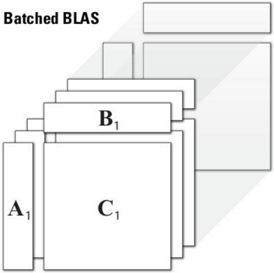


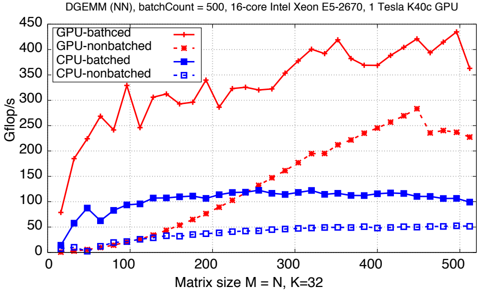
Figure: Memory hierarchy of a heterogeneous system from the point of view of a CUDA core of an NVIDIA K40c GPU with $2,880$ CUDA cores.

- Accelerators coprocessors, like GPUs, support high levels of parallelism.
- Can achieve very high performance for large data parallel computations if CPU handles computations on critical path.
- Currently, not the case for applications that involve large amounts of data that come in small units.

Batched BLAS

Multiple independent BLAS operations on small matrices grouped together as a single routine





Sample applications: Structural mechanics, Astrophysics, Direct sparse solvers, High-order FEM simulations

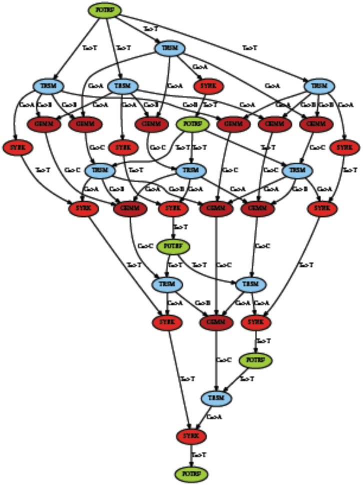
WP6: Cross-cutting issues and challenges!

Extreme-scale systems are hierarchical and heterogeneous in nature!

- Scheduling and Runtime Systems:
  - Task-graph-based multi-level scheduler for multi-level parallelism
  - Investigate user-guided schedulers: application-dependent balance between locality, concurrency, and scheduling overhead
  - Run-time system based on parallelizing critical tasks ($Ax = \lambda Bx$)
  - Address the thread-to-core mapping problem
- Auto-Tuning:
  - Off-line: tuning of critical numerical kernels across hybrid systems
  - Run-time: use feedback during and/or between executions on similar problems to tune in later stages of the algorithm
- Algorithm-Based Fault Tolerance:
  - Explore new NLA methods of resilence and develop algorithms with these capabilities.

Task-graph based scheduling and run-time systems

- Express algorithmic dataflow, *not* explicit data movement
- Blocked Cholesky tasks: POTRF, TRSM, GEMM, SYRK
- PTG representation: symbolic, problem size independent



Task-graph based scheduling and run-time system

- Data flow based execution using PaRSEC (ICL-UTK)
- Assigns computations threads to cores; overlaps comm. & comp.
- Distributed dynamic scheduler based on NUMA nodes and data reuse

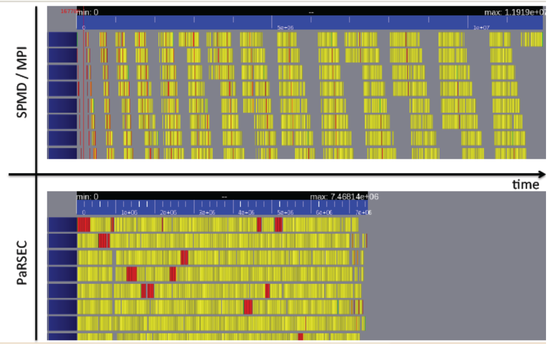
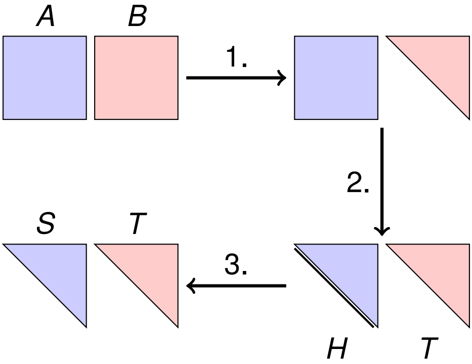


Figure: Cholesky PTG run by PaRSEC; 45% improvement

Generalized eigenvalue problem

Find pairs of eigenvalues λ and eigenvectors x s.t.

$$Ax = \lambda Bx$$



1. QR factorization
2. Hessenberg-Triangular reduction
3. QZ algorithm (generalized Schur decomposition)

Motivating (terrifying) example

Tunable parameters in state-of-the-art parallel QZ algorithm:

n_{min1}	Algorithm selection threshold.
n_{min2}	Algorithm selection threshold.
n_{min3}	Parallelization threshold.
P_{AED}	Number of processors for subproblems.
`MMULT`	Level-3 BLAS threshold.
`NCB`	Cache-blocking block size.
`NIBBLE`	Loop break threshold.
n_{AED}	Deflation window size.
n_{shift}	Number of shifts per iteration.
`NUMWIN`	Number of windows.
`WINEIG`	Eigenvalues per window.
`WINSIZE`	Window size.
`WNEICR`	Number of eigenvalues moved together.

WP5: Challenging applications—a selection

- *Dense solvers/eigensolvers in materials science and chemistry*
  - Thomas Schulthess, ETH Zurich, Switzerland
  - $Ax = \lambda Bx$, A Hermitian dense, B Hermitian positive definite
- *Load flow based calculations in large-scale power systems*
  - Bernd Klöss, DIgSILENT GmbH, Germany
  - Extreme scale, highly sparse, unsymmetrical and very ill-conditioned $Ax = b$
- *Energy solutions and Code Saturne*
  - Yvan Fournier, EDF, France
  - Communication-avoiding methods for sparse linear systems
- *Data analysis in astrophysics and the Midapack library*
  - Radoslaw Stompor, University Paris 7, France; Carlo Baccigalupi, SISSA Italy
  - Communication-avoiding methods adapted to generalized least-squares problem

NLAFET Summary

- Deliver a new generation of computational tools and software for problems in numerical linear algebra with a focus on extreme-scale systems
- Linear algebra is both fundamental and ubiquitous in computational science and its vast application areas
- Co-design effort for designing, prototyping, and deploying new NLA software libraries:
  - Exploration of new algorithms
  - Investigation of advanced scheduling strategies
  - Investigation of advanced auto-tuning strategies
  - Open source
- Stakeholder collaborations (users, academia, HW and SW vendors)

NLAFET
Parallel Numerical Linear Algebra
for Future Extreme Scale Systems