Designing Numerical Libraries for Millions of Cores

Jack Dongarra
University of Tennessee
Oak Ridge National Lab
University of Manchester


Manchester
Evening News
CITY
EDITION
What a relief! Sven's here at last...
ONLINE
More than 1,000

This picture was taken at Argonne around 1981

- Since then there have been tremendous changes in our scientific computing environment.
- Many changes in Mathematic Software and Numerical Libraries

This picture was taken at Argonne around 1981

- Since then there have been tremendous changes in our scientific computing environment.
- Many changes in Mathematic Software and Numerical Libraries

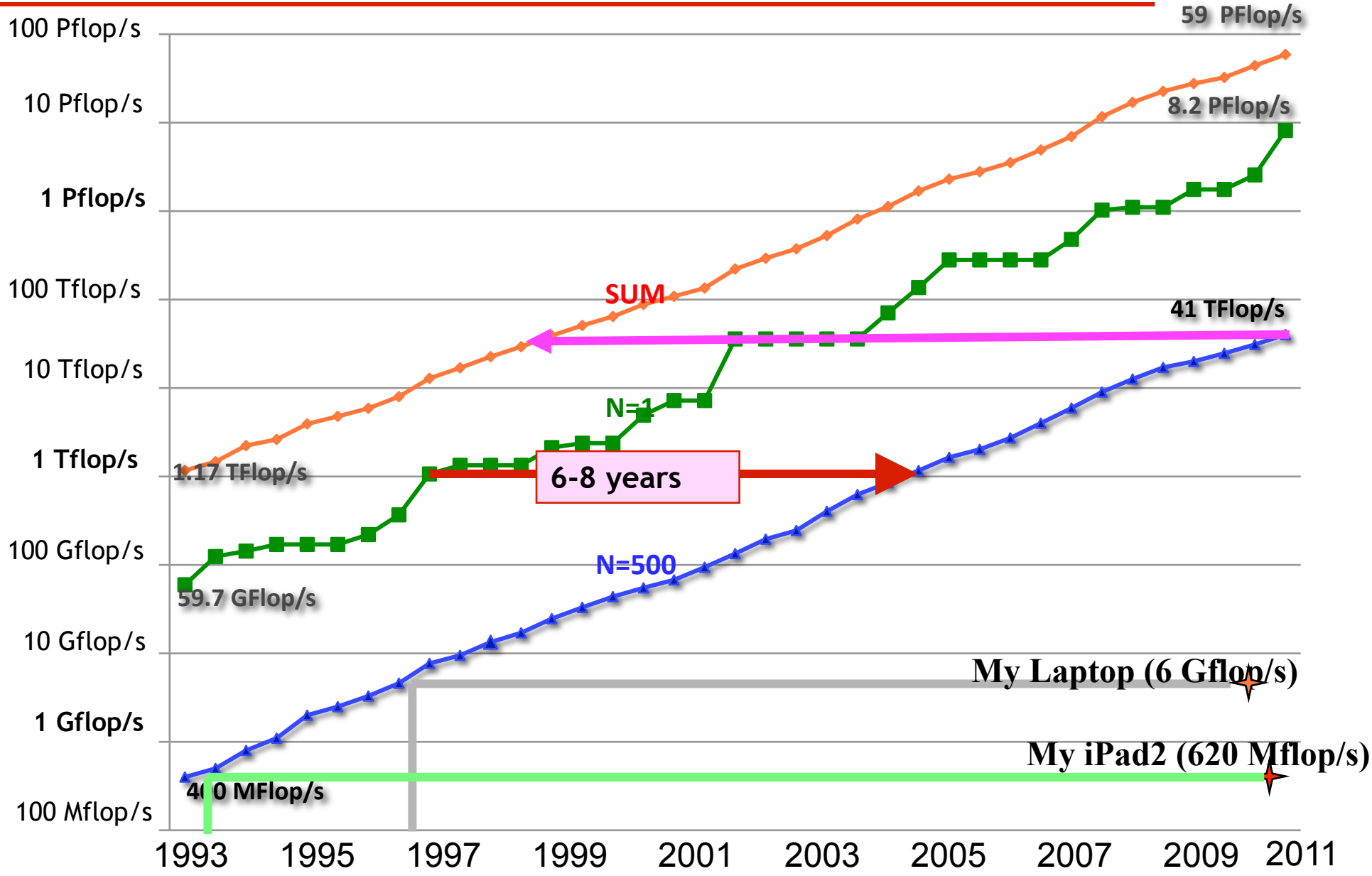
Numerical Linear Algebra Libraries

- Interested in developing numerical library for a range of computing platforms.

- Applications are given (as function of time)
- Architectures are given (as function of time)

- Algorithms and software must be adapted or created to bridge to architectures for the sake of the complex applications

Performance Development



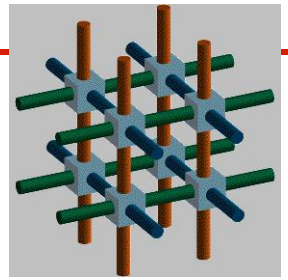
Three Design Points Today

- Gigascale Laptop: Uninode-Multicore
  (Your iPhone and iPad are Mflop/s devices)

- Terascale Deskside: Multinode-Multicore

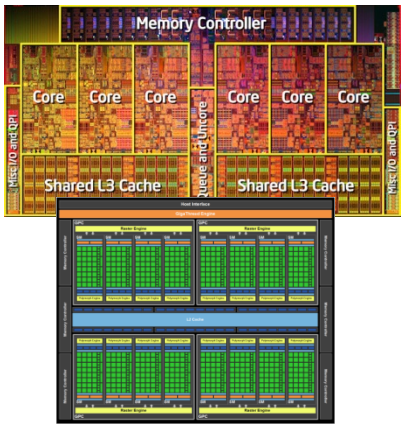
- Petacale Center: Multinode-Multicore

Programming Numerical Libraries for Parallel Machines

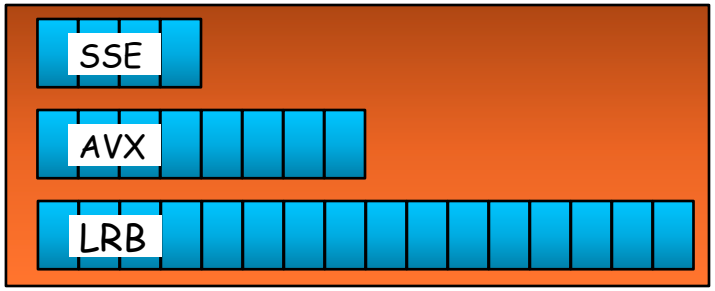


Node

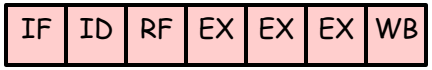
Socket



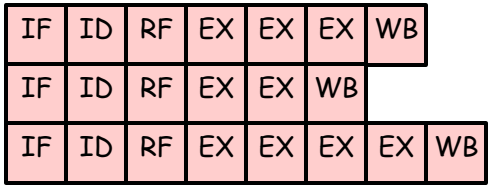
Core



Vector

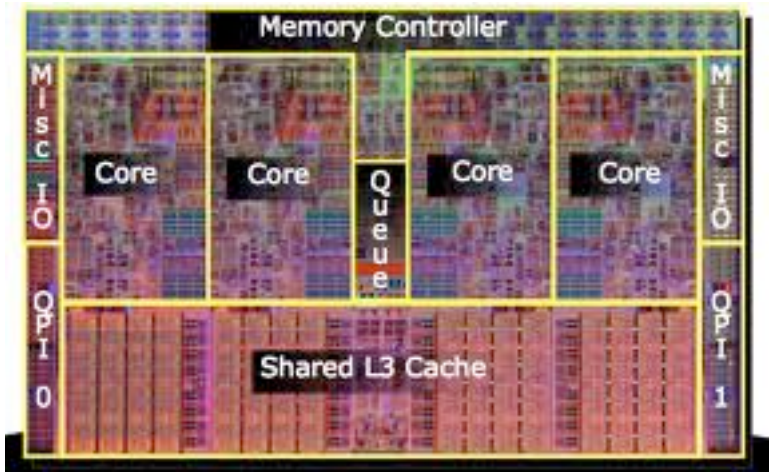


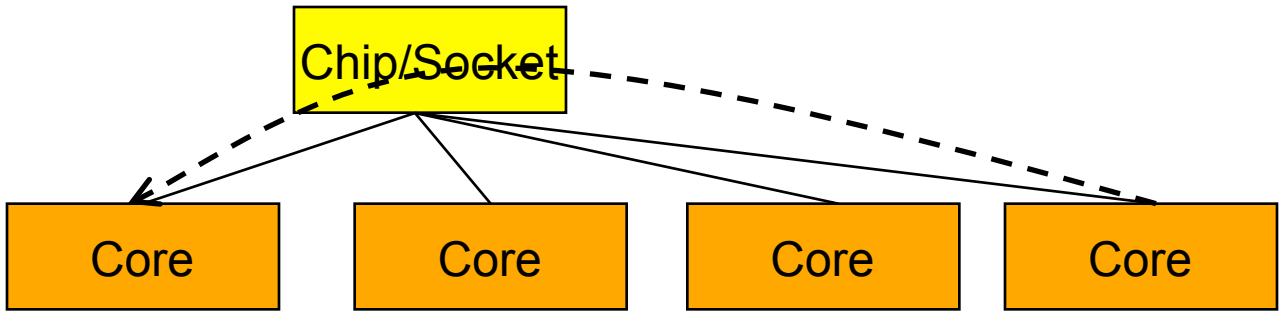
Pipeline



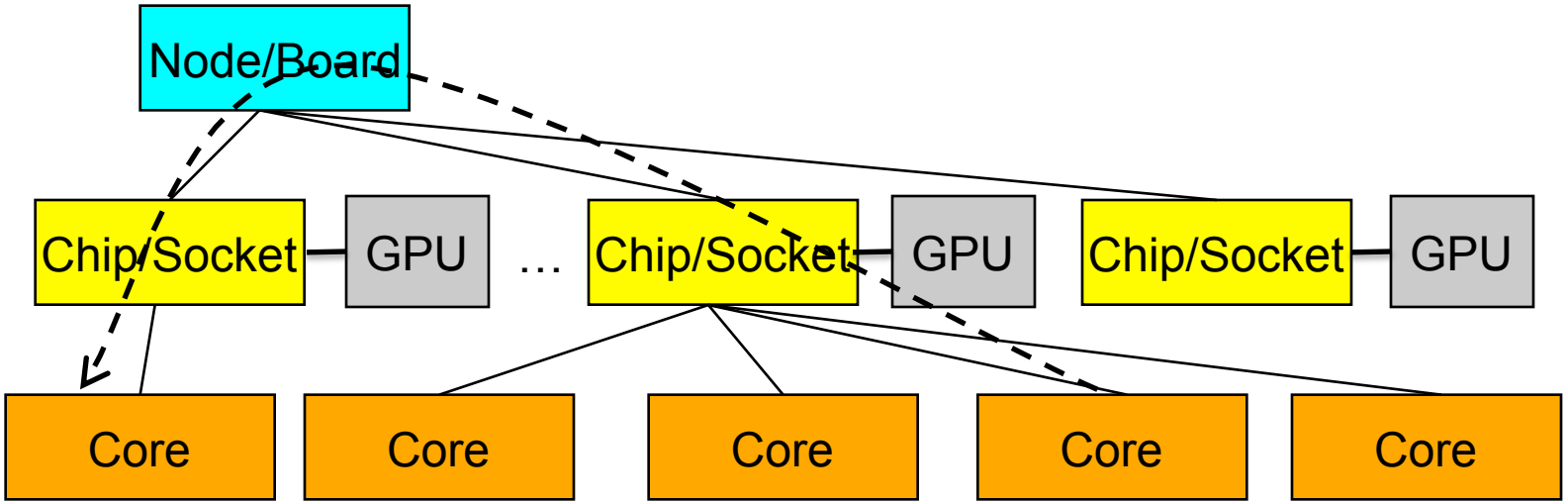
Instruction

Example of typical parallel machine



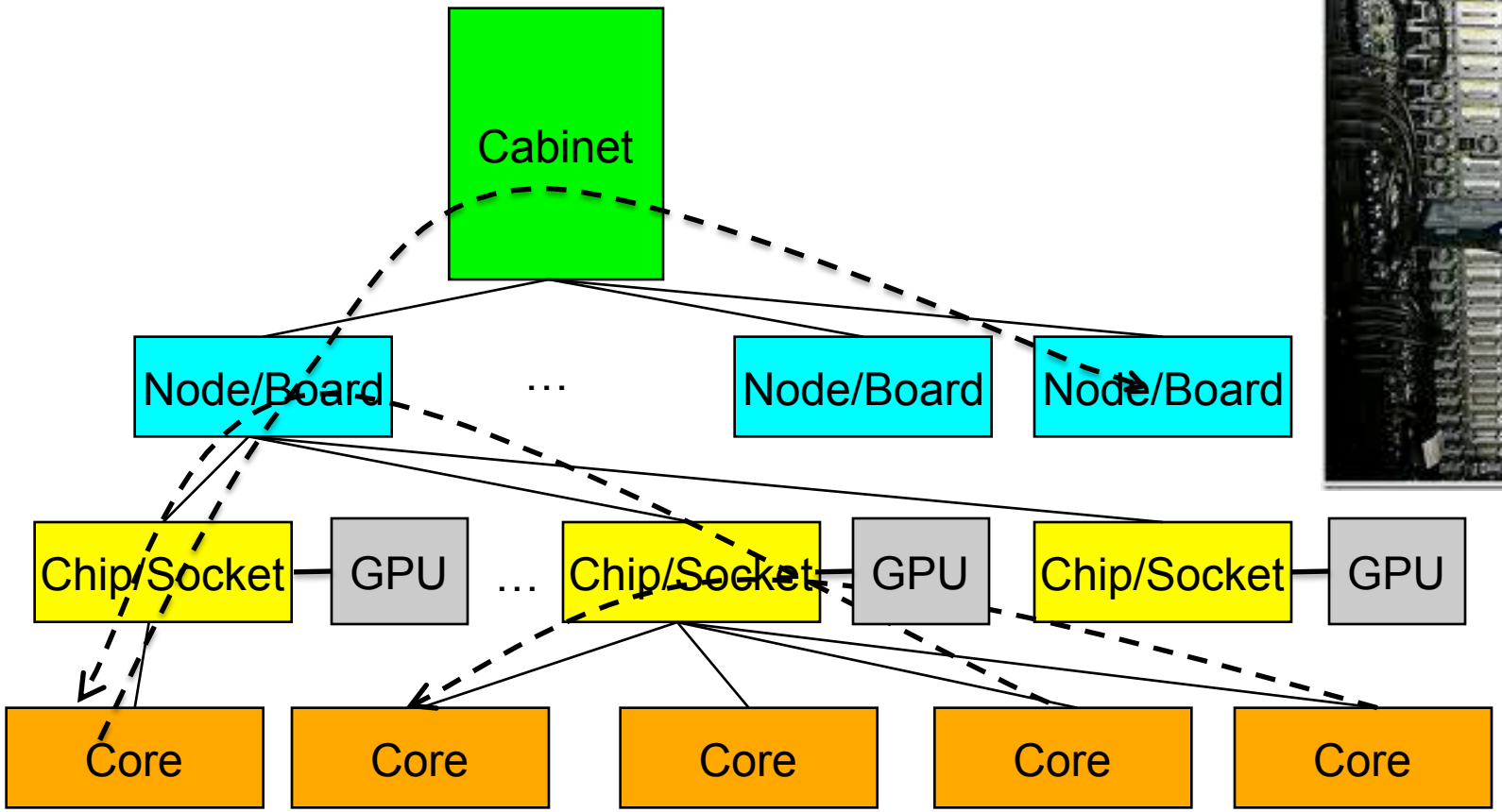


Example of typical parallel machine

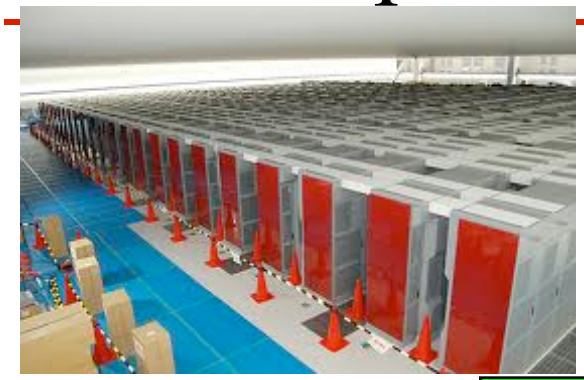


Example of typical parallel machine

Shared memory programming between processes on a board and a combination of shared memory and distributed memory programming between nodes and cabinets



Example of typical parallel machine



Combination of shared memory and distributed memory program

June 2011: The TOP10

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak
1	RIKEN Advanced Inst for Comp Sci	K Computer Fujitsu SPARC64 VIIIfx + custom	Japan	548,352	8.16	93
2	Nat. SuperComputer Center in Tianjin	Tianhe-1A, NUDT Intel + Nvidia GPU + custom	China	186,368	2.57	55
3	DOE / OS Oak Ridge Nat Lab	Jaguar, Cray AMD + custom	USA	224,162	1.76	75
4	Nat. Supercomputer Center in Shenzhen	Nebulea, Dawning Intel + Nvidia GPU + IB	China	120,640	1.27	43
5	GSIC Center, Tokyo Institute of Technology	Tusbame 2.0, HP Intel + Nvidia GPU + IB	Japan	73,278	1.19	52
6	DOE / NNSA LANL & SNL	Cielo, Cray AMD + custom	USA	142,272	1.11	81
7	NASA Ames Research Center/NAS	Plelades SGI Altix ICE 8200EX/8400EX + IB	USA	111,104	1.09	83
8	DOE / OS Lawrence Berkeley Nat Lab	Hopper, Cray AMD + custom	USA	153,408	1.054	82
9	Commissariat a l'Energie Atomique (CEA)	Tera-10, Bull Intel + IB	France	138,368	1.050	84
10	DOE / NNSA Los Alamos Nat Lab	Roadrunner, IBM AMD + Cell GPU + IB	USA	122,400	1.04	76

28 Supercomputers in the UK

Rank	Site	Computer	Cores	Rmax Tflop/s
24	University of Edinburgh	Cray XE6 12-core 2.1 GHz	44376	279
65	Atomic Weapons Establishment	Bullx B500 Cluster, Xeon X56xx 2.8Ghz, QDR Infiniband	12936	124
69	ECMWF	Power 575, p6 4.7 GHz, Infiniband	8320	115
70	ECMWF	Power 575, p6 4.7 GHz, Infiniband	8320	115
93	University of Edinburgh	Cray XT4, 2.3 GHz	12288	95
154	University of Southampton	iDataPlex, Xeon QC 2.26 GHz, Ifband, Windows HPC2008 R2	8000	66
160	IT Service Provider	Cluster Platform 4000 BL685c G7, Opteron 12C 2.2 Ghz, GigE	14556	65
186	IT Service Provider	Cluster Platform 3000 BL460c G7, Xeon X5670 2.93 Ghz, GigE	9768	59
190	Computacenter (UK) LTD	Cluster Platform 3000 BL460c G1, Xeon L5420 2.5 GHz, GigE	11280	58
191	Classified	xSeries x3650 Cluster Xeon QC GT 2.66 GHz, Infiniband	6368	58
211	Classified	BladeCenter HS22 Cluster, WM Xeon 6-core 2.66Ghz, Ifband	5880	55
212	Classified	BladeCenter HS22 Cluster, WM Xeon 6-core 2.66Ghz, Ifband	5880	55
213	Classified	BladeCenter HS22 Cluster, WM Xeon 6-core 2.66Ghz, Ifband	5880	55
228	IT Service Provider	Cluster Platform 4000 BL685c G7, Opteron 12C 2.1 Ghz, GigE	12552	54
233	Financial Institution	iDataPlex, Xeon X56xx 6C 2.66 GHz, GigE	9480	53
234	Financial Institution	iDataPlex, Xeon X56xx 6C 2.66 GHz, GigE	9480	53
278	UK Meteorological Office	Power 575, p6 4.7 GHz, Infiniband	3520	51
279	UK Meteorological Office	Power 575, p6 4.7 GHz, Infiniband	3520	51
339	Computacenter (UK) LTD	Cluster Platform 3000 BL460c, Xeon 54xx 3.0GHz, GigEthernet	7560	47
351	Asda Stores	BladeCenter HS22 Cluster, WM Xeon 6-core 2.93Ghz, GigE	8352	47
365	Financial Services	xSeries x3650M2 Cluster, Xeon QC E55xx 2.53 Ghz, GigE	8096	46
404	Financial Institution	BladeCenter HS22 Cluster, Xeon QC GT 2.53 GHz, GigEthernet	7872	44
405	Financial Institution	BladeCenter HS22 Cluster, Xeon QC GT 2.53 GHz, GigEthernet	7872	44
415	Bank	xSeries x3650M3, Xeon X56xx 2.93 GHz, GigE	7728	43
416	Bank	xSeries x3650M3, Xeon X56xx 2.93 GHz, GigE	7728	43
482	IT Service Provider	Cluster Platform 3000 BL460c G6, Xeon L5520 2.26 GHz, GigE	8568	40
484	IT Service Provider	Cluster Platform 3000 BL460c G6, Xeon X5670 2.93 GHz, 10G	4392	40

28 Supercomputers in the UK

Rank	Site	Computer	Cores	Rmax Tflop/s
24	University of Edinburgh	Cray XE6 12-core 2.1 GHz	44376	279
65	Atomic Weapons Establishment	Bullx B500 Cluster, Xeon X56xx 2.8Ghz, QDR Infiniband	12936	124
69	ECMWF	Power 575, p6 4.7 GHz, Infiniband	8320	115
70	ECMWF	Power 575, p6 4.7 GHz, Infiniband	8320	115
93	University of Edinburgh	Cray XT4, 2.3 GHz	12288	95
154	University of Southampton	iDataPlex, Xeon QC 2.26 GHz, Ifband, Windows HPC2008 R2	8000	66
160	IT Service Provider	Cluster Platform 4000 BL685c G7, Opteron 12C 2.2 Ghz, GigE	14556	65
186	IT Service Provider	Cluster Platform 3000 BL460c G7, Xeon X5670 2.93 Ghz, GigE	9768	59
190	Computacenter (UK) LTD	Cluster Platform 3000 BL460c G1, Xeon L5420 2.5 GHz, GigE	11280	58
191	Classified	xSeries x3650 Cluster Xeon QC GT 2.66 GHz, Infiniband	6368	58
211	Classified	BladeCenter HS22 Cluster, WM Xeon 6-core 2.66Ghz, Ifband	5880	55
212	Classified	BladeCenter HS22 Cluster, WM Xeon 6-core 2.66Ghz, Ifband	5880	55
213	Classified	BladeCenter HS22 Cluster, WM Xeon 6-core 2.66Ghz, Ifband	5880	55
228	IT Service Provider	Cluster Platform 4000 BL685c G7, Opteron 12C 2.1 Ghz, GigE	12552	54
233	Financial Institution	iDataPlex, Xeon X56xx 6C 2.66 GHz, GigE	9480	53
234	Financial Institution	iDataPlex, Xeon X56xx 6C 2.66 GHz, GigE	9480	53
278	UK Meteorological Office	Power 575, p6 4.7 GHz, Infiniband	3520	51
279	UK Meteorological Office	Power 575, p6 4.7 GHz, Infiniband	3520	51
339	Computacenter (UK) LTD	Cluster Platform 3000 BL460c, Xeon 54xx 3.0GHz, GigEthernet	7560	47
351	Asda Stores	BladeCenter HS22 Cluster, WM Xeon 6-core 2.93Ghz, GigE	8352	47
365	Financial Services	xSeries x3650M2 Cluster, Xeon QC E55xx 2.53 Ghz, GigE	8096	46
404	Financial Institution	BladeCenter HS22 Cluster, Xeon QC GT 2.53 GHz, GigEthernet	7872	44
405	Financial Institution	BladeCenter HS22 Cluster, Xeon QC GT 2.53 GHz, GigEthernet	7872	44
415	Bank	xSeries x3650M3, Xeon X56xx 2.93 GHz, GigE	7728	43
416	Bank	xSeries x3650M3, Xeon X56xx 2.93 GHz, GigE	7728	43
482	IT Service Provider	Cluster Platform 3000 BL460c G6, Xeon L5520 2.26 GHz, GigE	8568	40
484	IT Service Provider	Cluster Platform 3000 BL460c G6, Xeon X5670 2.93 GHz, 10G	4392	40

Three Design Points for Tomorrow

- Terascale Laptop: Manycore
- Petascale Deskside: Manynode-Manycore
- Exacale Center: Manynode-Manycore

Potential System Architecture

Systems	2011 K Computer
System peak	**8.7 Pflop/s**
Power	**10 MW**
System memory	1.6 PB
Node performance	128 GF
Node memory BW	64 GB/s
Node concurrency	8
Total Node Interconnect BW	20 GB/s
System size (nodes)	68,544
Total concurrency	548,352
MTTI	days

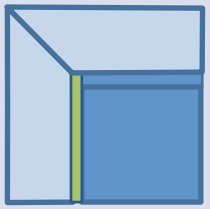
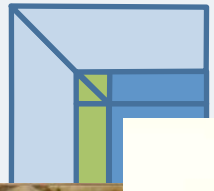
Potential System Architecture with a cap of $200M and 20MW

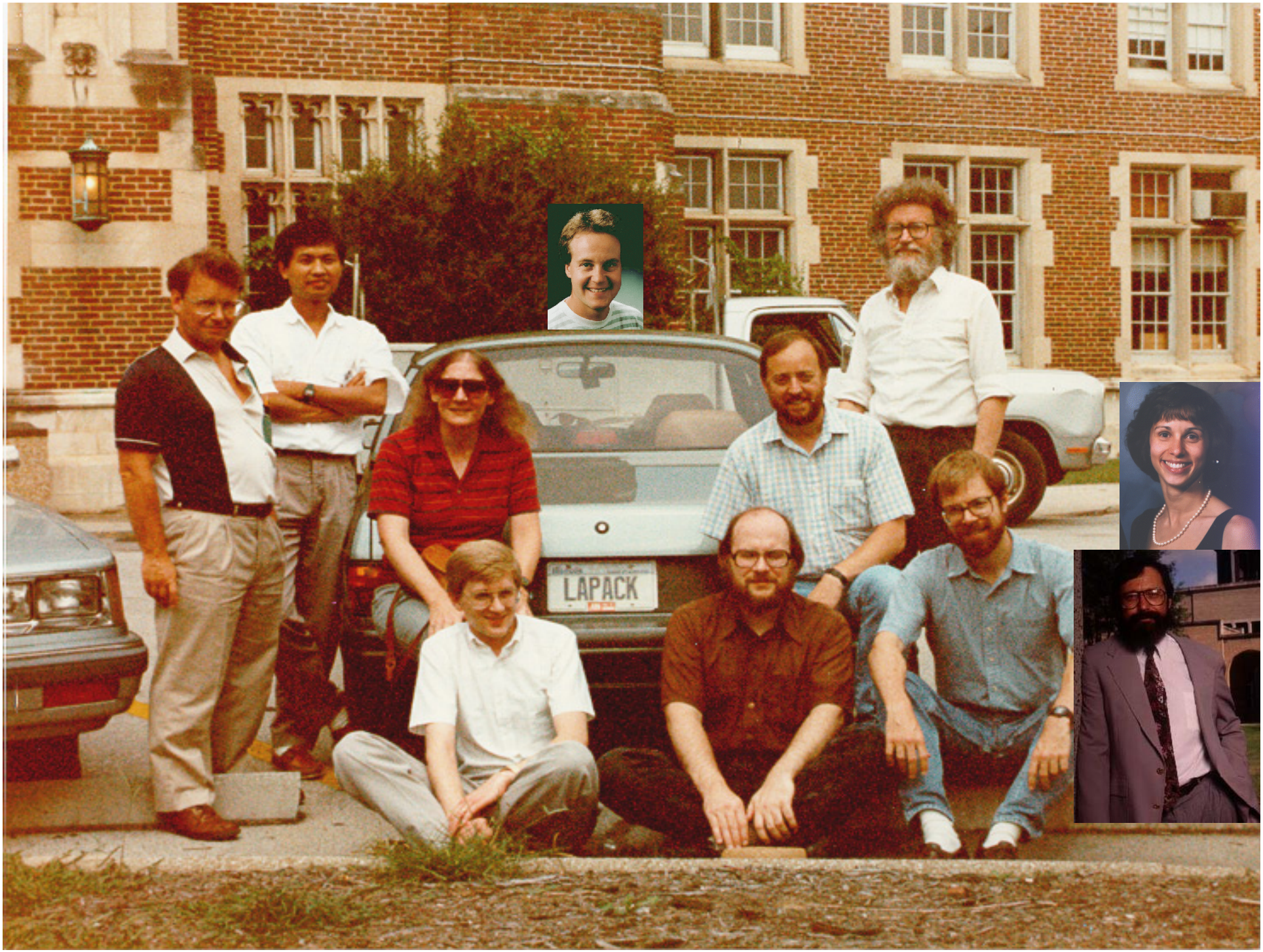
Systems	2011 K Computer	2019	Difference Today & 2019
System peak	**8.7 Pflop/s**	**1 Eflop/s**	**O(100)**
Power	**10 MW**	**~20 MW**	
System memory	1.6 PB	32 - 64 PB	O(10)
Node performance	128 GF	1,2 or 15TF	O(10) – O(100)
Node memory BW	64 GB/s	2 - 4TB/s	O(100)
Node concurrency	8	O(1k) or 10k	O(100) – O(1000)
Total Node Interconnect BW	20 GB/s	200-400GB/s	O(10)
System size (nodes)	68,544	O(100,000) or O(1M)	O(10) – O(100)
Total concurrency	548,352	O(billion)	O(1,000)
MTTI	days	O(1 day)	- O(10)

Critical Issues at Peta & Exascale for Algorithm and Software Design

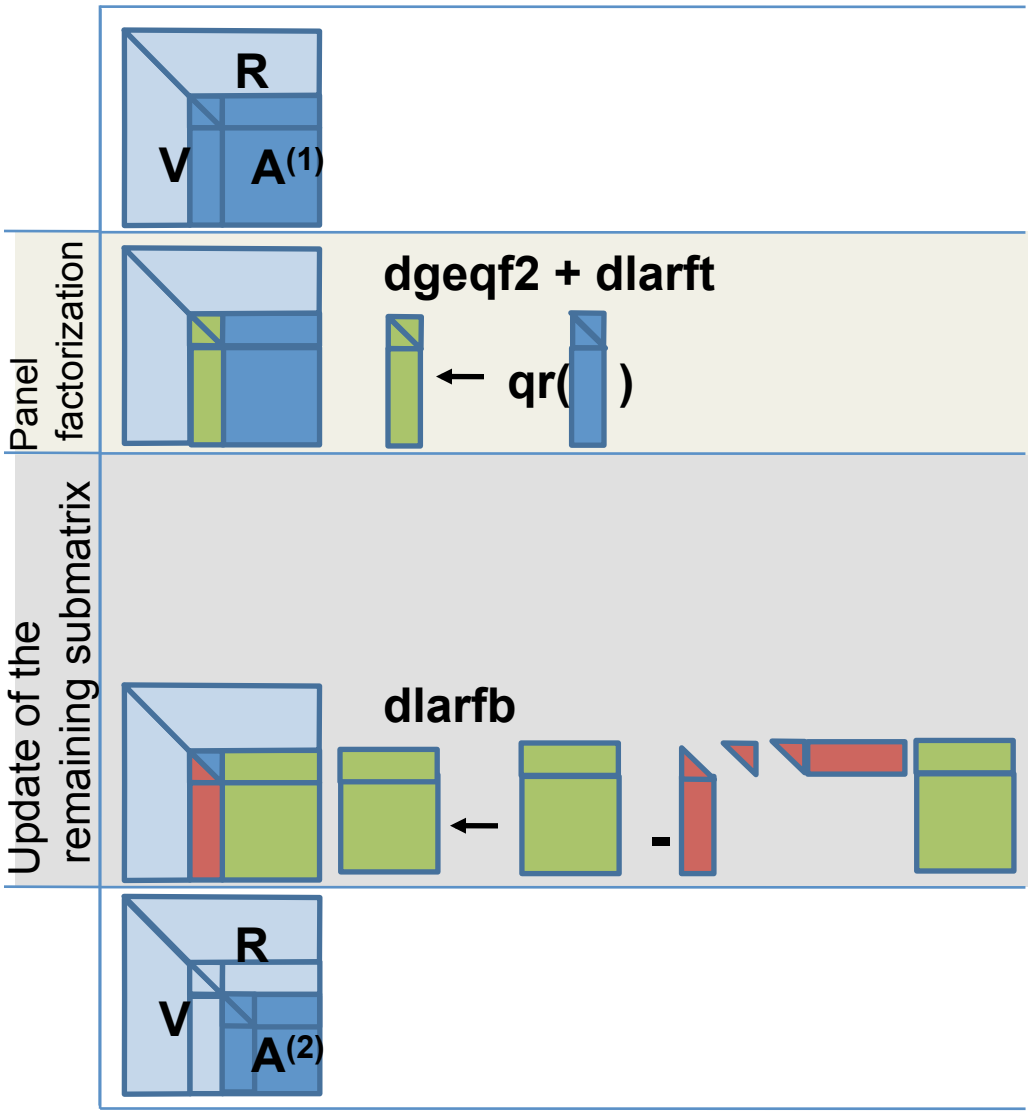
- Synchronization-reducing algorithms
  - Break Fork-Join model
- Communication-reducing algorithms
  - Use methods which have lower bound on communication
- Mixed precision methods
  - 2x speed of ops and 2x speed for data movement
- Autotuning
  - Today's machines are too complicated, build "smarts" into software to adapt to the hardware
- Fault resilient algorithms
  - Implement algorithms that can recover from failures/bit flips
- Reproducibility of results
  - Today we can't guarantee this. We understand the issues, but some of our "colleagues" have a hard time with this.

Do you remember the 80's and 90's?

Algorithms follow hardware evolution along time.		
LINPACK (80's) (Vector operations)		Rely on - Level-1 BLAS operations
LAPACK (90's) (Blocking, cache friendly)		Rely on - Level-3 BLAS operations


LAPACK

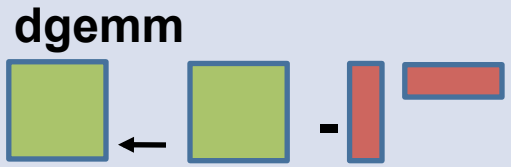
Blocked QR Factorization (LAPACK)

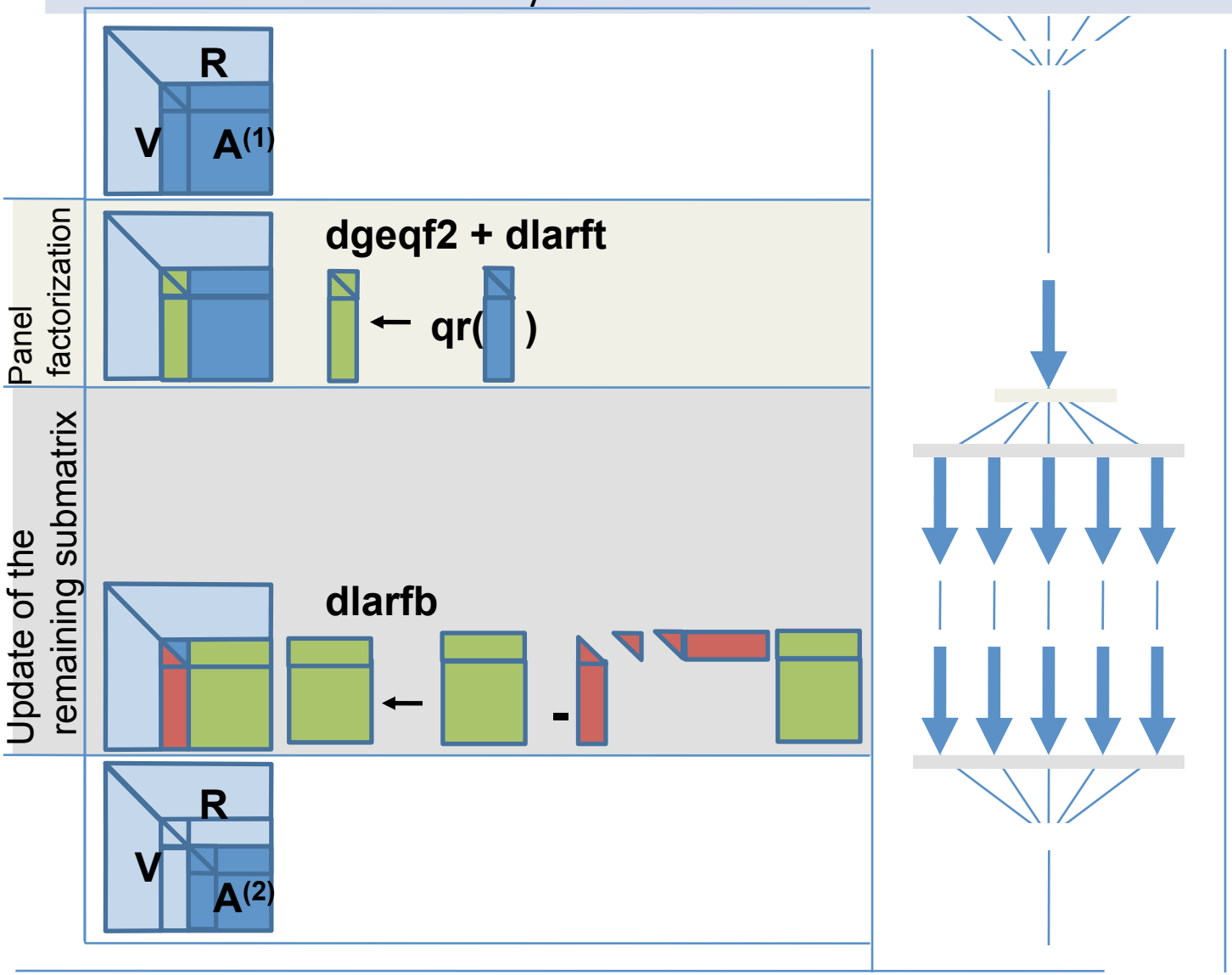


Parallelization of QR Factorization

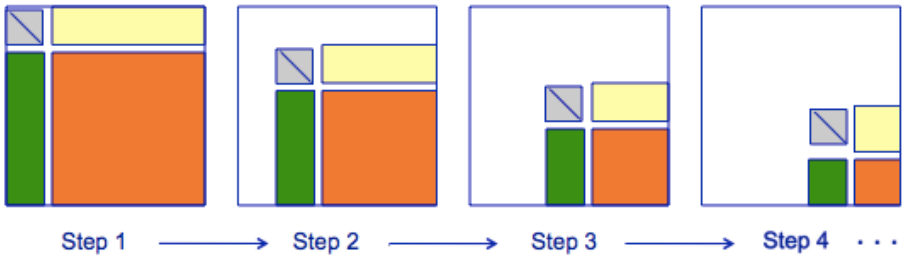
Parallelize the update:

- Easy and done in any reasonable software.
- This is the $2/3n^3$ term in the FLOPs count.
- Can be done efficiently with LAPACK+multithreaded BLAS

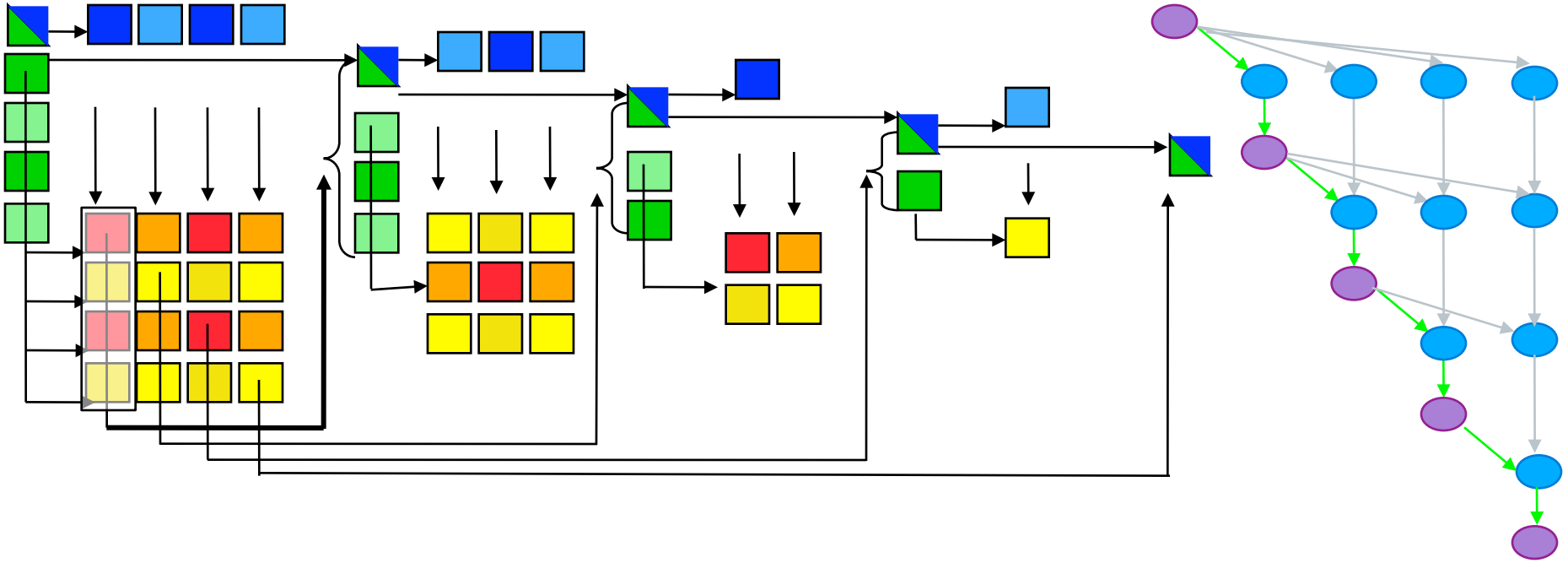




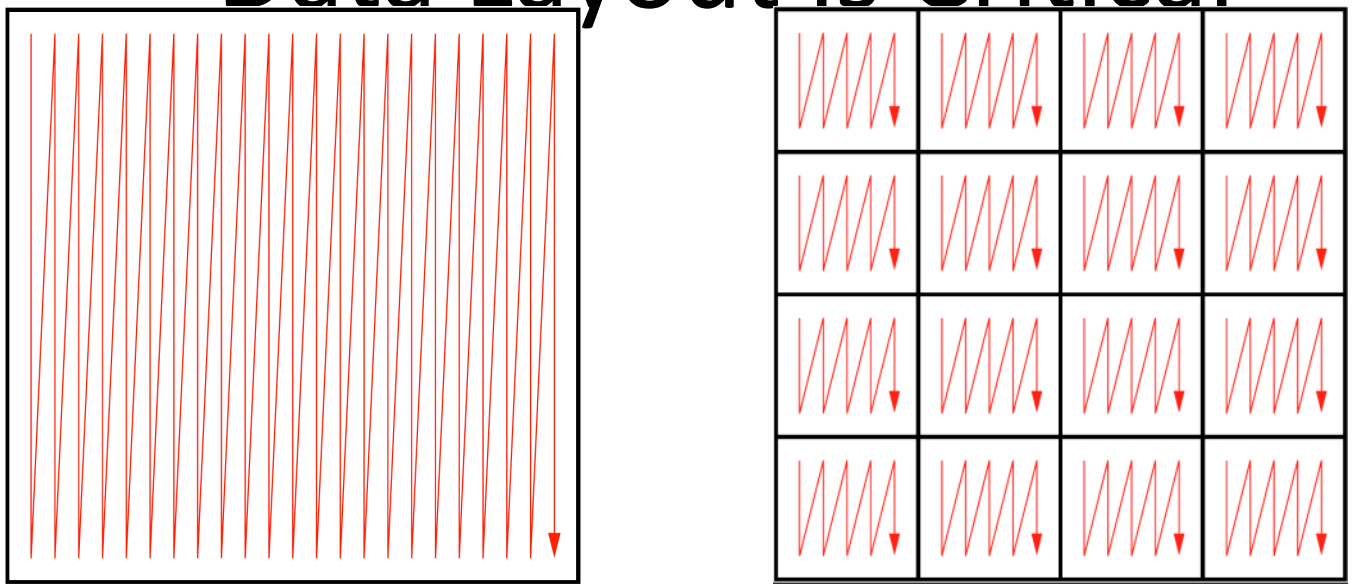
Parallel Tasks in LU/LL^T/QR



- Break into smaller tasks and remove dependencies



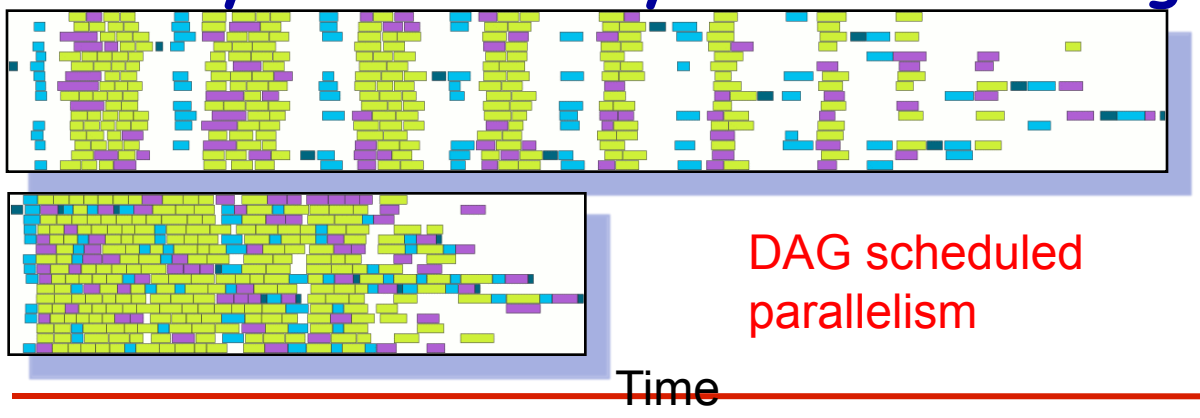
Data Layout is Critical

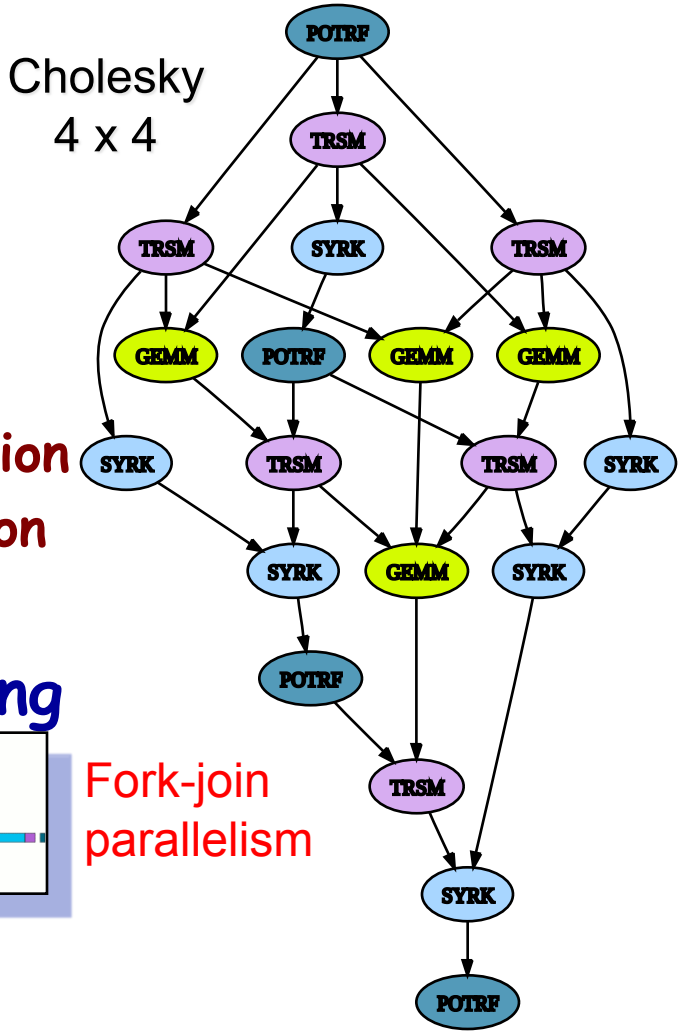


- Tile data layout where each data tile is contiguous in memory
- Decomposed into several fine-grained tasks, which better fit the memory of the small core caches

PLASMA: Parallel Linear Algebra s/w for Multicore Architectures

- **Objectives**
  - High utilization of each core
  - Scaling to large number of cores
  - Shared or distributed memory
- **Methodology**
  - Dynamic DAG scheduling
  - Split phases task generation and execution
  - Explicit parallelism/Implicit communication
  - Fine granularity / block data layout
- **Arbitrary DAG with dynamic scheduling**

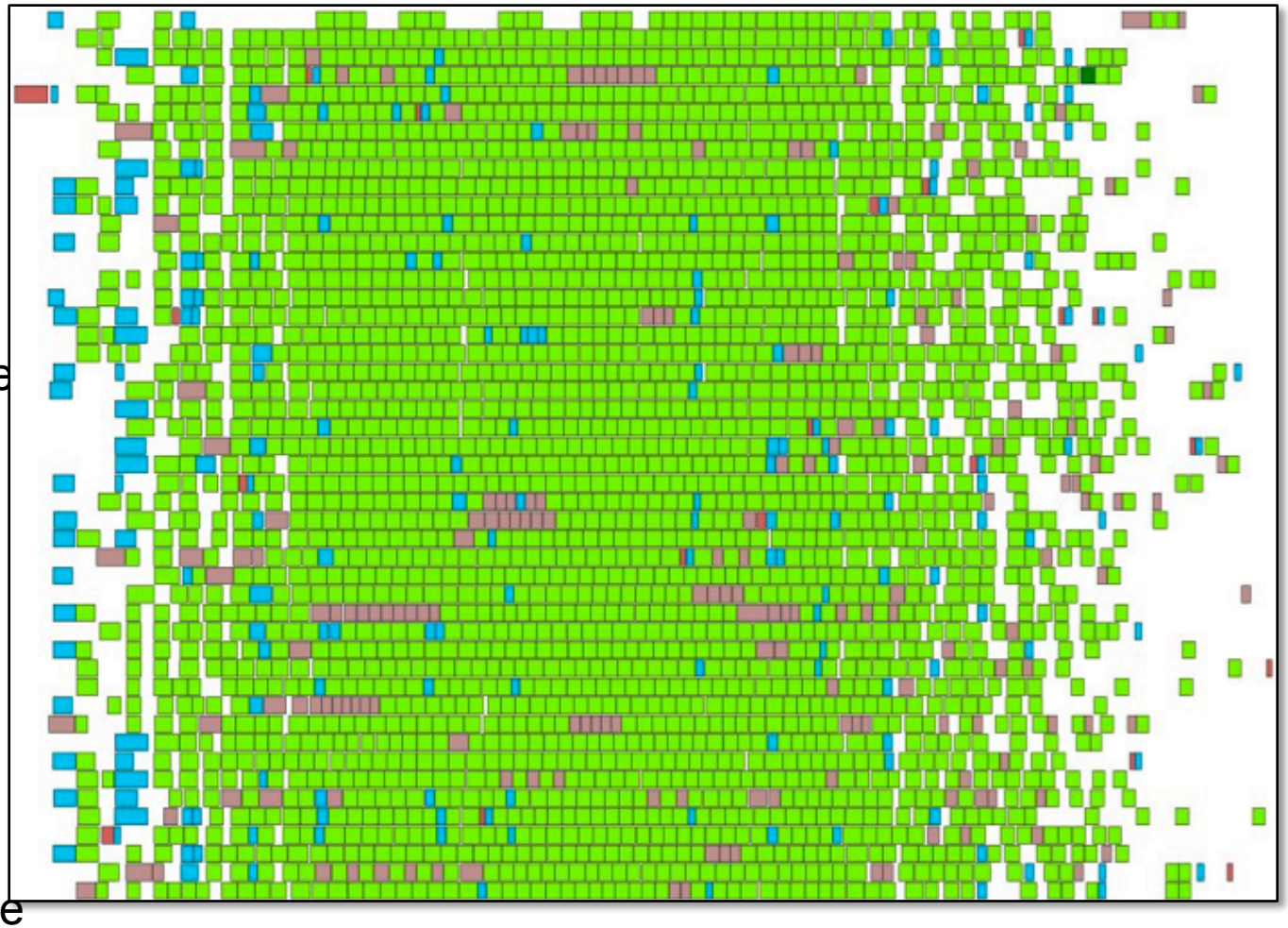




Synchronization Reducing Algorithms

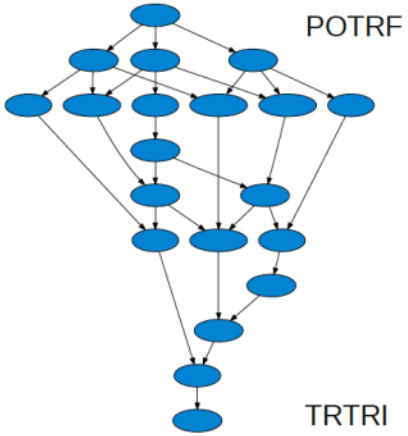
- Regular trace
- Factorization steps pipelined
- Stalling only due to natural load imbalance
- Dynamic
- Out of order execution
- Fine grain tasks
- Independent block operations

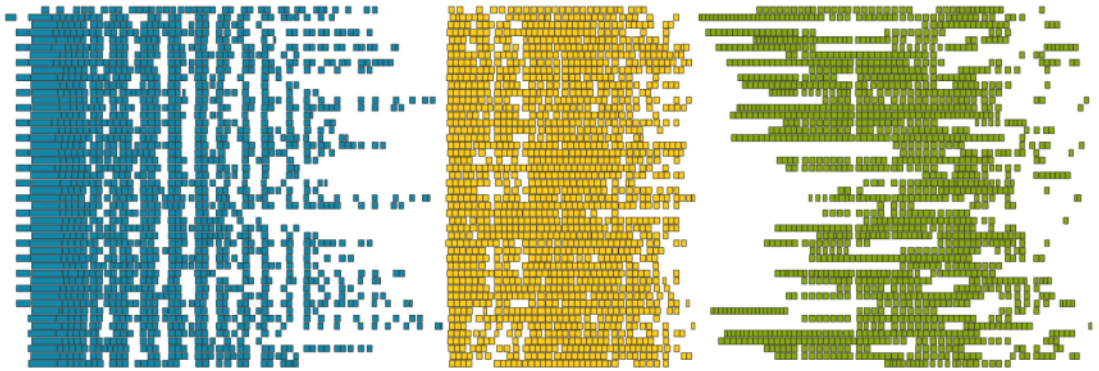
The colored area over the rectangle is the efficiency

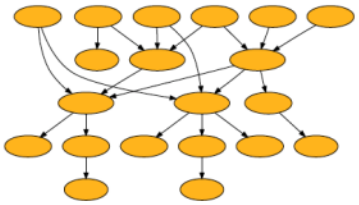


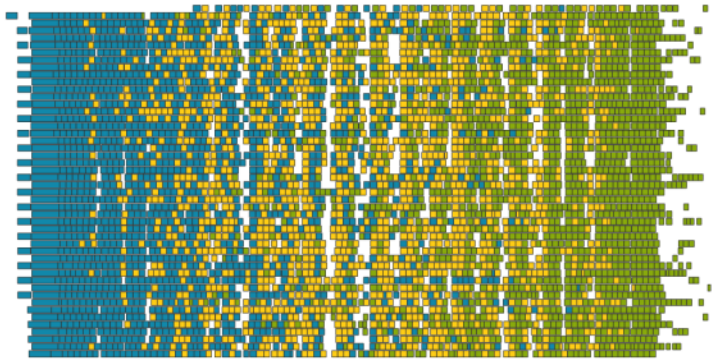
Tile QR factorization; Matrix size 4000x4000, Tile size 200
8-socket, 6-core (48 cores total) AMD Istanbul 2.8 GHz

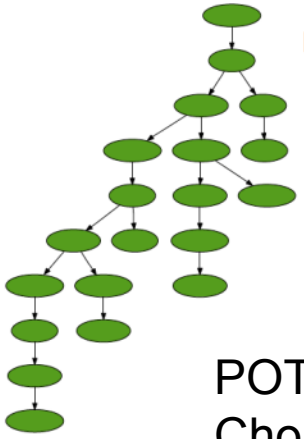
Pipelining: Cholesky Inversion
3 Steps: Factor, Invert L, Multiply L's

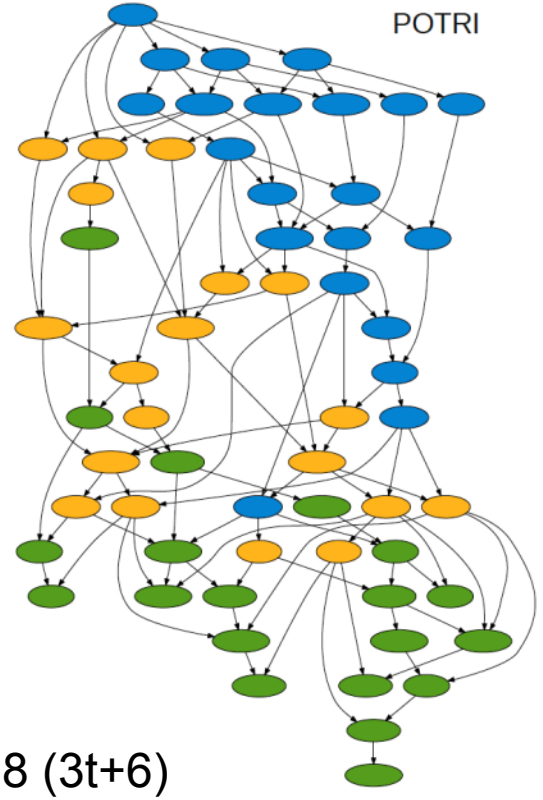












48 cores
POTRF, TRTRI and LAUUM.
The matrix is 4000 x 4000,tile size is 200 x

POTRF+TRTRI+LAUUM: 25 (7t-3)
Cholesky Factorization alone: 3t-2

Pipelined: 18 (3t+6)

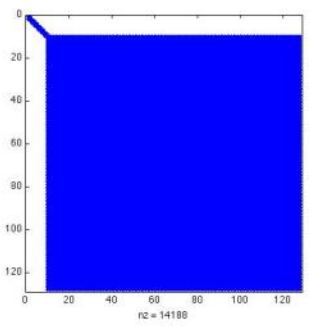
The standard Tridiagonal reduction xSYTRD

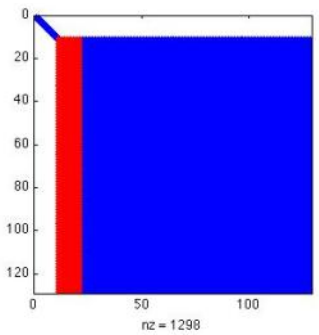
- **LAPACK xSYTRD:**

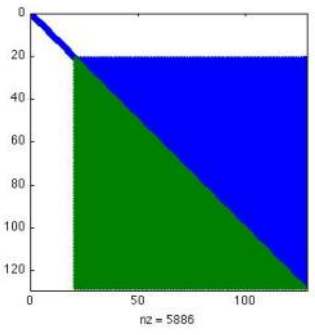
1. Apply left-right transformations Q A Q* to the panel $\begin{pmatrix} A_{22} \\ A_{32} \end{pmatrix}$
2. Update the remaining submatrix A_{33}

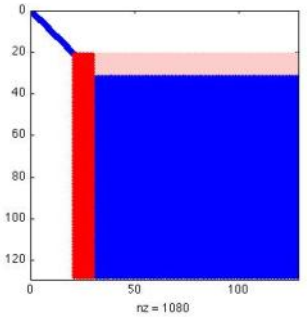
$$\begin{pmatrix} T_{11} & T_{21}^T & 0 \\ T_{21} & A_{22} & A_{32}^T \\ 0 & A_{32} & A_{33} \end{pmatrix} \equiv \begin{pmatrix} T_{11} & T_{21}^T & 0 \\ T_{21} & A_{22} & A_{32}^T \\ 0 & A_{32} & A_{33} \end{pmatrix} \Longrightarrow \begin{pmatrix} T_{11} & T_{21}^T & 0 \\ T_{21} & T_{22} & T_{23}^T \\ 0 & T_{23} & A_{33} \end{pmatrix}$$

where $A_{33} = A_{33} - YW^T - WY^T$









step *k*: Q A Q* then update ➔ step *k+1*

For the symmetric eigenvalue problem:
First stage takes:
- **90% of the time if only eigenvalues**
- **50% of the time if eigenvalues and eigenvectors**

The standard Tridiagonal reduction xSYTRD

Characteristics

1. Phase 1 requires :
   - 4 panel vector multiplications,
   - 1 symmetric matrix vector multiplication with A_{33},
   - Cost $2(n-k)^2b$ flops.
2. Phase 2 requires:
   - Symmetric update of A_{33} using SYRK,
   - Cost $2(n-k)^2b$ flops.

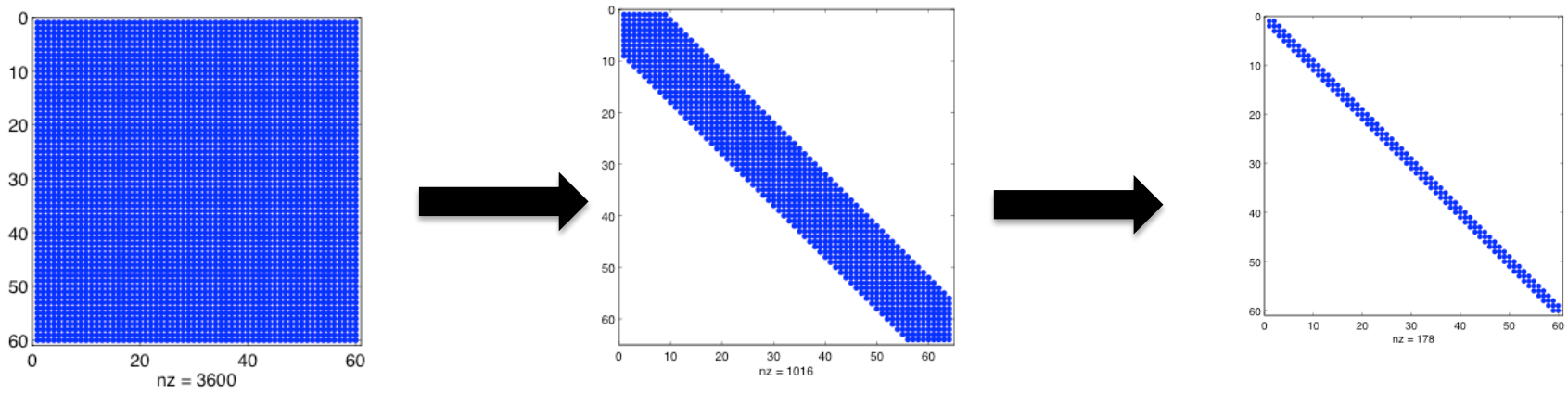
Observations

- Too many Blas-2 op,
- Relies on panel factorization,
- Total cost $4n^3/3$
- ➔Bulk sync phases,
- ➔Memory bound algorithm.

Symmetric Eigenvalue Problem

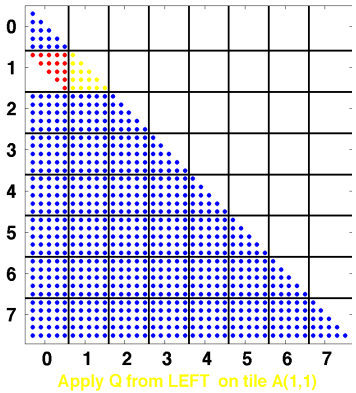
- Standard reduction algorithm are very slow on multicore.

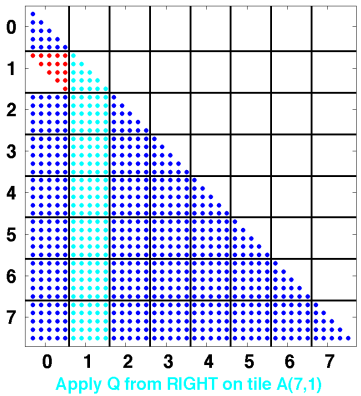
- Step1: Reduce the dense matrix to band.
  - Matrix-matrix operations, high degree of parallelism
- Step2: Bulge Chasing on the band matrix
  - by group and cache aware

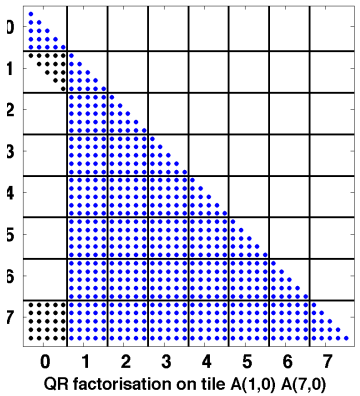


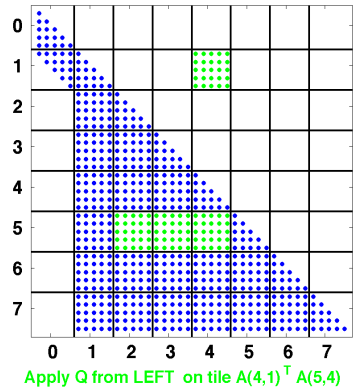
The PLASMA reduction: stage -1-

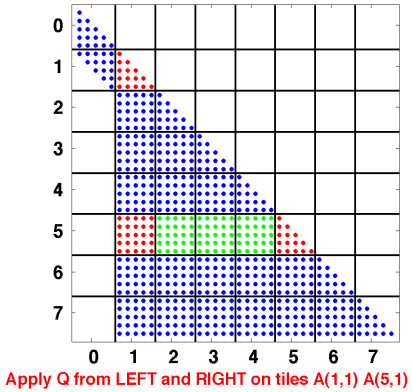
✸ Tile Band reduction:

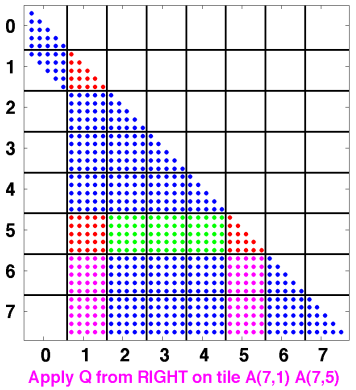




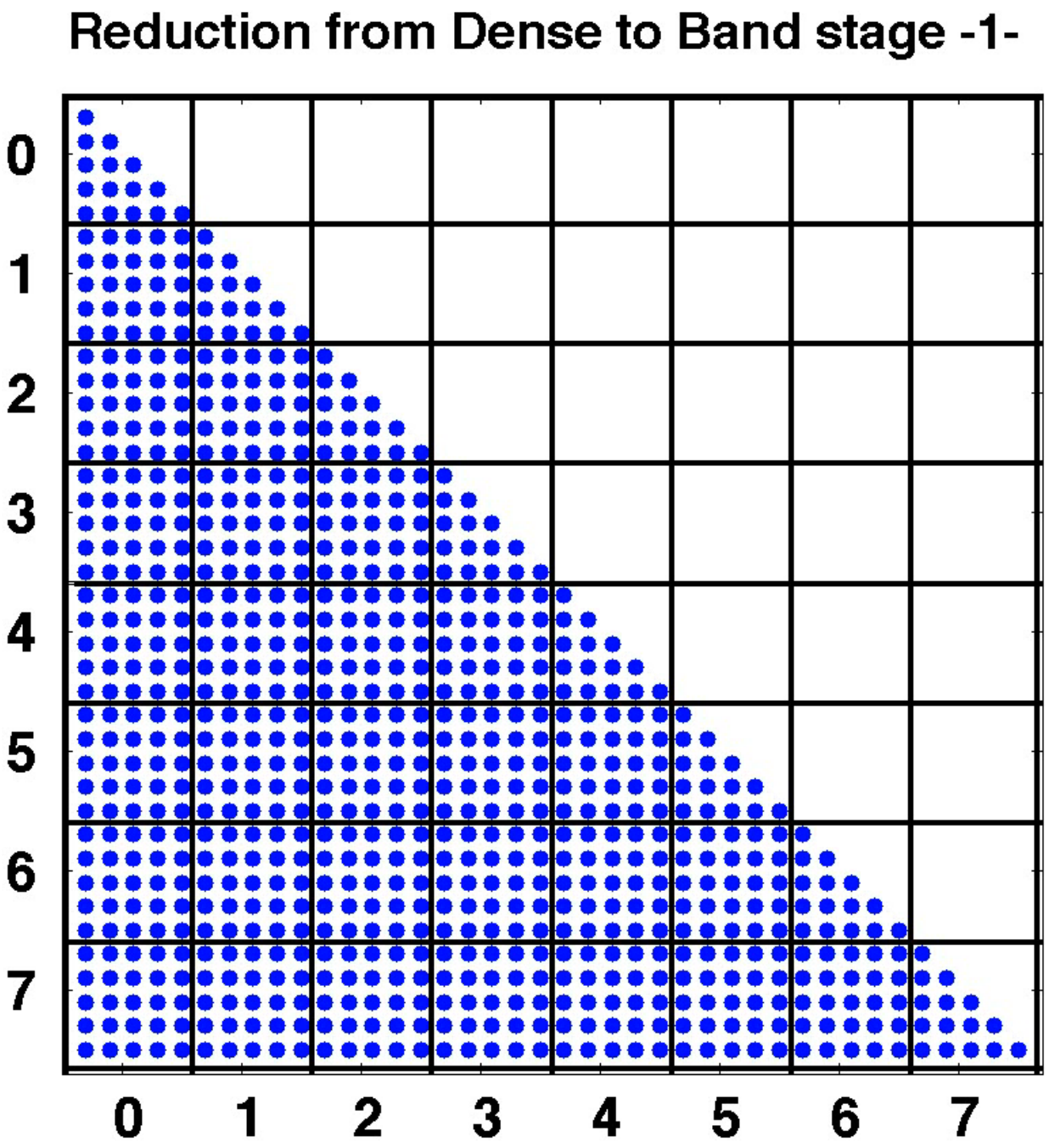




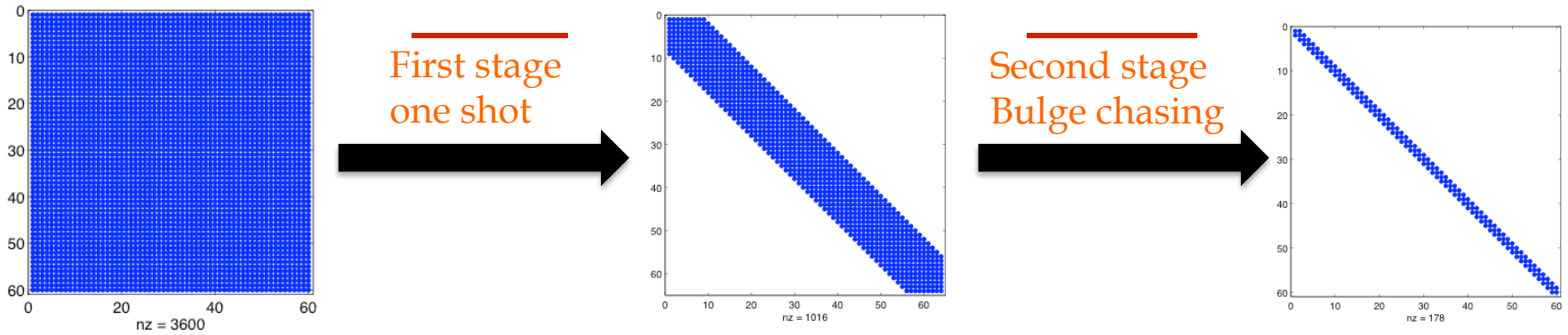





```
1:for step = 1; 2 to NT-1
2:    QR factorize
3:    apply Q from LEFT
4:    for i = step+1 to NT
5:       apply Q from RIGHT
6:    end for
7:    for k = step+2 to NT do
8:       factorize 2 tiles
9:       for j = step+2 to k-1
10:         LEFT update on 2 tiles
11:      end for
12:      apply a LEFT and
         RIGHT update diagonal
13:      for m = k+1 to NT do
14:         RIGHT updates
15:      end for
16:   end for
17:end for
```


Reduction from Dense to Band stage -1-
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7

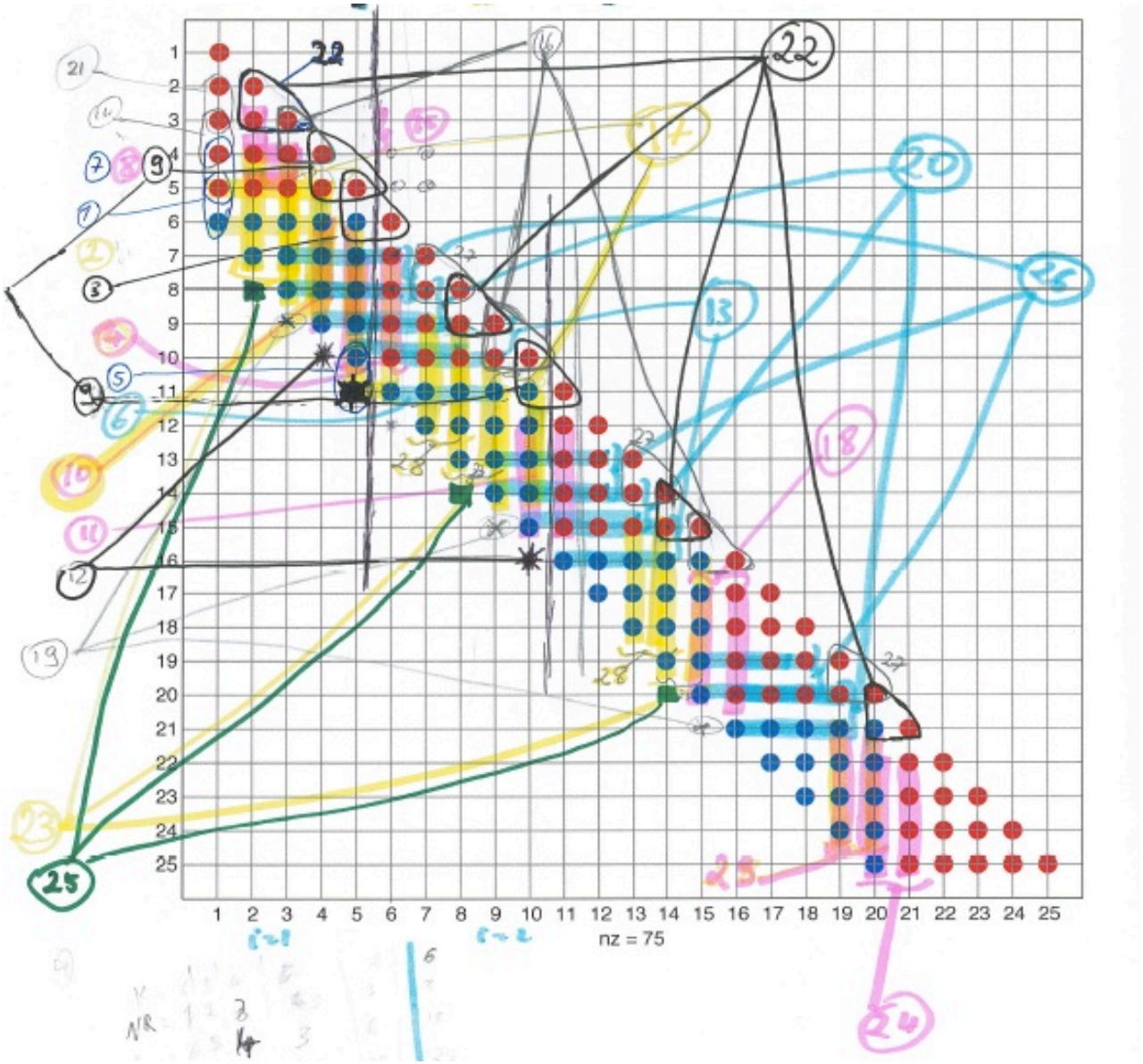
Description: the PLASMA reduction algorithms

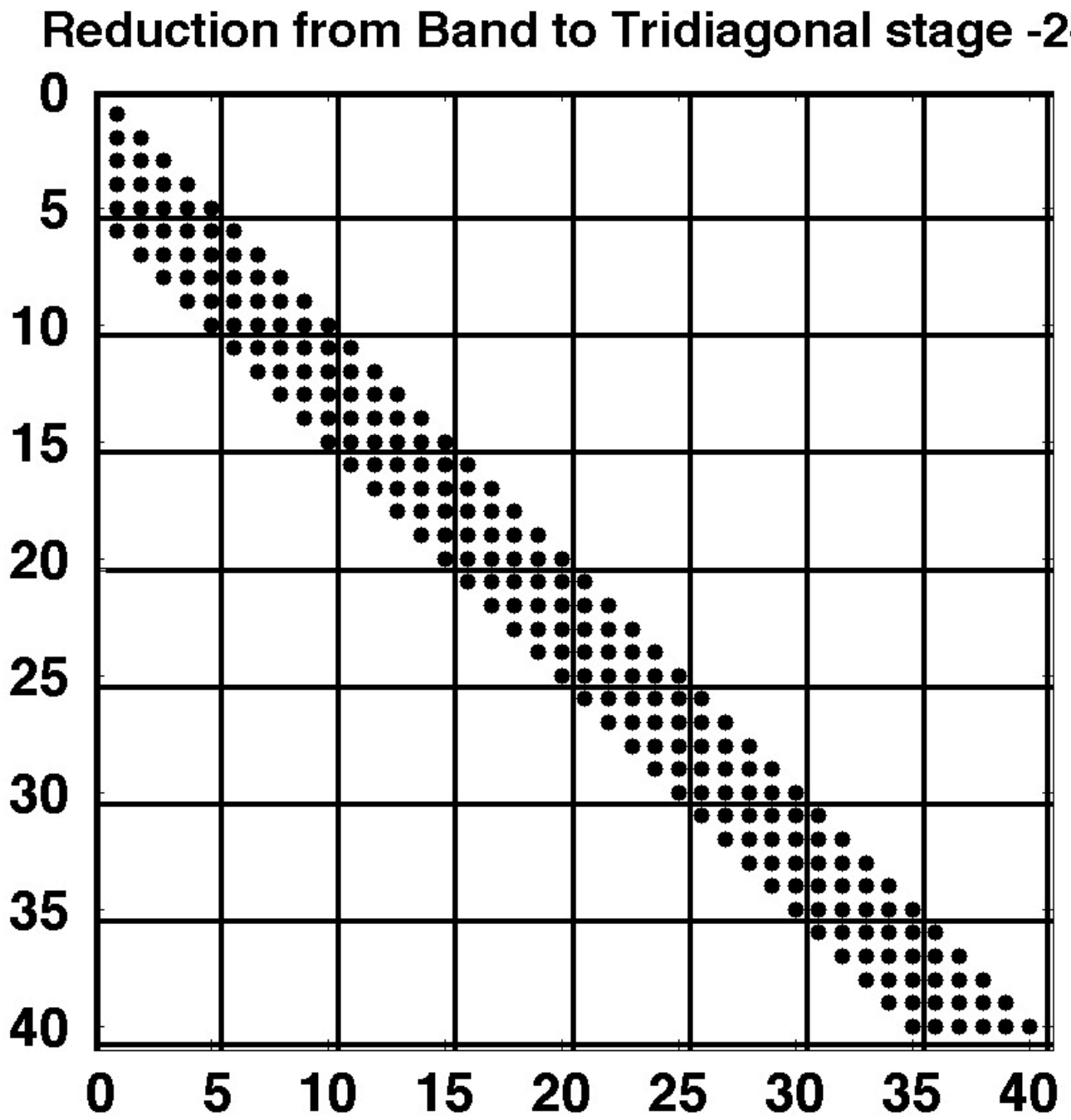


SBR-toolbox	PLASMA

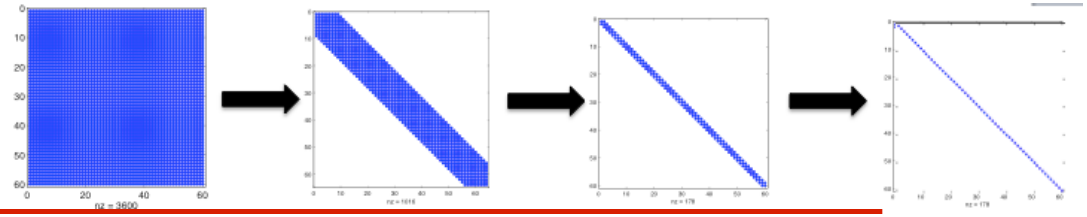
• Stage1:
  • BLAS-3
  • successive reduction
  • fork join
• Stage2:
  • BLAS-1
  • column-wise
	• Stage 1:
	  • BLAS-3
	  • one shot reduction
	  • asynchronous execution
	• Stage2:
	  • BLAS-1,
	  • element-wise, in groups
	  • asynchronous execution
	  • new cache friendly kernel

Bulge Chasing- The algorithm



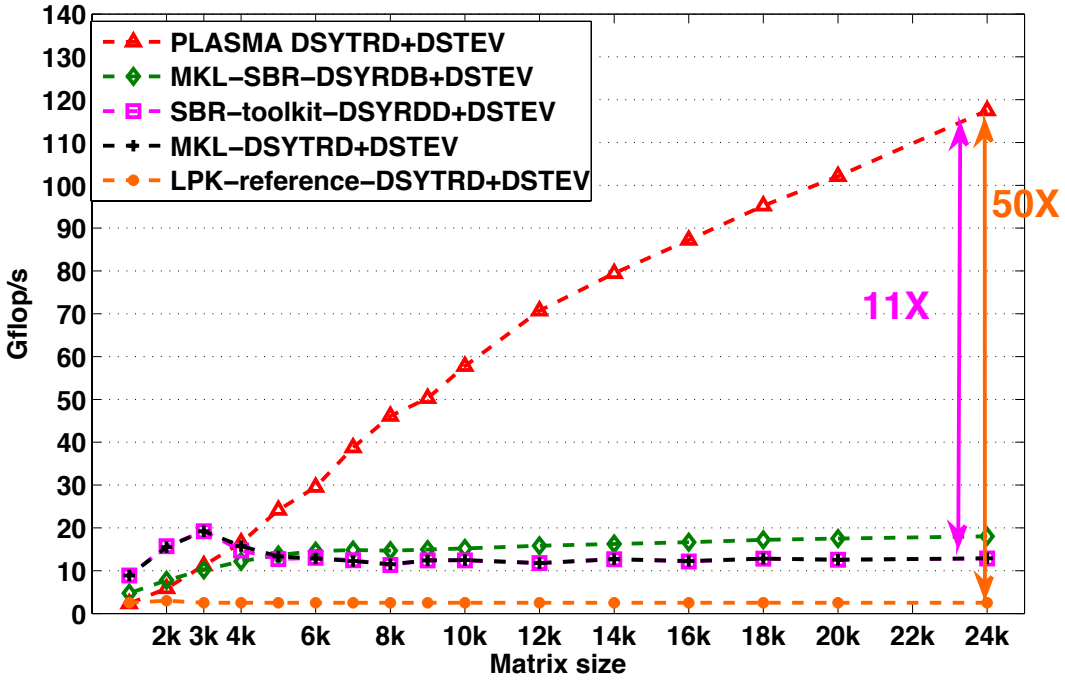

Reduction from Band to Tridiagonal stage -2-
0
5
10
15
20
25
30
35
40
0
5
10
15
20
25
30
35
40

Performance



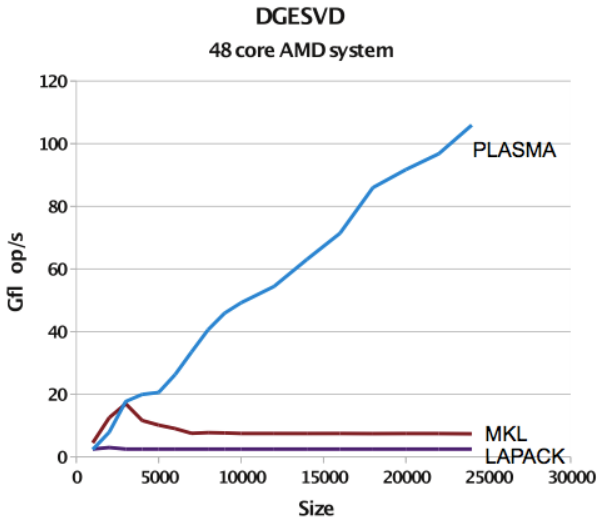
Eigenvalues

eigenvalues only



Singular Values

singular values only



Experiments on eight-socket six-core AMD Opteron 2.4 GHz processors with MKL V10.3.

- Block DAG based to banded form, then pipelined group chasing to tridiagaonal form.
- The reduction to condensed form accounts for the factor of 50 improvement over LAPACK
- Execution rates based on $4/3n^3$ ops

Mixed Precision Methods

- **Mixed precision, use the lowest precision required to achieve a given accuracy outcome**
  - **Improves runtime, reduce power consumption, lower data movement**
  - **Reformulate to find correction to solution, rather than solution; Δx rather than x.**

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$x_{i+1} - x_i = -\frac{f(x_i)}{f'(x_i)}$$

Idea Goes Something Like This…

- Exploit 32 bit floating point as much as possible.
  - Especially for the bulk of the computation
- Correct or update the solution with selective use of 64 bit floating point to provide a refined results
- Intuitively:
  - Compute a 32 bit result,
  - Calculate a correction to 32 bit result using selected higher precision and,
  - Perform the update of the 32 bit results with the correction using high precision.

Mixed-Precision Iterative Refinement

- Iterative refinement for dense systems, $Ax = b$, can work this way.

```
L U = lu(A)                                   O(n^3)
x = L\(U\b)                                   O(n^2)
r = b – Ax                                    O(n^2)
WHILE || r || not small enough
        z = L\(U\r)                           O(n^2)
        x = x + z                             O(n^1)
        r = b – Ax                            O(n^2)
END
```

  - Wilkinson, Moler, Stewart, & Higham provide error bound for SP fl pt results when using DP fl pt.

Mixed-Precision Iterative Refinement

- Iterative refinement for dense systems, $Ax = b$, can work this way.

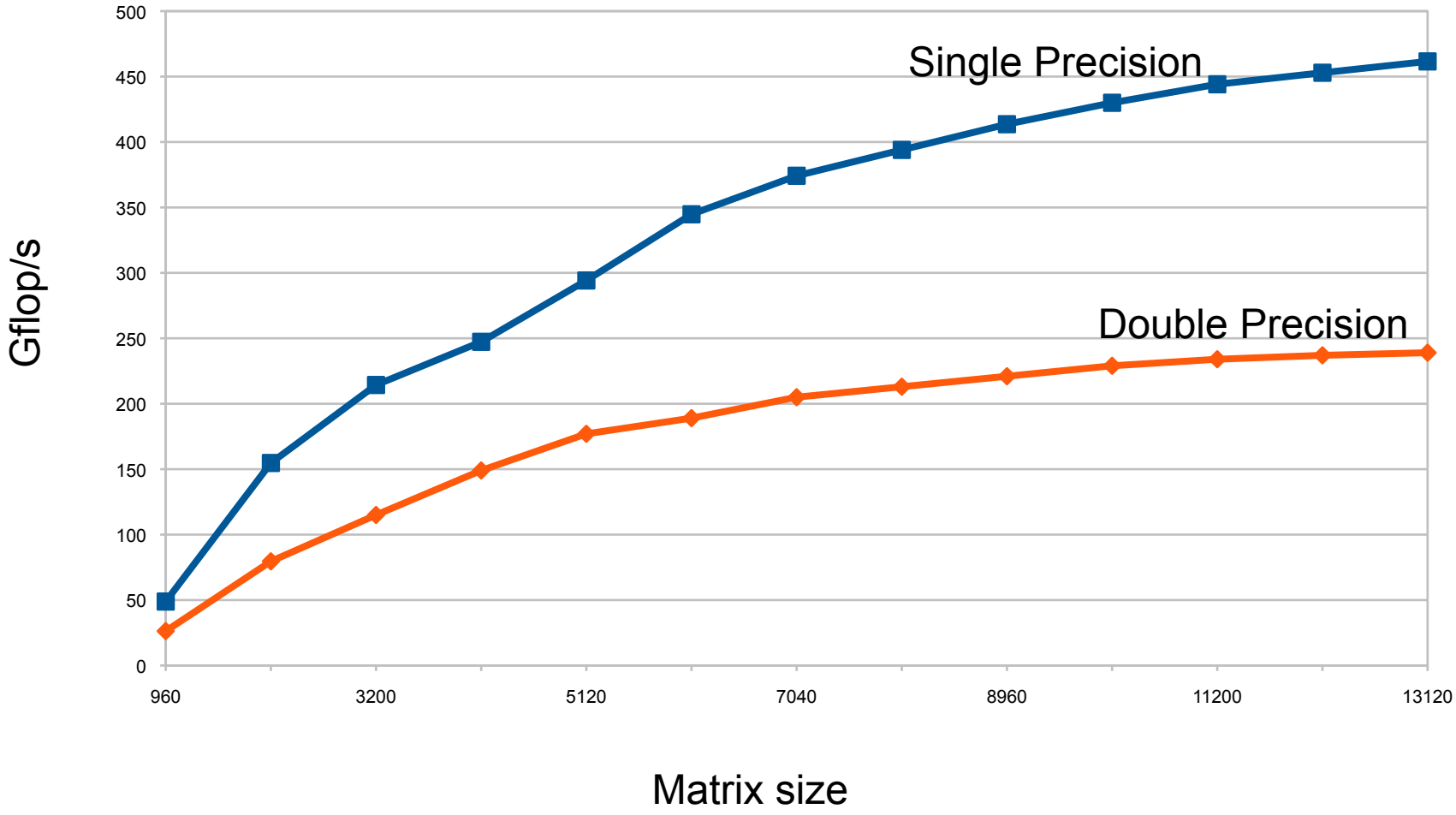
L U = lu(A)	SINGLE	$O(n^3)$
x = L\(U\b)	SINGLE	$O(n^2)$
r = b – Ax	DOUBLE	$O(n^2)$
WHILE || r || not small enough		
z = L\(U\r)	SINGLE	$O(n^2)$
x = x + z	DOUBLE	$O(n^1)$
r = b – Ax	DOUBLE	$O(n^2)$
END		

- Wilkinson, Moler, Stewart, & Higham provide error bound for SP fl pt results when using DP fl pt.
- It can be shown that using this approach we can compute the solution to 64-bit floating point precision.

- Requires extra storage, total is 1.5 times normal;
- $O(n^3)$ work is done in lower precision
- $O(n^2)$ work is done in high precision
- Problems if the matrix is ill-conditioned in sp; $O(10^8)$

Ax = b

FERMI Tesla C2050: 448 CUDA cores @ 1.15GHz
SP/DP peak is 1030 / 515 GFlop/s

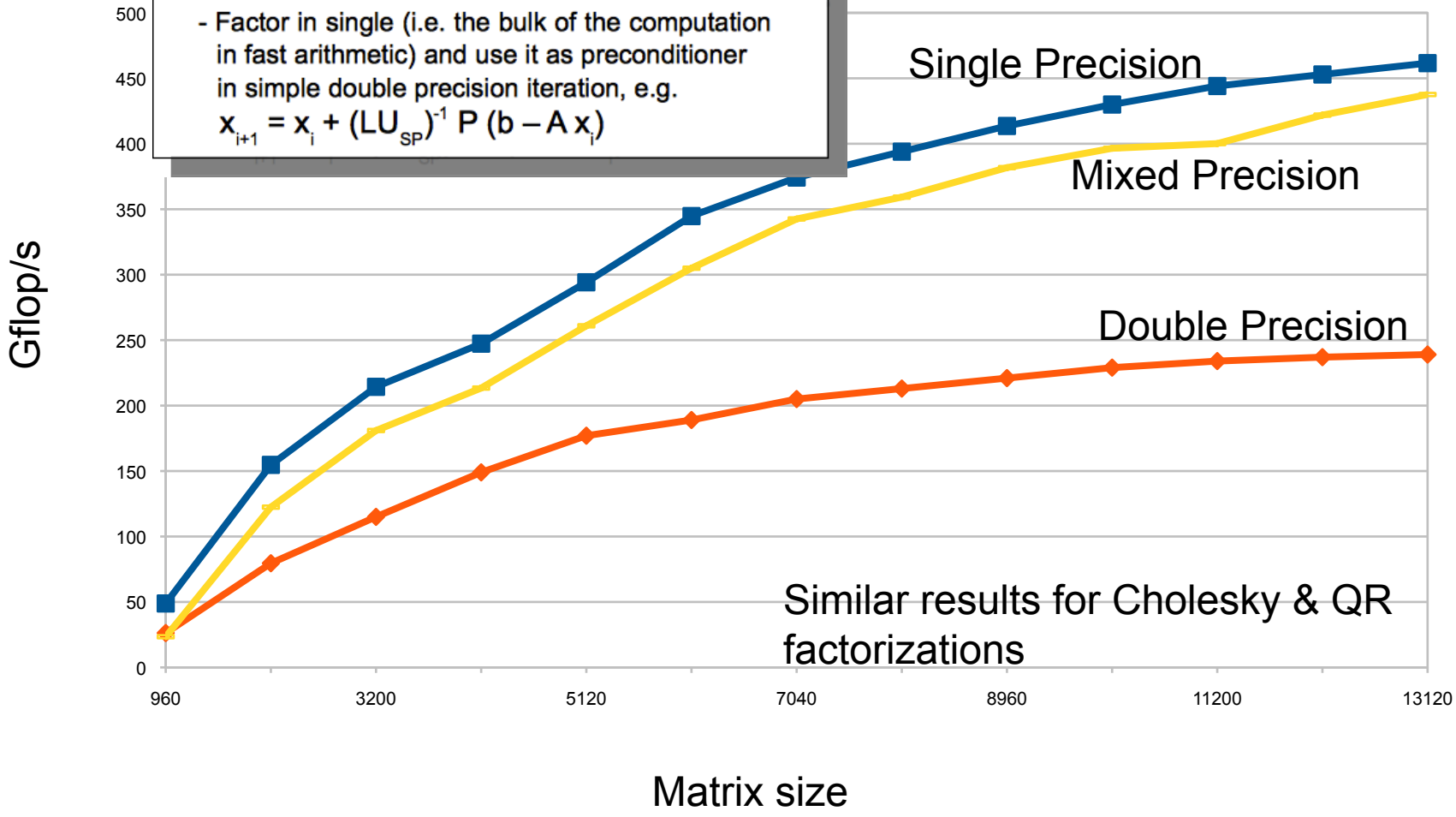


Ax = b

FERMI Tesla C2050: 448 CUDA cores @ 1.15GHz
SP/DP peak is 1030 / 515 GFlop/s

- **Direct solvers**
  - Factor and solve in working precision
- **Mixed Precision Iterative Refinement**
  - Factor in single (i.e. the bulk of the computation in fast arithmetic) and use it as preconditioner in simple double precision iteration, e.g.

$$x_{i+1} = x_i + (LU_{SP})^{-1} P (b - A x_i)$$



Quadruple Precision

n	Quad Precision Ax = b	Iter. Refine. DP to QP	
	time (s)	time (s)	Speedup
100	0.29	0.03	9.5
200	2.27	0.10	20.9
300	7.61	0.24	30.5
400	17.8	0.44	40.4
500	34.7	0.69	49.7
600	60.1	1.01	59.0
700	94.9	1.38	68.7
800	141.	1.83	77.3
900	201.	2.33	86.3
1000	276.	2.92	94.8

Intel Xeon 3.2 GHz

Reference implementation of the quad precision BLAS

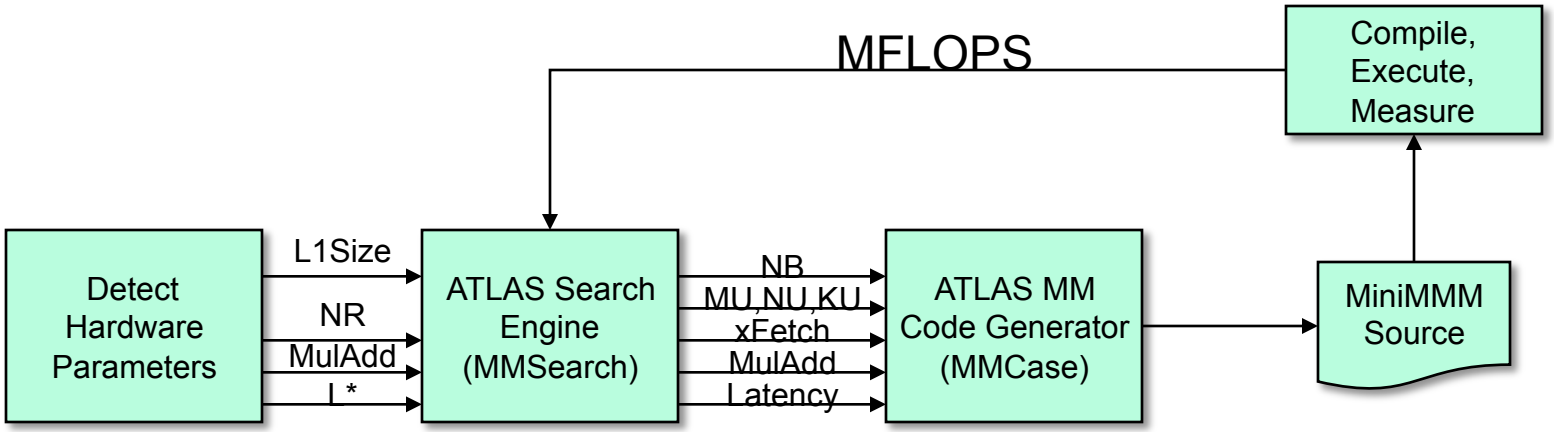
Accuracy: 10^{-32}

No more than 3 steps of iterative refinement are needed.

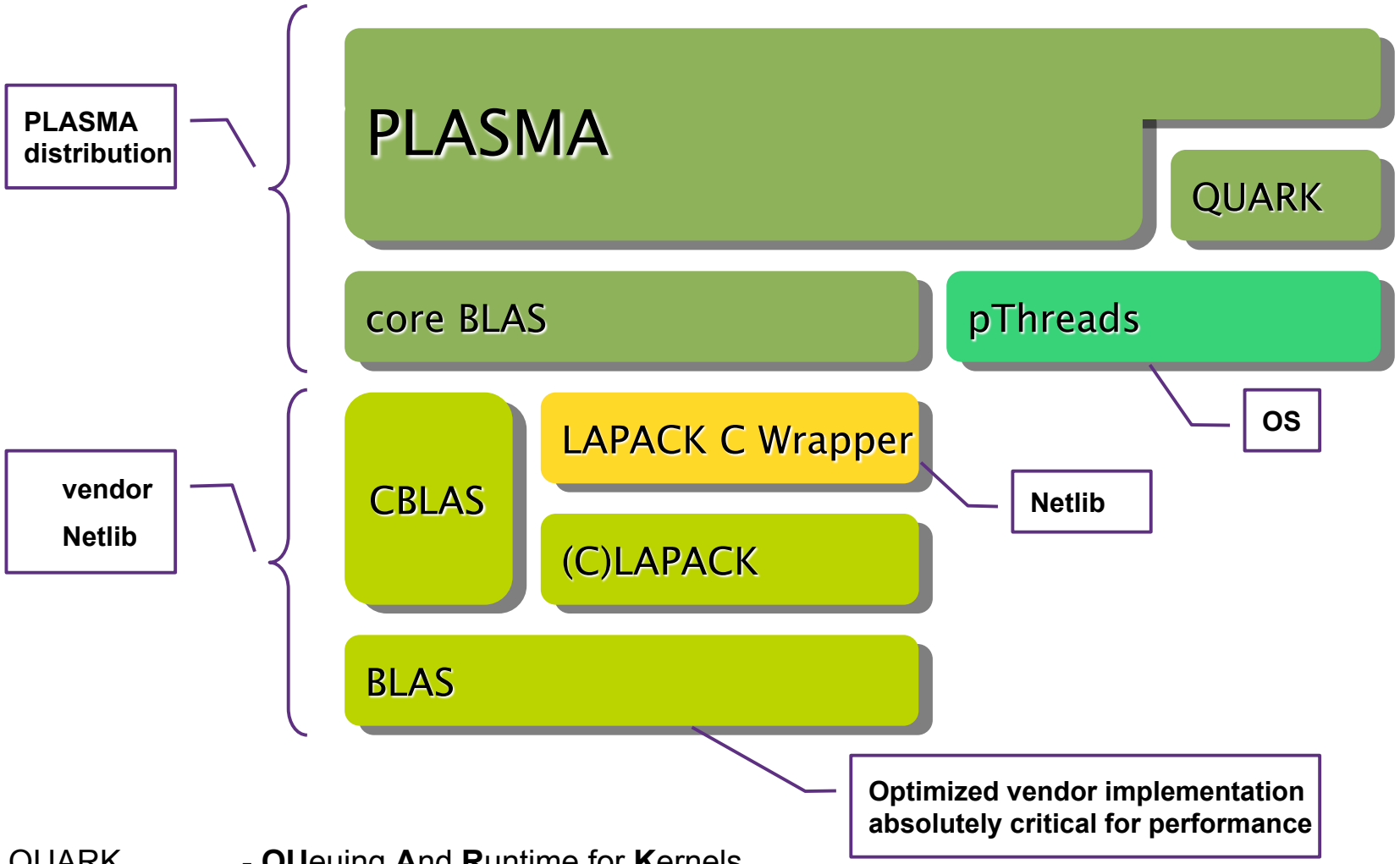
- Variable precision factorization (with say < 32 bit precision) plus 64 bit refinement produces 64 bit accuracy

How to Deal with Complexity?

- Many parameters in the code needs to be optimized.
- Software adaptivity is the key for applications to effectively use available resources whose complexity is exponentially increasing



PLASMA: Software Stack



QUARK - **QU**euing **A**nd **R**untime for **K**ernels

LAPACK - **L**inear **A**lgebra **PACK**age

BLAS - **B**asic **L**inear **A**lgebra **S**ubroutines

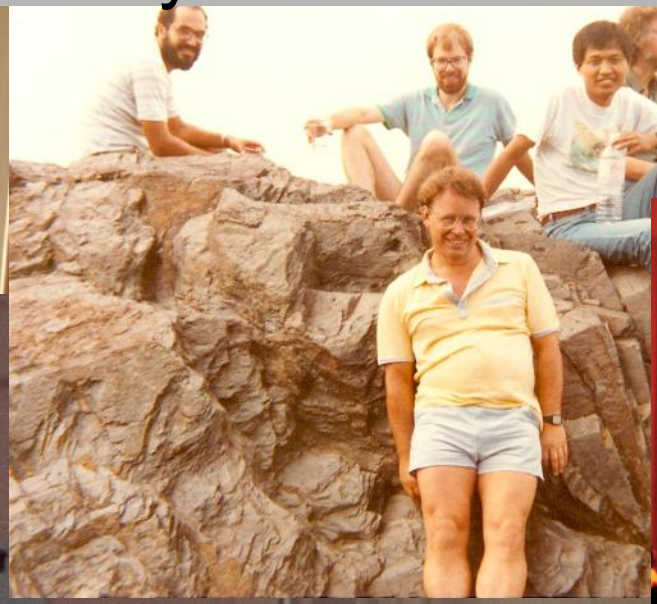
“Friends are the family we choose for ourselves”

“Friends are the family we choose for ourselves”

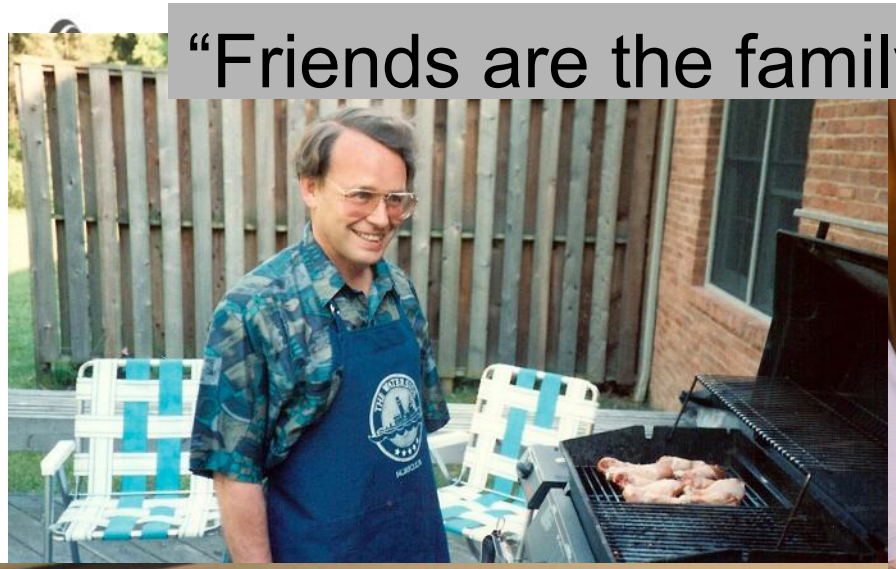
“Friends are the family we choose for ourselves”

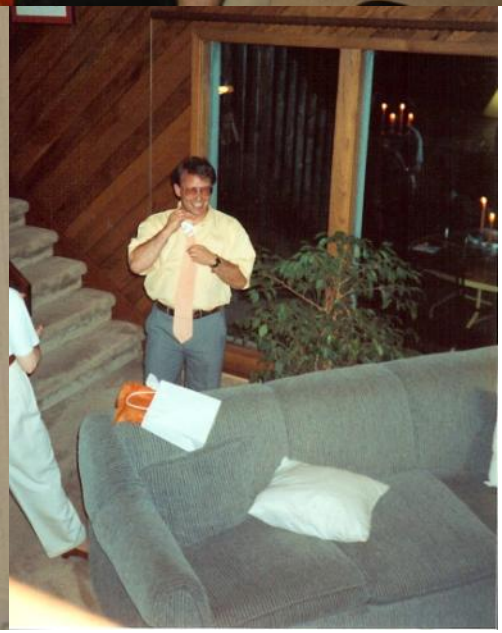
“Friends are the family we choose for ourselves”

“Friends are the family we choose for ourselves”



“Friends are the family we choose for ourselves”





Happy 70th Birthday Sven!