Multicore performance of the Block Low-Rank multifrontal factorization

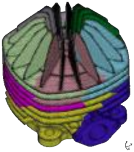
P. Amestoy[*,1] A. Buttari[*,2] J.-Y. L'Excellent[†,3] T. Mary[*,4]

[*]Université de Toulouse [†]ENS Lyon
[1]INPT-IRIT [2]CNRS-IRIT [3]INRIA-LIP [4]UPS-IRIT

Journée Lyon Calcul, Lyon, December 15, 2016

Introduction

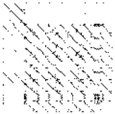
Sparse direct solvers



Discretization of a physical problem
(e.g. Code_Aster, finite elements)

$\Downarrow$

A X = **B**, **A** large and sparse, **B** dense or sparse
Sparse direct methods : $\mathbf{A} = \mathbf{LU}$ ($\mathbf{LDL^T}$)

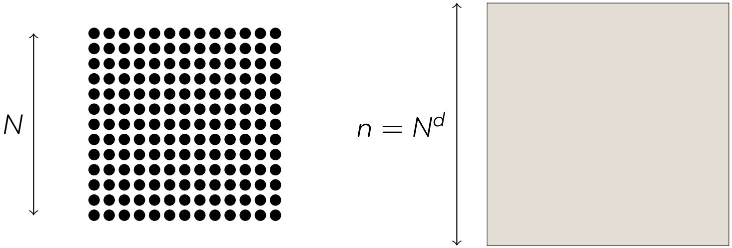


Often a significant part of simulation cost

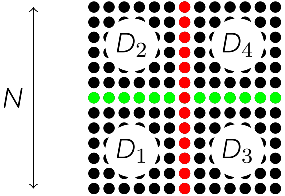
Objective discussed in this talk:
how to reduce the cost of sparse direct solvers?

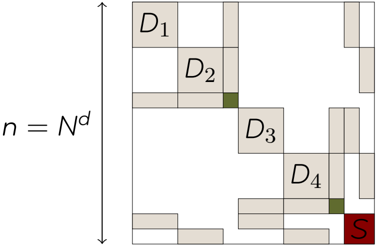
Focus on multicore architectures

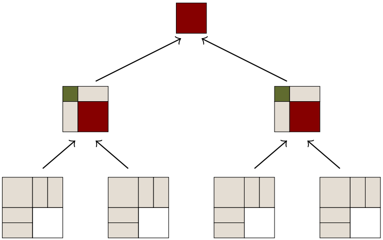
Multifrontal Factorization with Nested Dissection



Multifrontal Factorization with Nested Dissection



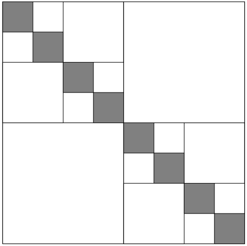




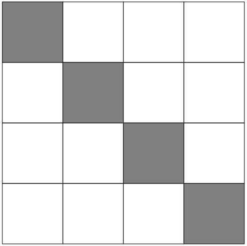
3D problem complexity

→ Flops: $O(n^2)$, mem: $O(n^{4/3})$

$\mathcal{H}$ and BLR matrices

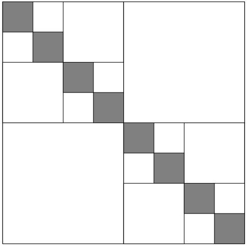


$\mathcal{H}$-matrix

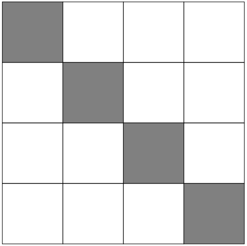


BLR matrix

$\mathcal{H}$ and BLR matrices



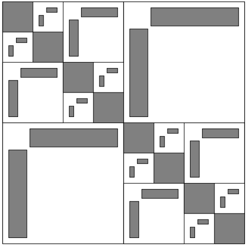
$\mathcal{H}$-matrix



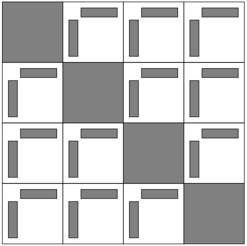
BLR matrix

A block B represents the interaction between two subdomains. If they have a small diameter and are far away their interaction is weak $\Rightarrow$ rank is low.

$\mathcal{H}$ and BLR matrices



$\mathcal{H}$-matrix



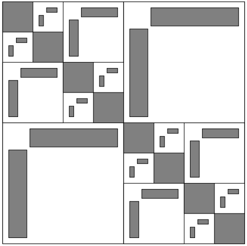
BLR matrix

A block B represents the interaction between two subdomains. If they have a small diameter and are far away their interaction is weak $\Rightarrow$ rank is low.

$$\tilde{B} = XY^T \text{ such that } \operatorname{rank}(\tilde{B}) = k_\varepsilon \text{ and } \|B - \tilde{B}\| \leq \varepsilon$$

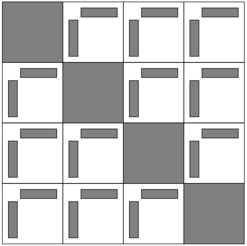
If $k_\varepsilon \ll \text{size}(B) \Rightarrow$ memory and flops can be reduced with a controlled loss of accuracy ($\leq \varepsilon$)

$\mathcal{H}$ and BLR matrices



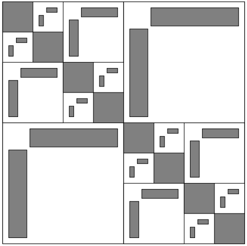
$\mathcal{H}$-matrix

- Theoretical complexity can be as low as $O(n)$
- Complex, hierarchical structure

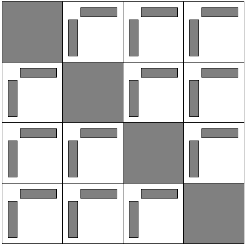


BLR matrix

- Theoretical complexity can be as low as $O(n^{4/3})$
- Simple structure



$\mathcal{H}$-matrix



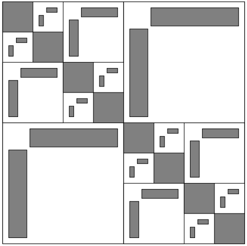
BLR matrix

- Theoretical complexity can be as low as $O(n)$
- Complex, hierarchical structure

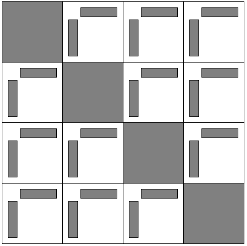
- Theoretical complexity can be as low as $O(n^{4/3})$
- Simple structure

Find a good comprise between complexity and performance

$\mathcal{H}$ and BLR matrices



$\mathcal{H}$-matrix



BLR matrix

- Theoretical complexity can be as low as $O(n)$
- Complex, hierarchical structure

- Theoretical complexity can be as low as $O(n^{4/3})$
- Simple structure

Find a good comprise between complexity and performance

$\Rightarrow$ Ongoing collaboration with STRUMPACK team (LBNL) to compare BLR and hierarchical formats

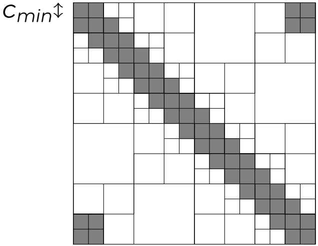
Complexity of the BLR factorization

$\mathcal{H}$ vs. BLR complexity

Until recently, BLR complexity was unknown.
Can we use $\mathcal{H}$ theory on BLR matrices?

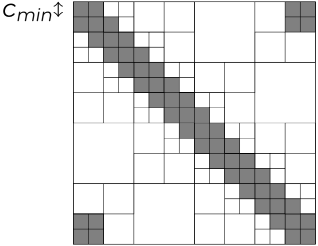
$\mathcal{H}$ vs. BLR complexity

Until recently, BLR complexity was unknown.
Can we use $\mathcal{H}$ theory on BLR matrices?



$\mathcal{H}$ vs. BLR complexity

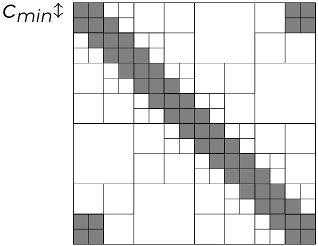
Until recently, BLR complexity was unknown.
Can we use $\mathcal{H}$ theory on BLR matrices?



Complexity mainly depends on r_{max}, the maximal rank of the blocks
With $\mathcal{H}$ partitioning, r_{max} is small

$\mathcal{H}$ vs. BLR complexity

Until recently, BLR complexity was unknown.
Can we use $\mathcal{H}$ theory on BLR matrices?

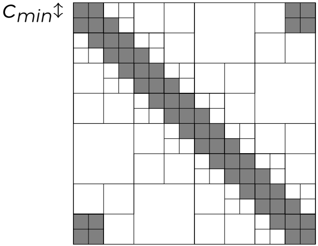


Complexity mainly depends on r_{max}, the maximal rank of the blocks
With $\mathcal{H}$ partitioning, r_{max} is small

- Problem: in $\mathcal{H}$ formalism, the maxrank of the blocks of a BLR matrix is $r_{max} = b$ (due to full-rank blocks)

$\mathcal{H}$ vs. BLR complexity

Until recently, BLR complexity was unknown.
Can we use $\mathcal{H}$ theory on BLR matrices?



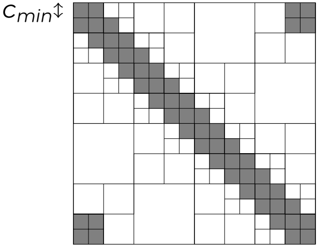
Complexity mainly depends on r_{max}, the maximal rank of the blocks

With $\mathcal{H}$ partitioning, r_{max} is small

- Problem: in $\mathcal{H}$ formalism, the maxrank of the blocks of a BLR matrix is $r_{max} = b$ (due to full-rank blocks)
- $\mathcal{H}$ theory applied to BLR does not give a satisfying result

$\mathcal{H}$ vs. BLR complexity

Until recently, BLR complexity was unknown.
Can we use $\mathcal{H}$ theory on BLR matrices?



Complexity mainly depends on r_{max}, the maximal rank of the blocks
With $\mathcal{H}$ partitioning, r_{max} is small

- Problem: in $\mathcal{H}$ formalism, the maxrank of the blocks of a BLR matrix is $r_{max} = b$ (due to full-rank blocks)
- $\mathcal{H}$ theory applied to BLR does not give a satisfying result
- Solution: extend the theory by bounding the number of full-rank blocks
  - Amestoy, Buttari, L'Excellent, and Mary. *On the Complexity of the Block Low-Rank Multifrontal Factorization*, under review, SIAM SISC, 2016.

Complexity of multifrontal BLR factorization

	operations (OPC)		factor size (NNZ)	
	$r = O(1)$	$r = O(N)$	$r = O(1)$	$r = O(N)$
FR	$O(n^2)$	$O(n^2)$	$O(n^{\frac{4}{3}})$	$O(n^{\frac{4}{3}})$
BLR	$O(n^{\frac{4}{3}}) - O(n^{\frac{5}{3}})$	$O(n^{\frac{5}{3}}) - O(n^{\frac{11}{6}})$	$O(n \log n)$	$O(n^{\frac{7}{6}} \log n)$
$\mathcal{H}$	$O(n^{\frac{4}{3}})$	$O(n^{\frac{5}{3}})$	$O(n)$	$O(n^{\frac{7}{6}})$
$\mathcal{H}$ (fully structured)	$O(n)$	$O(n^{\frac{4}{3}})$	$O(n)$	$O(n^{\frac{7}{6}})$

in the 3D case (similar analysis possible for 2D)

Important properties: with both $r = O(1)$ or $r = O(N)$

- Complexity depends on how the BLR factorization is performed
- The BLR complexity exponent is always lower than the FR one
- The best BLR complexity is not so far from the $\mathcal{H}$-case

Complexity of multifrontal BLR factorization

	operations (OPC)		factor size (NNZ)	
	$r = O(1)$	$r = O(N)$	$r = O(1)$	$r = O(N)$
FR	$O(n^2)$	$O(n^2)$	$O(n^{\frac{4}{3}})$	$O(n^{\frac{4}{3}})$
BLR	$O(n^{\frac{4}{3}}) - O(n^{\frac{5}{3}})$	$O(n^{\frac{5}{3}}) - O(n^{\frac{11}{6}})$	$O(n \log n)$	$O(n^{\frac{7}{6}} \log n)$
$\mathcal{H}$	$O(n^{\frac{4}{3}})$	$O(n^{\frac{5}{3}})$	$O(n)$	$O(n^{\frac{7}{6}})$
$\mathcal{H}$ (fully structured)	$O(n)$	$O(n^{\frac{4}{3}})$	$O(n)$	$O(n^{\frac{7}{6}})$

in the 3D case (similar analysis possible for 2D)

Important properties: with both $r = O(1)$ or $r = O(N)$

- Complexity depends on how the BLR factorization is performed
- The BLR complexity exponent is always lower than the FR one
- The best BLR complexity is not so far from the $\mathcal{H}$-case

How to convert complexity reduction into performance gain?

$\Rightarrow$ answer in the rest of this talk

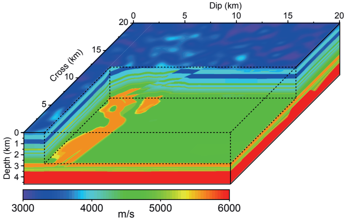
Experimental setting

Experimental Setting: Machines

Experiments are done on the shared-memory machines of the LIP laboratory of Lyon:

1. **`brunch`**
   - Four Intel(r) 24-cores Broadwell @ 2,2 GHz
   - Peak per core is 35.2 GF/s
   - Total memory is 1.5 TB

2. **`grunch`**
   - Two Intel(r) 14-cores Haswell @ 2,3 GHz
   - Peak per core is 36.8 GF/s
   - Total memory is 768 GB



3D Seismic Modeling

Helmholtz equation

Single complex (c) arithmetic

Unsymmetric *LU* factorization

Required accuracy: $\varepsilon = 10^{-3}$

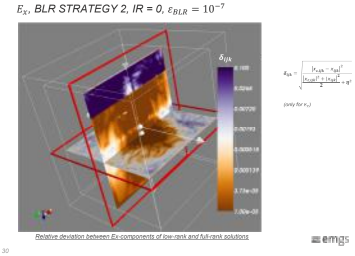
Credits: SEISCOPE

matrix	n	nnz	flops	storage
5Hz	2.9M	70M	65.0 TF	59.7 GB
7Hz	7.2M	177M	404.2 TF	205.0 GB
10Hz	17.2M	446M	2.6 PF	710.8 GB

Full-Rank statistics

- Amestoy, Brossier, Buttari, L'Excellent, Mary, Métivier, Miniussi, and Operto. *Fast 3D frequency-domain full waveform inversion with a parallel Block Low-Rank multifrontal direct solver: application to OBC data from the North Sea*, Geophysics, 2016.

Experimental Setting: Matrices (2/3)



3D Electromagnetic Modeling

Maxwell equation

Double complex (z) arithmetic

Symmetric LDL^T factorization

Required accuracy: $\varepsilon = 10^{-7}$

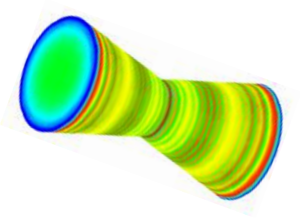
Credits: EMGS

matrix	n	nnz	flops	storage
E3	2.9M	37M	57.9 TF	77.5 GB
S3	3.3M	43M	78.0 TF	94.6 GB
E4	17.4M	226M	1.8 PF	837.0 GB
S4	20.6M	266M	2.6 PF	1.0 TB

Full-Rank statistics

- Shantsev, Jaysaval, de la Kethulle de Ryhove, Amestoy, Buttari, L'Excellent, and Mary. *Large-scale 3D EM modeling with a Block Low-Rank multifrontal direct solver*, submitted to Geophysical Journal International, 2016.

Experimental Setting: Matrices (3/3)



3D Structural Mechanics

Double real (d) arithmetic

Symmetric LDL^T factorization

Required accuracy: $\varepsilon = 10^{-9}$

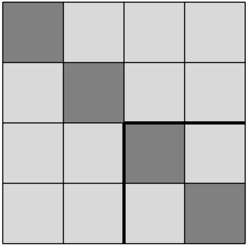
Credits: Code_Aster (EDF)

matrix	n	nnz	flops	storage
perf008d	1.9M	81M	101.0 TF	52.6 GB
perf008ar	3.9M	159M	377.5 TF	129.8 GB
perf009ar	5.4M	209M	23.4 TF	40.2 GB
perf008cr	7.9M	321M	1.6 PF	341.1 GB

Full-Rank statistics

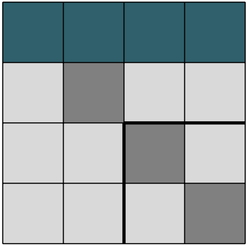
Sequential performance analysis of the BLR factorization

Standard BLR factorization: FSCU



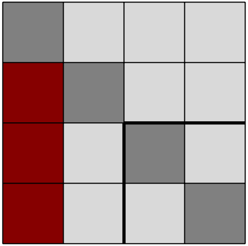
- FSCU

Standard BLR factorization: FSCU



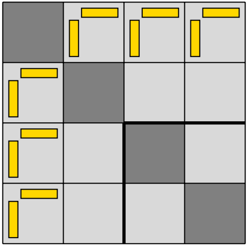
- FSCU (Factor,

Standard BLR factorization: FSCU



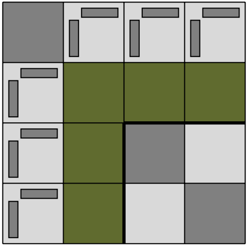
- FSCU (Factor, Solve,

Standard BLR factorization: FSCU



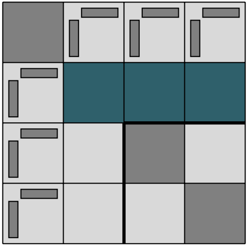
- FSCU (Factor, Solve, Compress,

Standard BLR factorization: FSCU



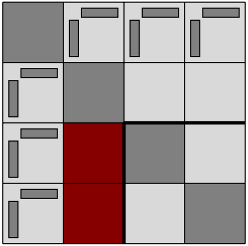
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



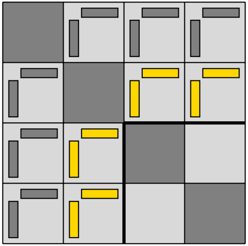
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



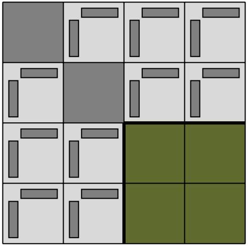
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



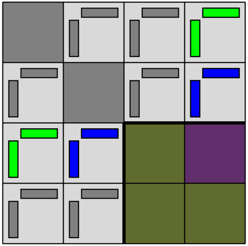
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



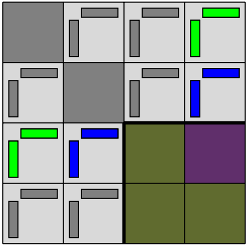
- FSCU (Factor, Solve, Compress, Update)

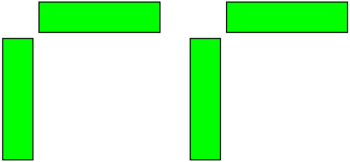
Standard BLR factorization: FSCU



- FSCU (Factor, Solve, Compress, Update)

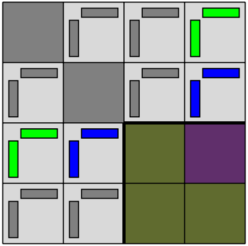
Standard BLR factorization: FSCU

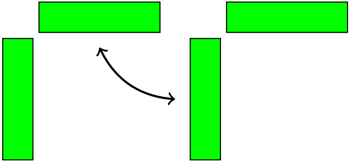




- FSCU (Factor, Solve, Compress, Update)

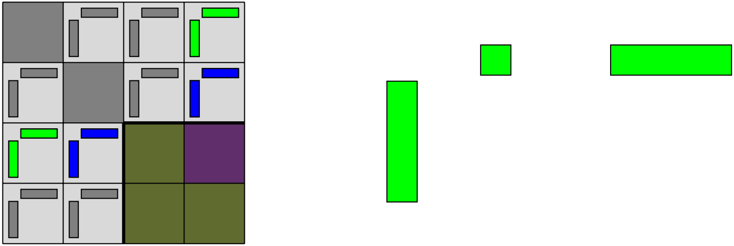
Standard BLR factorization: FSCU





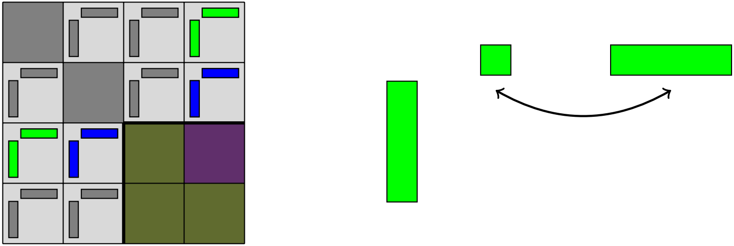
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



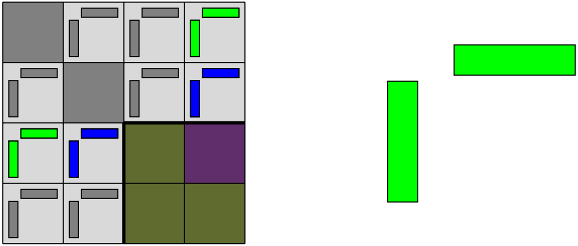
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



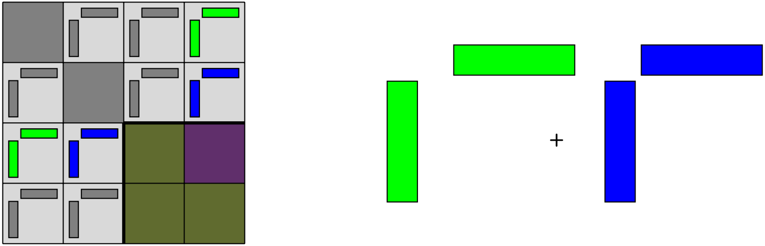
- FSCU (Factor, Solve, Compress, Update)

Standard BLR factorization: FSCU



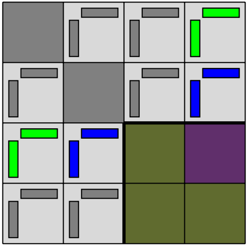
- FSCU (Factor, Solve, Compress, Update)

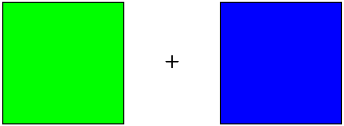
Standard BLR factorization: FSCU



- FSCU (Factor, Solve, Compress, Update)

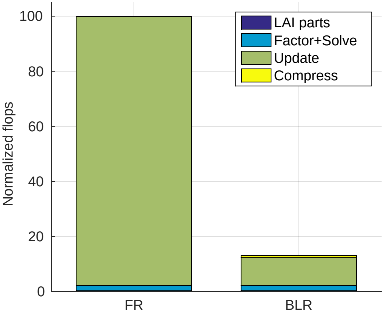
Standard BLR factorization: FSCU



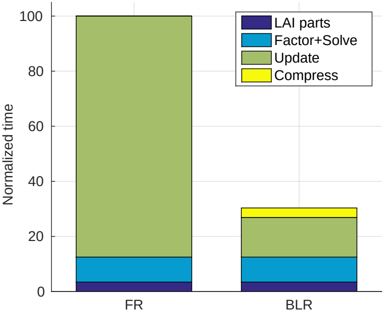


- FSCU (Factor, Solve, Compress, Update)

Sequential result



Normalized Flops



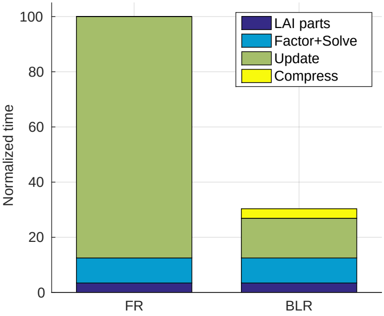
Normalized Time

7.7 gain in flops only translated to a 3.3 gain in time: why?

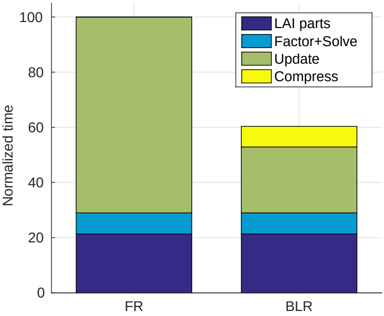
- lower granularity of the Update
- higher relative weight of the FR parts
- inefficient Compress

Multithreading the BLR factorization

Multithreaded result on 24 threads



Normalized Time (Seq.)

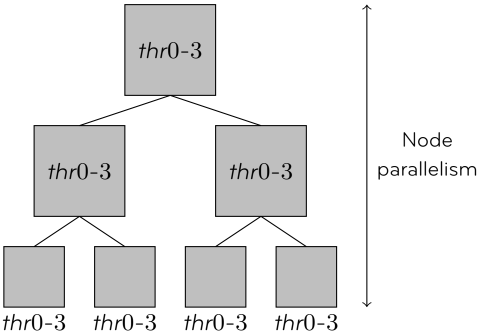


Normalized Time (MT)

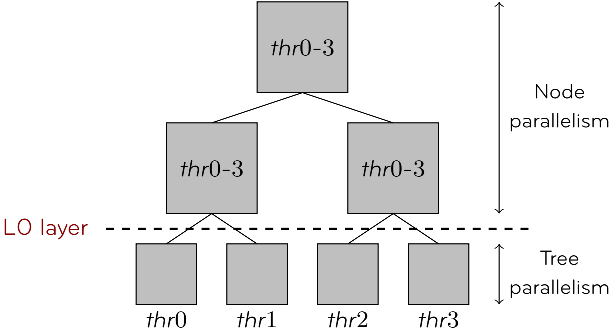
3.3 gain in sequential becomes 1.7 in multithreaded: why?

- LAI parts have become critical
- Update and Compress are memory-bound

Exploiting tree-based multithreading in MF solvers



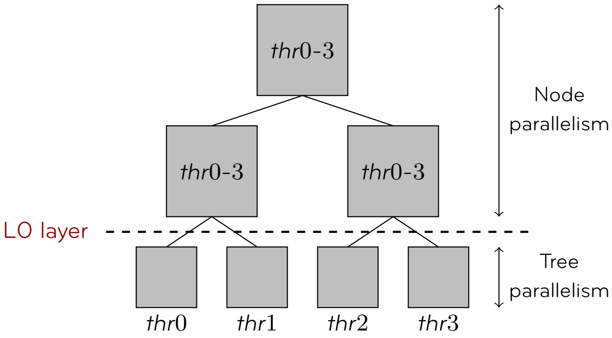
Exploiting tree-based multithreading in MF solvers



- Work based on W. M. Sid-Lakhdar's PhD thesis
  - L0 layer computed with a variant of the Geist-Ng algorithm
  - NUMA-aware implementation
  - use of Idle Core Recycling technique (variant of work-stealing)

▶ L'Excellent and Sid-Lakhdar. *A study of shared-memory parallelism in a multifrontal solver*, Parallel Computing.

Exploiting tree-based multithreading in MF solvers

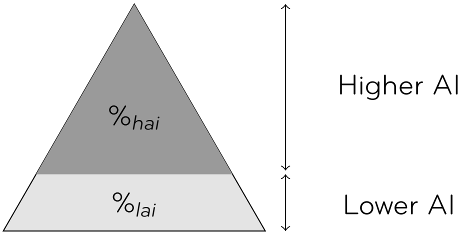


- Work based on W. M. Sid-Lakhdar's PhD thesis
  - L0 layer computed with a variant of the Geist-Ng algorithm
  - NUMA-aware implementation
  - use of Idle Core Recycling technique (variant of work-stealing)

▶ L'Excellent and Sid-Lakhdar. *A study of shared-memory parallelism in a multifrontal solver*, Parallel Computing.

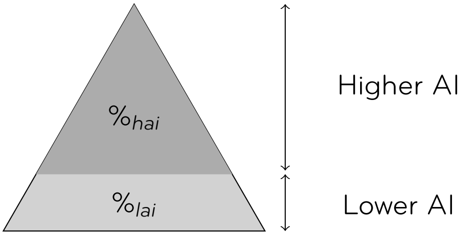
⇒ how big an impact can tree-based multithreading make?

Impact of tree-based multithreading on BLR



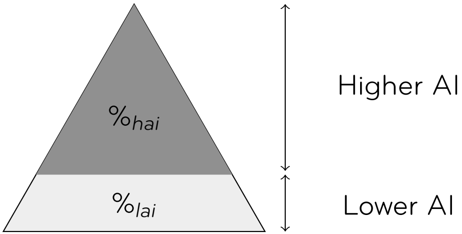
	24 threads		24 threads + tree MT	
	time	$\%_{lai}$	time	$\%_{lai}$
FR BLR	509	21%		

Impact of tree-based multithreading on BLR



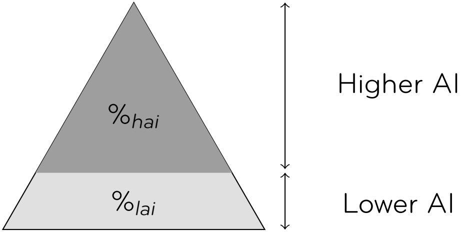
	24 threads		24 threads + tree MT	
	time	$\%_{lai}$	time	$\%_{lai}$
FR	509	21%		
BLR	307	35%		

Impact of tree-based multithreading on BLR



	24 threads		24 threads + tree MT	
	time	%$_{lai}$	time	%$_{lai}$
FR	509	21%	424	13%
BLR	307	35%		

Impact of tree-based multithreading on BLR



	24 threads		24 threads + tree MT	
	time	$\%_{lai}$	time	$\%_{lai}$
FR	509	21%	424	13%
BLR	307	35%	221	24%

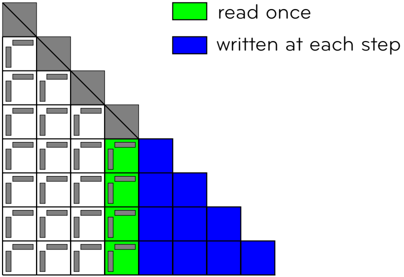
⇒ 1.7 gain becomes 1.9 thanks to tree-based MT

Right Looking Vs. Left-Looking analysis

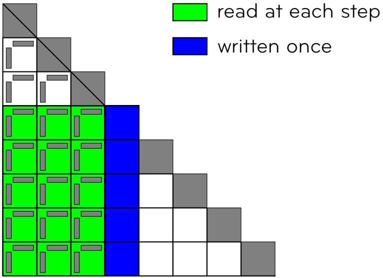
		FR		BLR	
		RL	LL	RL	LL
1 thread	Update	6467		1064	
	Total	7390		2242	
24 threads	Update	338	336	110	67
	Total	424	421	221	175

Right Looking Vs. Left-Looking analysis

		FR		BLR	
		RL	LL	RL	LL
1 thread	Update	6467		1064	
	Total	7390		2242	
24 threads	Update	338	336	110	67
	Total	424	421	221	175



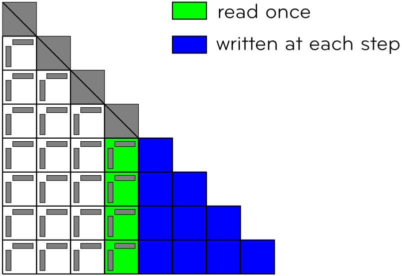
RL factorization



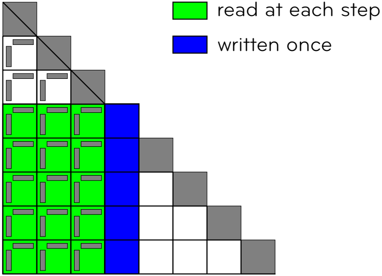
LL factorization

Right Looking Vs. Left-Looking analysis

		FR		BLR	
		RL	LL	RL	LL
1 thread	Update	6467		1064	
	Total	7390		2242	
24 threads	Update	338	336	110	67
	Total	424	421	221	175



RL factorization

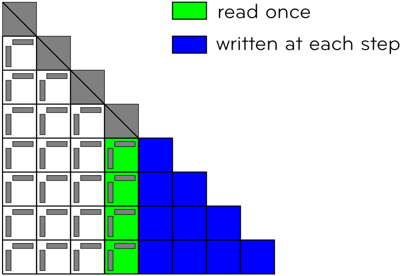


LL factorization

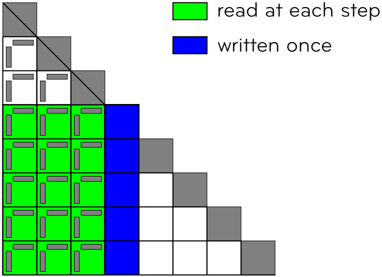
⇒ Lower volume of memory transfers in LL (more critical in MT)

Right Looking Vs. Left-Looking analysis

		FR		BLR	
		RL	LL	RL	LL
1 thread	Update	6467		1064	
	Total	7390		2242	
24 threads	Update	338	336	110	67
	Total	424	421	221	175



RL factorization



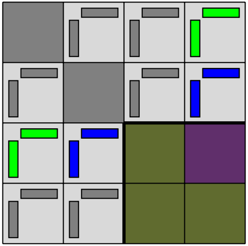
LL factorization

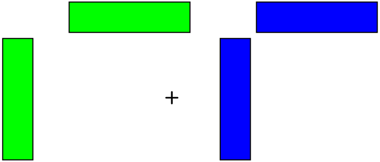
⇒ Lower volume of memory transfers in LL (more critical in MT)

Update is now less memory-bound: 1.9 gain becomes 2.4 in LL

Improving the BLR factorization with algorithmic variants

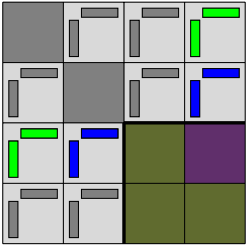
LUAR variant: accumulation and recompression

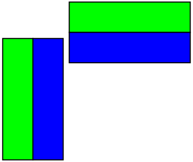




- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR

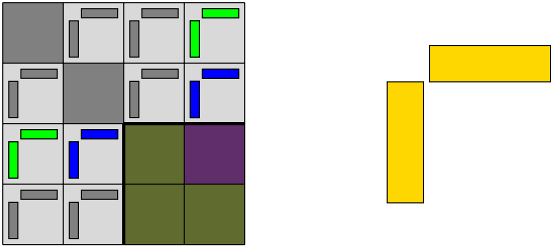
LUAR variant: accumulation and recompression





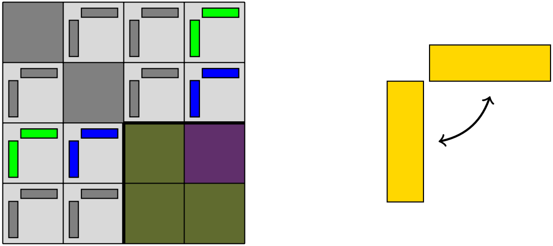
- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
  - Better granularity in Update operations

LUAR variant: accumulation and recompression



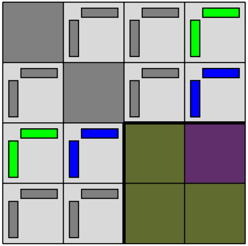
- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
  - Better granularity in Update operations
  - Potential recompression ⇒ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
    - Anton, Ashcraft, and Weisbecker. *A Block Low-Rank multithreaded factorization for dense BEM operators*, presented at SIAM PP'16.

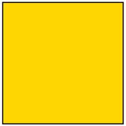
LUAR variant: accumulation and recompression



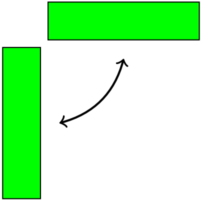
- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
  - Better granularity in Update operations
  - Potential recompression ⇒ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
    - Anton, Ashcraft, and Weisbecker. *A Block Low-Rank multithreaded factorization for dense BEM operators*, presented at SIAM PP'16.

LUAR variant: accumulation and recompression

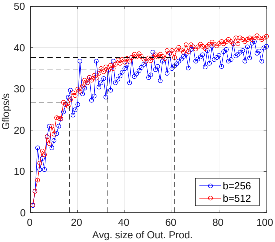




- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
  - Better granularity in Update operations
  - Potential recompression $\Rightarrow$ complexity reduction: $O(n^{\frac{5}{3}}) \to O(n^{\frac{11}{6}})$
    - Anton, Ashcraft, and Weisbecker. *A Block Low-Rank multithreaded factorization for dense BEM operators*, presented at SIAM PP'16.



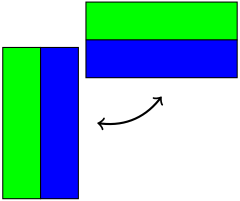
Double complex (z) performance benchmark of Outer Product



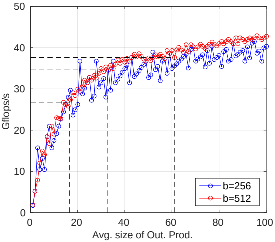
		LL	LUA	LUAR*
average size of Outer Product		16.5	61.0	32.8
flops ($\times 10^{12}$)	Outer Product	3.76	3.76	1.59
	Total	10.19	10.19	8.15
time (s)	Outer Product	21	14	6
	Total	175	167	160

* All metrics include the Recompression overhead

Performance of Outer Product with LUA(R) (24 threads)



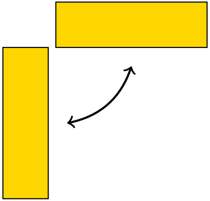
Double complex (z) performance benchmark of Outer Product



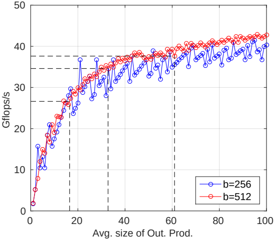
		LL	LUA	LUAR*
average size of Outer Product		16.5	61.0	32.8
flops ($\times 10^{12}$)	Outer Product	3.76	3.76	1.59
	Total	10.19	10.19	8.15
time (s)	Outer Product	21	14	6
	Total	175	167	160

* All metrics include the Recompression overhead

Performance of Outer Product with LUA(R) (24 threads)



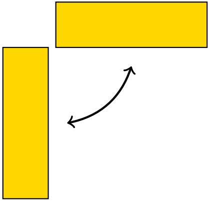
Double complex (z) performance benchmark of Outer Product



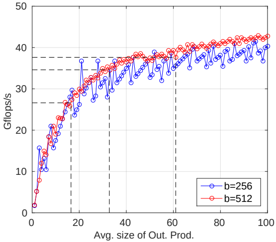
		LL	LUA	LUAR*
average size of Outer Product		16.5	61.0	32.8
flops ($\times 10^{12}$)	Outer Product	3.76	3.76	1.59
	Total	10.19	10.19	8.15
time (s)	Outer Product	21	14	6
	Total	175	167	160

* All metrics include the Recompression overhead

Performance of Outer Product with LUA(R) (24 threads)



Double complex (z) performance benchmark of Outer Product

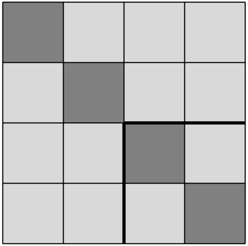


		LL	LUA	LUAR*
average size of Outer Product		16.5	61.0	32.8
flops ($\times 10^{12}$)	Outer Product	3.76	3.76	1.59
	Total	10.19	10.19	8.15
time (s)	Outer Product	21	14	6
	Total	175	167	160

* All metrics include the Recompression overhead

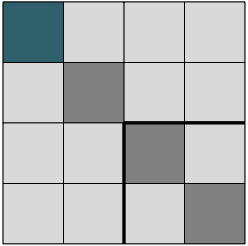
$\Rightarrow$ Higher granularity and lower flops in Update: 2.4 gain becomes 2.6

FCSU variant: compress before solve



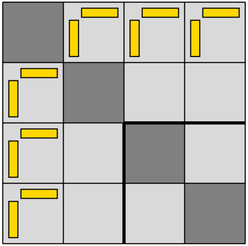
- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
  - Better granularity in Update operations
  - Potential recompression ⇒ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
    - Anton, Ashcraft, and Weisbecker. *A Block Low-Rank multithreaded factorization for dense BEM operators*, presented at SIAM PP'16.
- FCSU(+LUAR)

FCSU variant: compress before solve



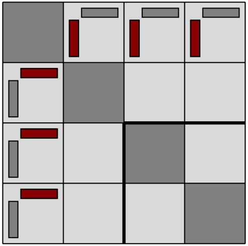
- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
  - Better granularity in Update operations
  - Potential recompression ⇒ complexity reduction: $O(n^{\frac{5}{3}}) \to O(n^{\frac{11}{6}})$
    - Anton, Ashcraft, and Weisbecker. *A Block Low-Rank multithreaded factorization for dense BEM operators*, presented at SIAM PP'16.
- FCSU(+LUAR)
  - Restricted pivoting, e.g. to diagonal blocks

FCSU variant: compress before solve



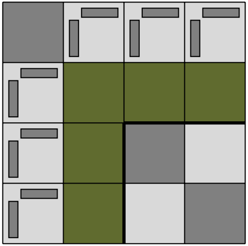
- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
  - Better granularity in Update operations
  - Potential recompression ⇒ complexity reduction: $O(n^{\frac{5}{3}}) \to O(n^{\frac{11}{6}})$
    - Anton, Ashcraft, and Weisbecker. *A Block Low-Rank multithreaded factorization for dense BEM operators*, presented at SIAM PP'16.
- FCSU(+LUAR)
  - Restricted pivoting, e.g. to diagonal blocks

FCSU variant: compress before solve



- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
  - Better granularity in Update operations
  - Potential recompression ⇒ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
    - Anton, Ashcraft, and Weisbecker. *A Block Low-Rank multithreaded factorization for dense BEM operators*, presented at SIAM PP'16.
- FCSU(+LUAR)
  - Restricted pivoting, e.g. to diagonal blocks
  - Low-rank Solve ⇒ complexity reduction: $O(n^{\frac{11}{6}}) \rightarrow O(n^{\frac{4}{3}})$
  - Better BLAS-3/BLAS-2 ratio in Solve operations

FCSU variant: compress before solve



- FSCU (Factor, Solve, Compress, Update)
- FSCU+LUAR
  - Better granularity in Update operations
  - Potential recompression ⇒ complexity reduction: $O(n^{\frac{5}{3}}) \rightarrow O(n^{\frac{11}{6}})$
    - Anton, Ashcraft, and Weisbecker. *A Block Low-Rank multithreaded factorization for dense BEM operators*, presented at SIAM PP'16.
- FCSU(+LUAR)
  - Restricted pivoting, e.g. to diagonal blocks
  - Low-rank Solve ⇒ complexity reduction: $O(n^{\frac{11}{6}}) \rightarrow O(n^{\frac{4}{3}})$
  - Better BLAS-3/BLAS-2 ratio in Solve operations

Performance and accuracy of FCSU vs FSCU

	full pivoting		restricted pivoting		
	FR	FSCU +LUAR	FR	FSCU +LUAR	FCSU +LUAR
flops ($\times 10^{12}$)	77.97	8.15	77.97	8.15	3.95
time (s)	424	160	404	143	111
scaled residual	4.5e-16	1.5e-09	5.0e-16	1.9e-09	2.7e-09

- In many cases...
  - restricted pivoting is enough ⇒ better BLAS-3/BLAS-2 ratio
  - compressing before the Solve has little impact ⇒ flop reduction

⇒ 2.6 gain becomes 3.7

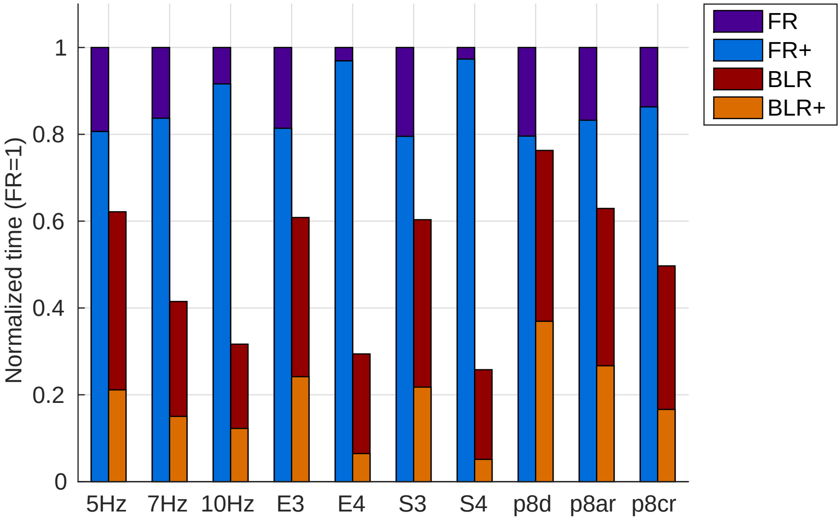
Performance and accuracy of FCSU vs FSCU

	full pivoting		restricted pivoting		
	FR	FSCU +LUAR	FR	FSCU +LUAR	FCSU +LUAR
flops ($\times 10^{12}$)	77.97	8.15	77.97	8.15	3.95
time (s)	424	160	404	143	111
scaled residual	4.5e-16	1.5e-09	5.0e-16	1.9e-09	2.7e-09

- In many cases...
  - restricted pivoting is enough ⇒ better BLAS-3/BLAS-2 ratio
  - compressing before the Solve has little impact ⇒ flop reduction

  ⇒ 2.6 gain becomes 3.7
- When pivoting cannot be restricted...
  - Solve step remains in BLAS-2
  - but Compress before Solve is possible by extending pivoting strategy to low-rank blocks

Results on complete set of problems on 24 threads



Impact of machine properties on BLR

	specs		time (s) for BLR factorization		
	peak (GF/s)	bw (GB/s)	RL	LL	LUA
grunch (28 threads)	37	57	248	228	196
brunch (24 threads)	46	102	221	175	167

S3 matrix

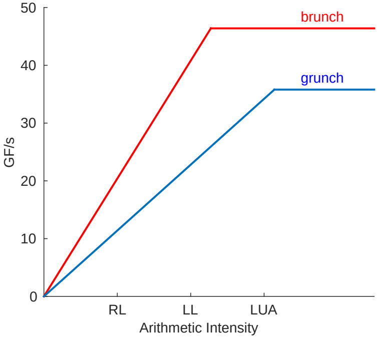
Impact of machine properties on BLR

	specs peak (GF/s)	specs bw (GB/s)	time (s) for BLR factorization RL	LL	LUA
grunch (28 threads)	37	57	248	228	196
brunch (24 threads)	46	102	221	175	167

S3 matrix

Arithmetic Intensity in BLR:

- LL > RL (lower volume of memory transfers)
- LUA > LL (higher granularities $\Rightarrow$ more efficient cache use)



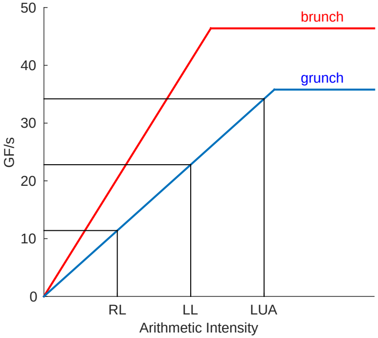
Impact of machine properties on BLR

	specs		time (s) for BLR factorization		
	peak (GF/s)	bw (GB/s)	RL	LL	LUA
grunch (28 threads)	37	57	248	228	196
brunch (24 threads)	46	102	221	175	167

S3 matrix

Arithmetic Intensity in BLR:

- LL > RL (lower volume of memory transfers)
- LUA > LL (higher granularities ⇒ more efficient cache use)



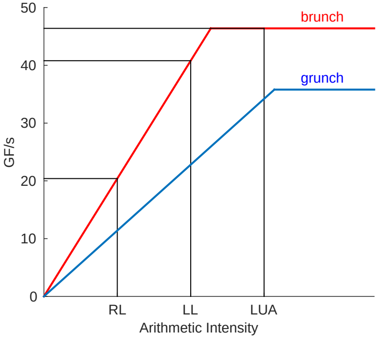
Impact of machine properties on BLR

	specs peak (GF/s)	specs bw (GB/s)	time (s) for BLR factorization RL	LL	LUA
grunch (28 threads)	37	57	248	228	196
brunch (24 threads)	46	102	221	175	167

S3 matrix

Arithmetic Intensity in BLR:

- LL > RL (lower volume of memory transfers)
- LUA > LL (higher granularities $\Rightarrow$ more efficient cache use)



Conclusion and perspectives

Multicore performance of MF BLR factorization

Summary

- Flop reduction is not fully translated into performance gain, especially with multithreading
- Revisited implementation choices: tree-based multithreading and left-looking factorization become critical in BLR
- Introduced BLR variants with better properties
- Improved BLR leads to speedups up to 3 w.r.t. standard BLR and up to 4 w.r.t FR on 24 threads

Perspectives

- Efficient strategies to recompress LR updates
- Extension of pivoting strategy to low-rank blocks (FCSU variant)
- Task-based multithreading
- Reduction of the cost of the Compress

References and acknowledgements

References

- Amestoy, Buttari, L'Excellent, and Mary. *On the Complexity of the Block Low-Rank Multifrontal Factorization*, under review, SIAM SISC, 2016.
- Amestoy, Buttari, L'Excellent, and Mary. *Performance and Scalability of the Multithreaded Block Low-Rank Multifrontal Factorization on Multicore Architectures*, in preparation, 2016.
- Amestoy, Brossier, Buttari, L'Excellent, Mary, Métivier, Miniussi, and Operto. *Fast 3D frequency-domain full waveform inversion with a parallel Block Low-Rank multifrontal direct solver: application to OBC data from the North Sea*, Geophysics, 2016.
- Shantsev, Jaysaval, de la Kethulle de Ryhove, Amestoy, Buttari, L'Excellent, and Mary. *Large-scale 3D EM modeling with a Block Low-Rank multifrontal direct solver*, submitted to Geophysical Journal International, 2016.

Acknowledgements

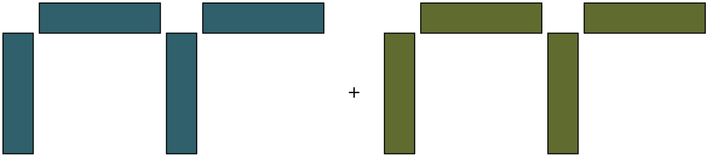
- LIP for providing access to the machines
- EMGS, SEISCOPE and EDF for providing the test matrices
- LSTC members for scientific discussions

Thanks!

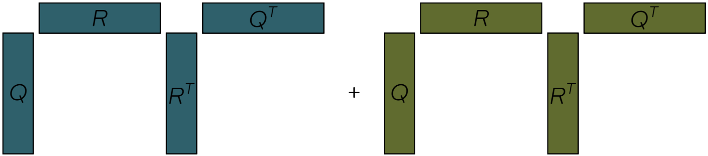
Questions?

Backup Slides

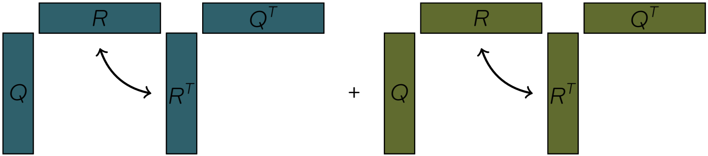
Accumulator recompression



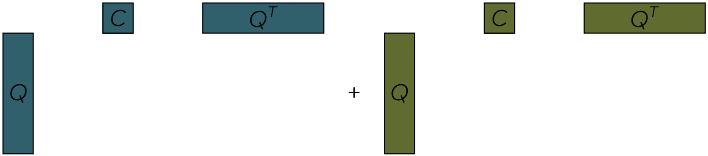
Accumulator recompression



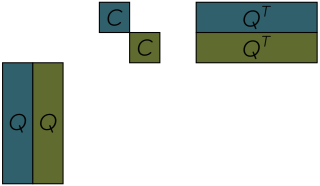
Accumulator recompression



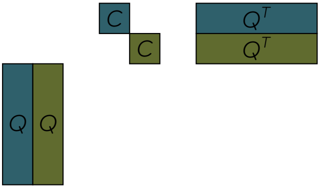
Accumulator recompression



Accumulator recompression



Accumulator recompression



- Weight recompression on $\{C_i\}_i$

  $\Rightarrow$ With absolute threshold ε, each C_i can be compressed separately

- Redundancy recompression on $\{Q_i\}_i$

  $\Rightarrow$ Bigger recompression overhead, when is it worth it?