Intel® Direct Sparse Solver for Clusters, a research project for solving large sparse systems of linear algebraic equation

Alexander Kalinkin
Anders Anton
Anders Roman

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference www.intel.com/software/products.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Atom, Centrino Atom Inside, Centrino Inside, Centrino logo, Cilk, Core Inside, FlashFile, i960, InstantIP, Intel, the Intel logo, Intel386, Intel486, IntelDX2, IntelDX4, IntelSX2, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skoool, Sound Mark, The Journey Inside, Viiv Inside, vPro Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

http://intel.com/software/products

Agenda

- Intro
- Algorithm
- Reordering step
- Factorization step
- Experiments
- Conclusion

Problem statement

$$Ax = b$$

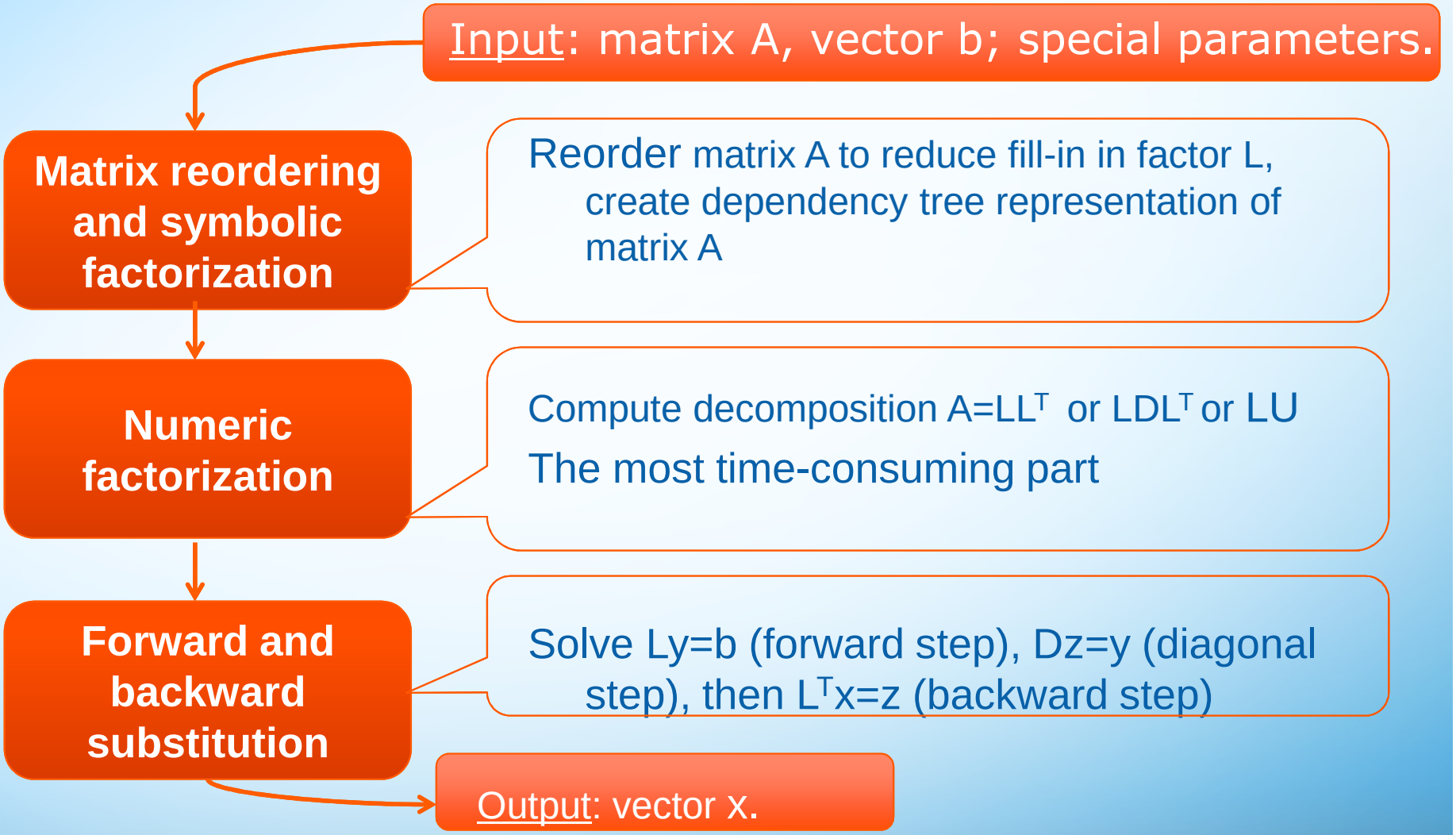
✓ **Cons**

- No extra data available for matrix but some global properties (positive define, hermitian…)
- Huge size

✓ **Pros**

- Clusters with modern Intel® CPUs
- Intel® MKL library with optimized BLAS, LAPACK, PARDISO functionality

Algorithm (Ax=b)

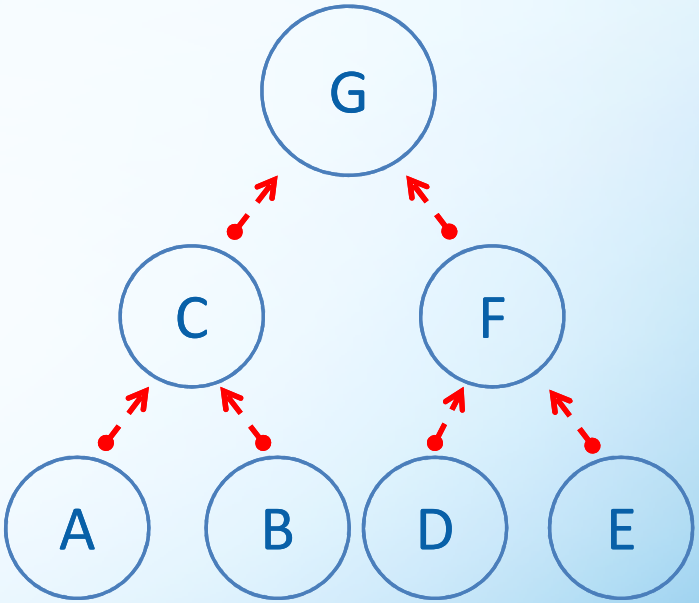


Reordering step

Matrix A after reordering
(example of 4 leafs/process)

	A	B	C	D	E	F	G
E	■						
B		■					
C	■	■	■				
D				■			
E					■		
F				■	■	■	
G	■	■	■	■	■	■	■

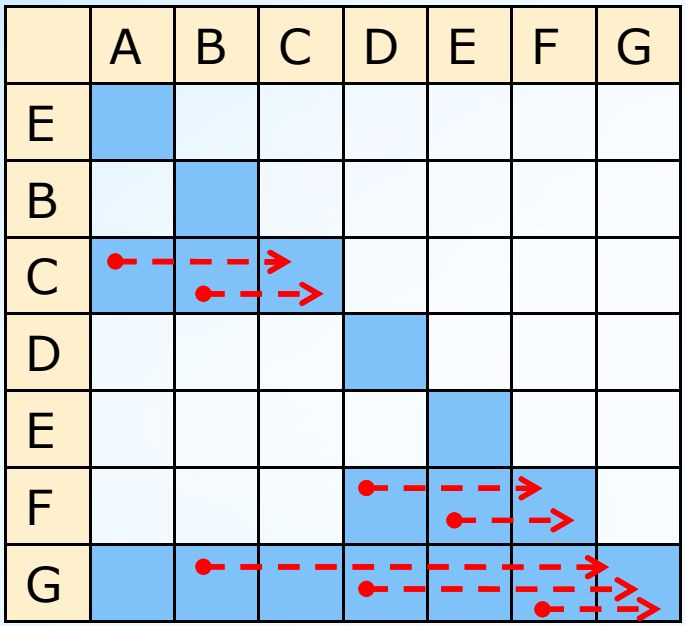
Tree representation of
matrix A after reordering



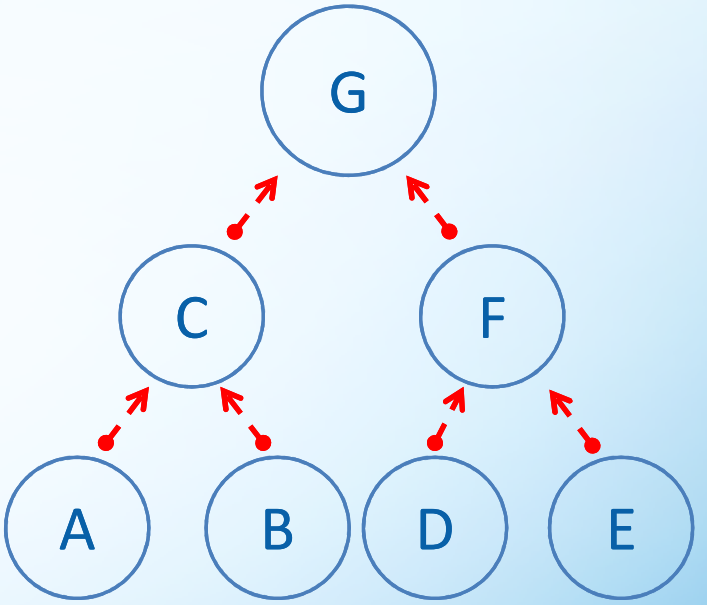
■ - non-zero block

Factorization step

Matrix A after reordering
(example of 4 leafs/process)



Tree representation of
matrix A after reordering

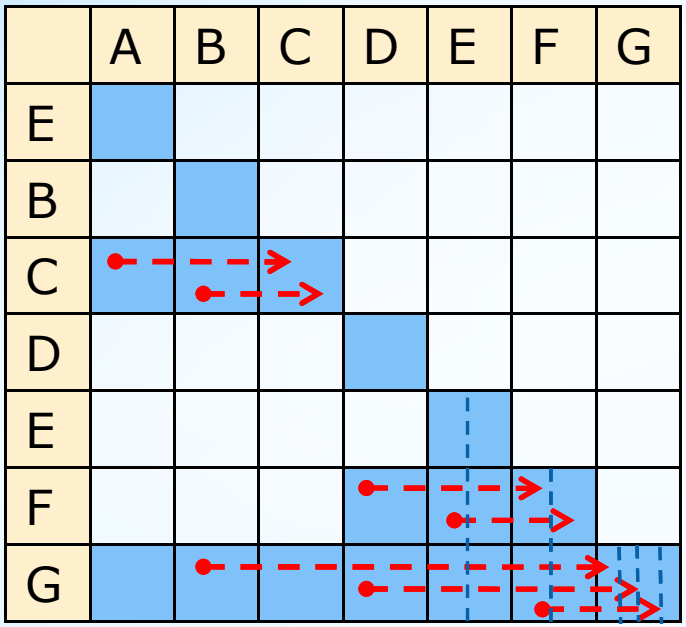


- non-zero block

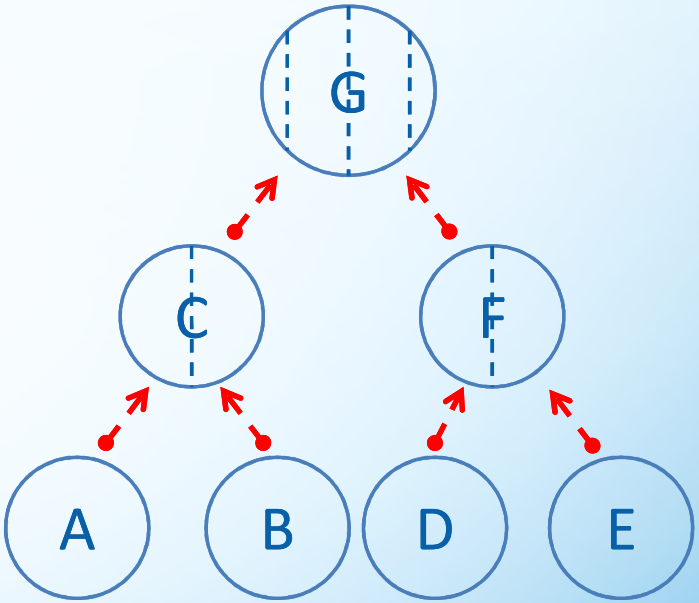
- L-block updates R-block
(or Right depends on Left)

Factorization step

Matrix A after reordering
(example of 4 leafs/process)



Tree representation of
matrix A after reordering

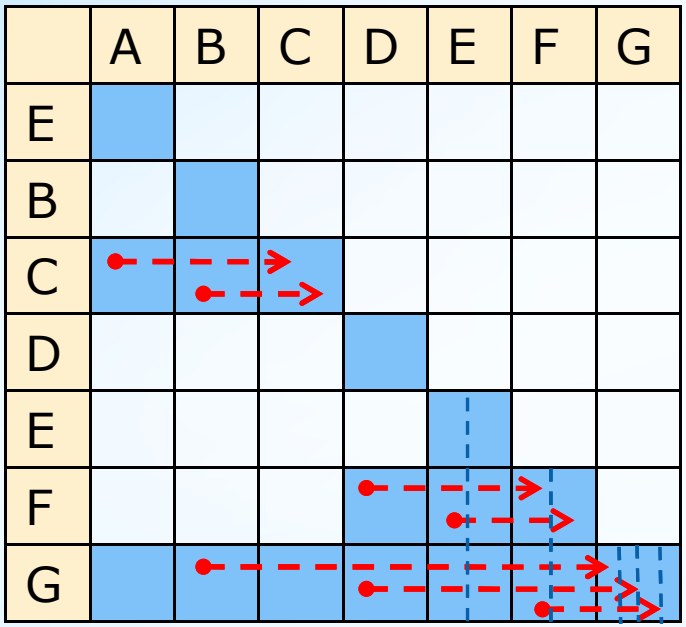


- non-zero block

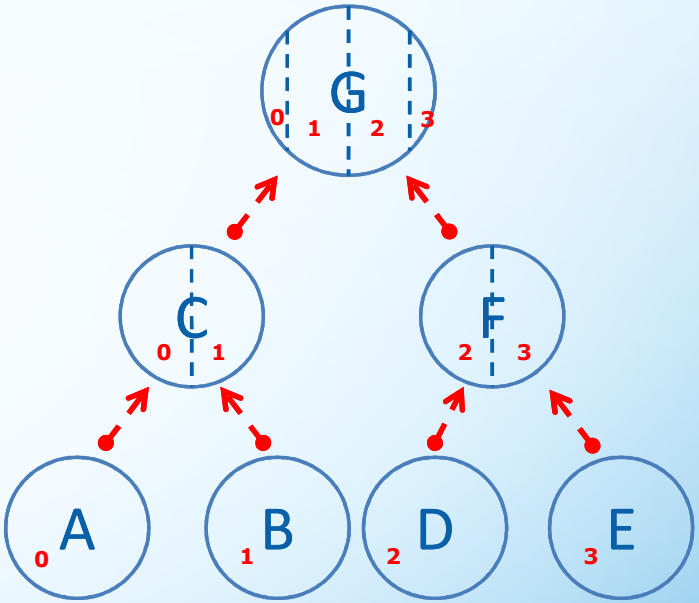
- L-block updates R-block
(or Right depends on Left)

Factorization step

Matrix A after reordering
(example of 4 leafs/process)



Tree representation of
matrix A after reordering

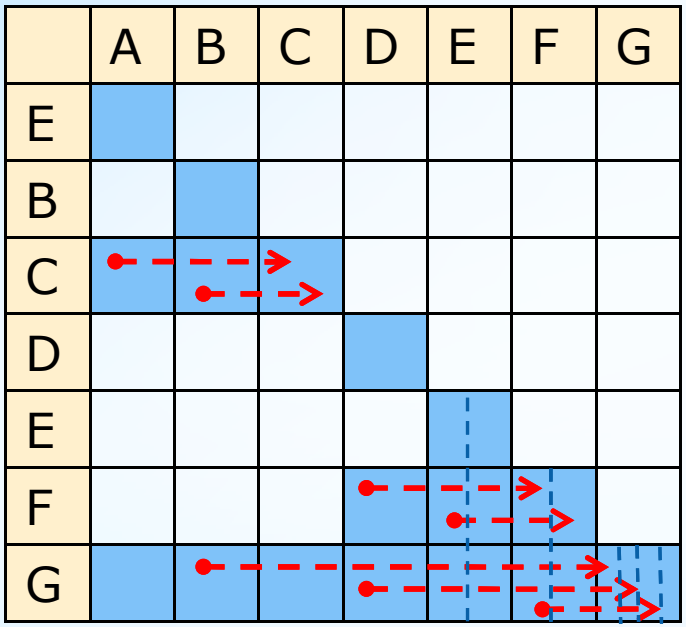


- non-zero block

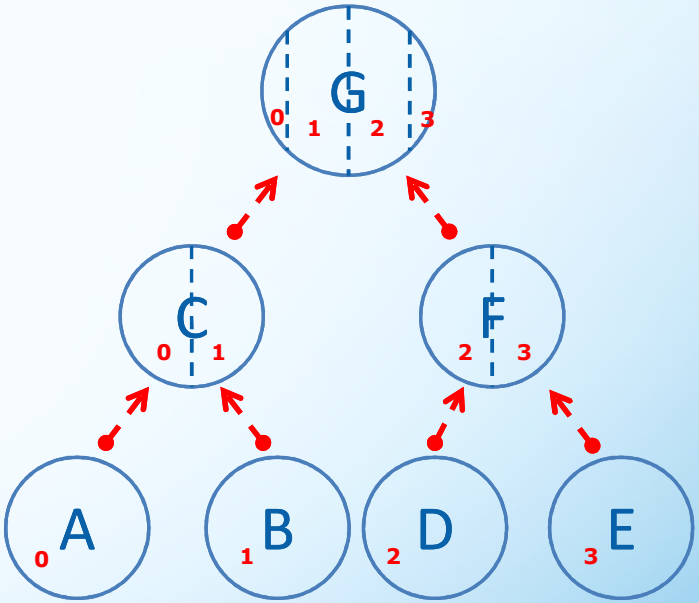
- L-block updates R-block
(or Right depends on Left)

Factorization step

Matrix A after reordering
(example of 4 leafs/process)



Tree representation of
matrix A after reordering



- Both tree and tree-node parallelization used
- All computations within the node are based on functionality from Intel® MKL
- Computation of leafs & updates of a block are independent on each process
- Data distributed between processes uniformly

- non-zero block

- L-block updates R-block
(or Right depends on Left)

Factorization step

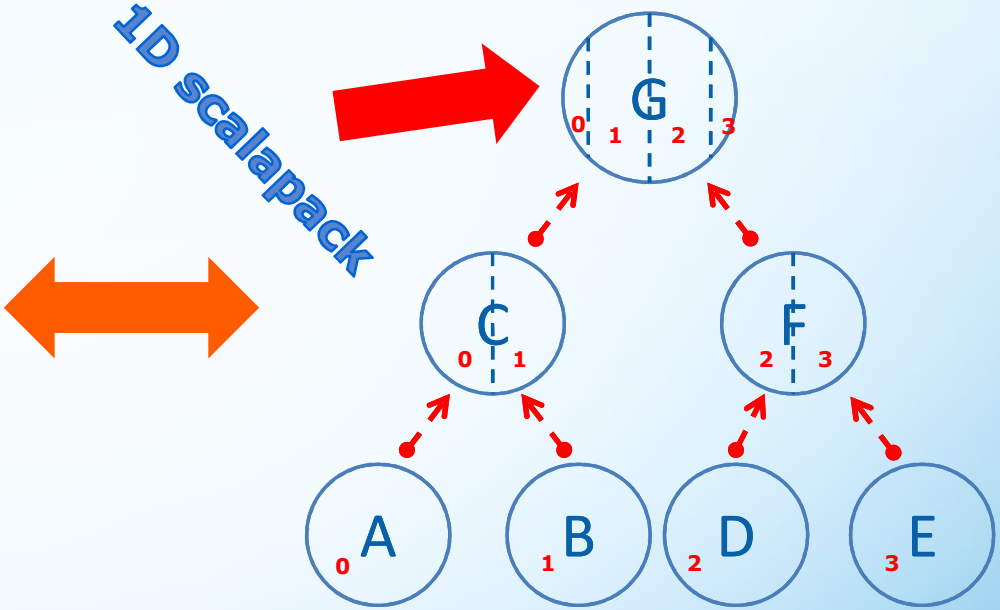
Matrix A after reordering
(example of 4 leafs/process)

	A	B	C	D	E	F	G
E							
B							
C							
D							
E							
F							
G							

- non-zero block

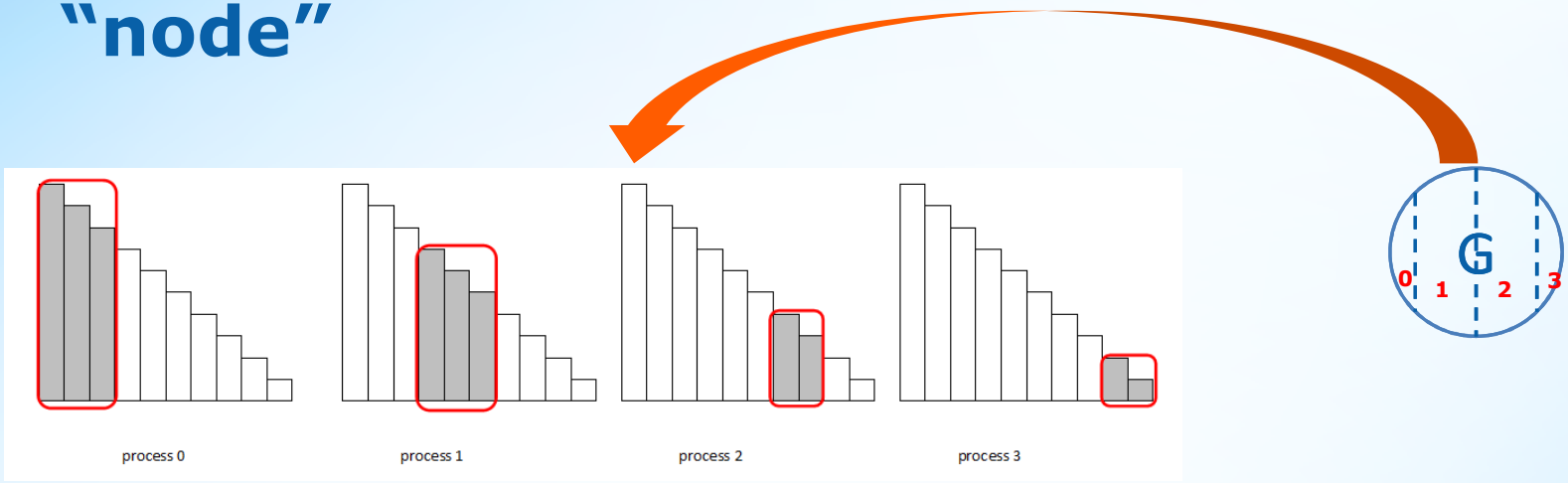
- L-block updates R-block
(or Right depends on Left)

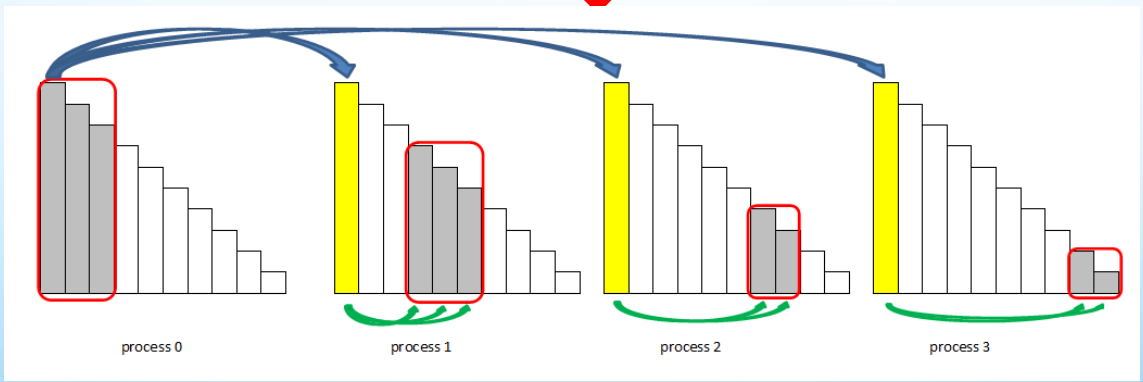
Tree representation of
matrix A after reordering



- Both tree and tree-node parallelization used
- All computations within the node are based on functionality from Intel® MKL
- Computation of leafs & updates of a block are independent on each process
- Data distributed between processes uniformly

Implementation of LU decomposition in "node"





Choosing one thread per process allow us to "mask" data transfer time under computational process

Current status/interface

Supported as 2 additional libraries, Lnx & Win 64 bit only.
Ported by different MPI via user-compiled wrapper.

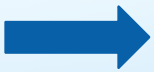
C:

```
{
….

PARDISO (pt, &maxfct, &mnum, &mtype,
     &phase, &n, a, ia, ja, &idum, &nrhs,
                iparm, &msglvl, b, x, &error);
…
}
```

```
{
….
comm =  MPI_Comm_c2f(MPI_COMM_WORLD);
CPARDISO (pt, &maxfct, &mnum, &mtype,
     &phase, &n, a, ia, ja, &idum, &nrhs,
     iparm, &msglvl, b, x, comm, &error);
…
}
```

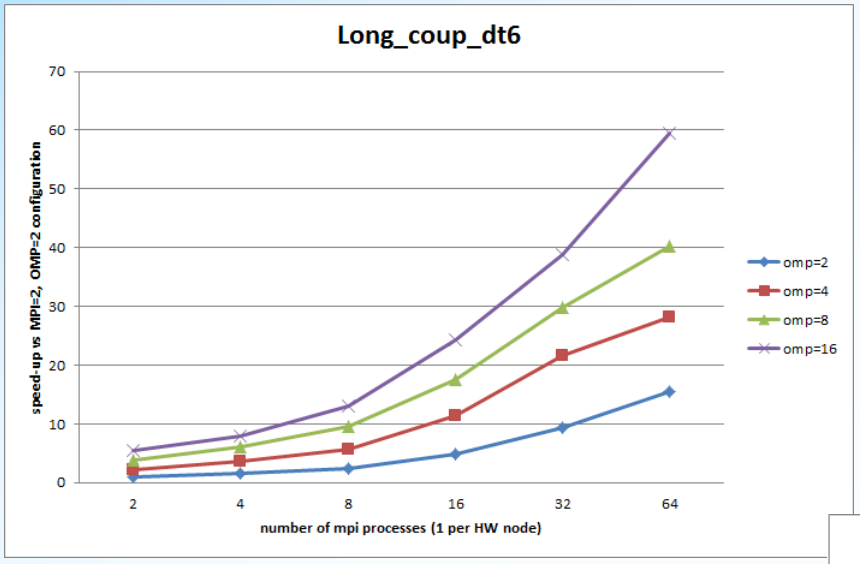
Fortran:

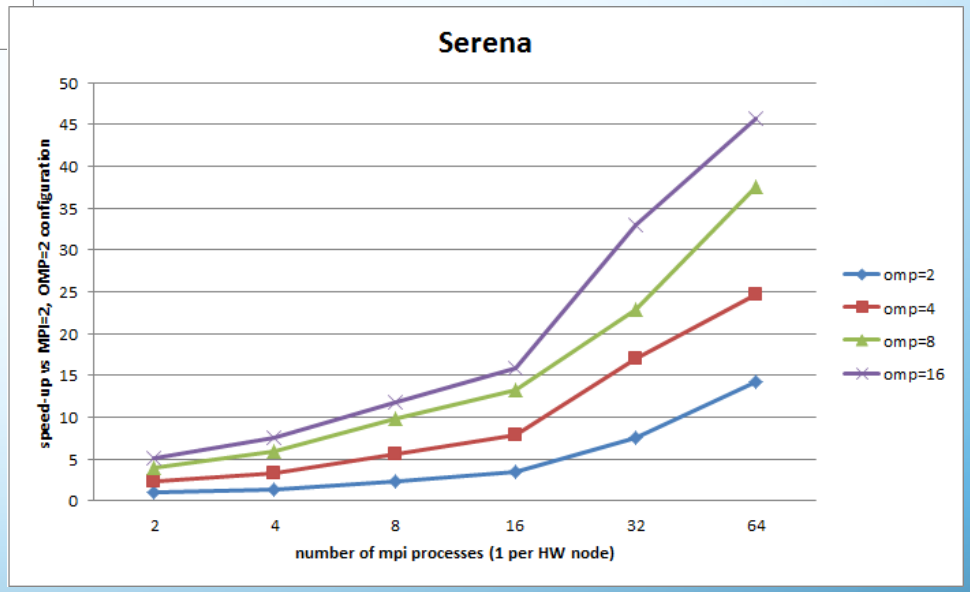
```
….

Call PARDISO(pt, maxfct, mnum, mtype,
     phase, n, a, ia, ja, idum, nrhs,
                iparm, msglvl, b, x, error);
…
```



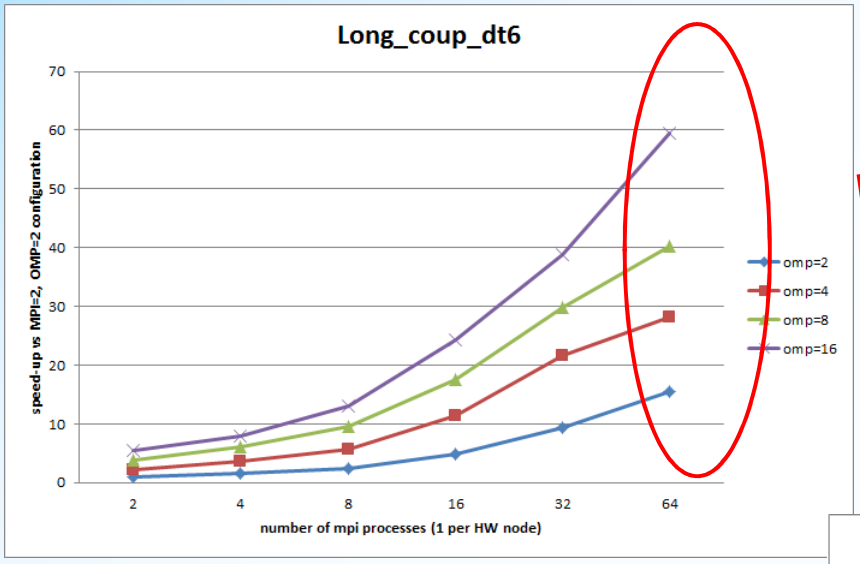
```
…

Call CPARDISO(pt, maxfct, mnum, mtype,
     phase, n, a, ia, ja, idum, nrhs,
     iparm, msglvl, b, x, comm, &error);
…
```

Experiments (scalability of time)

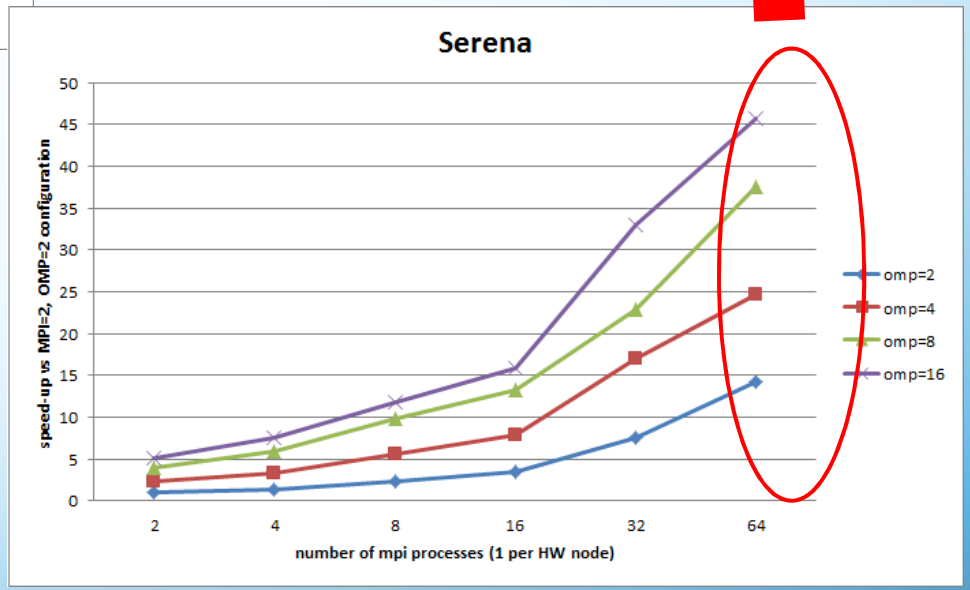




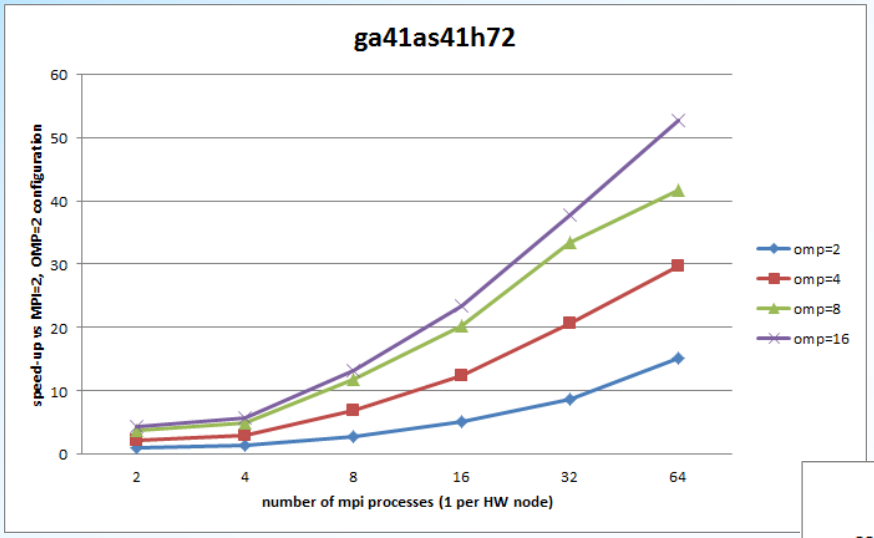
Experiments (scalability of time)

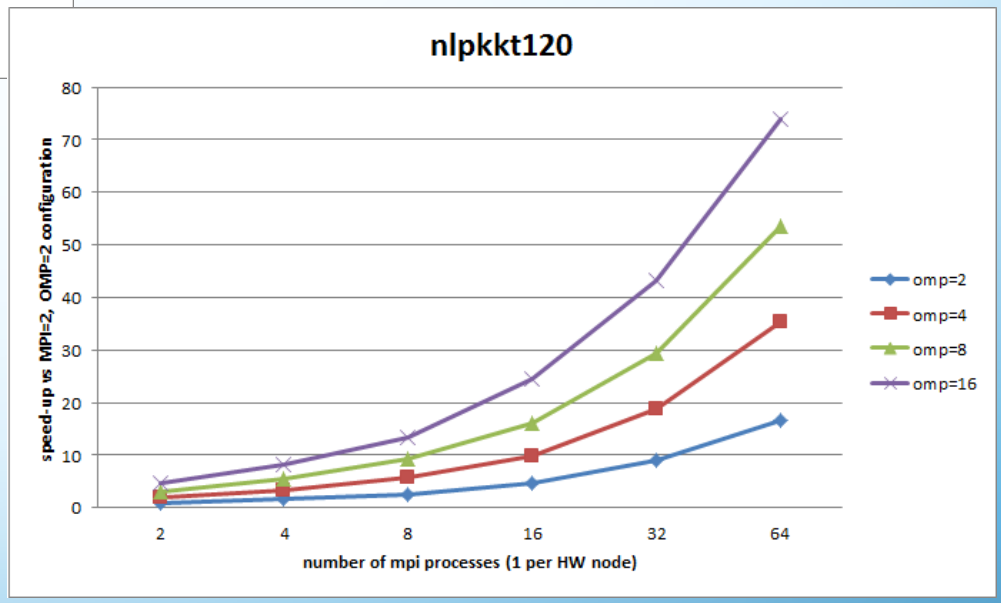


Additional processes reduce computational time!!!

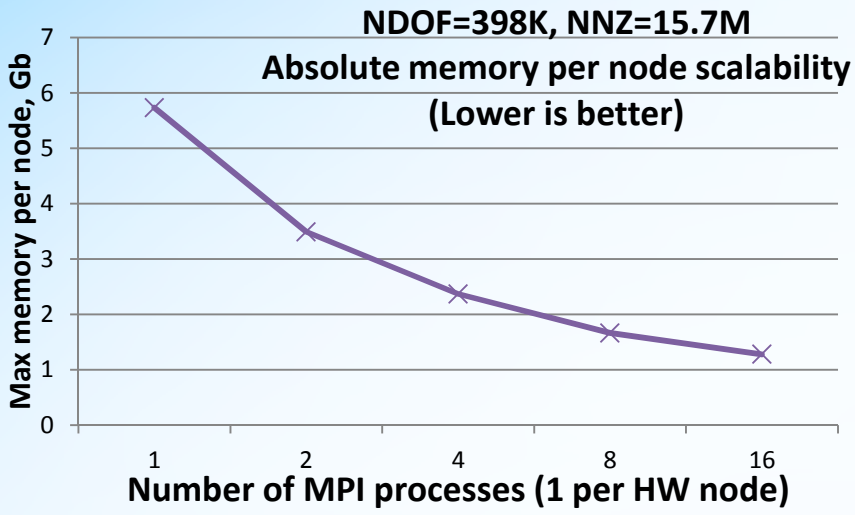


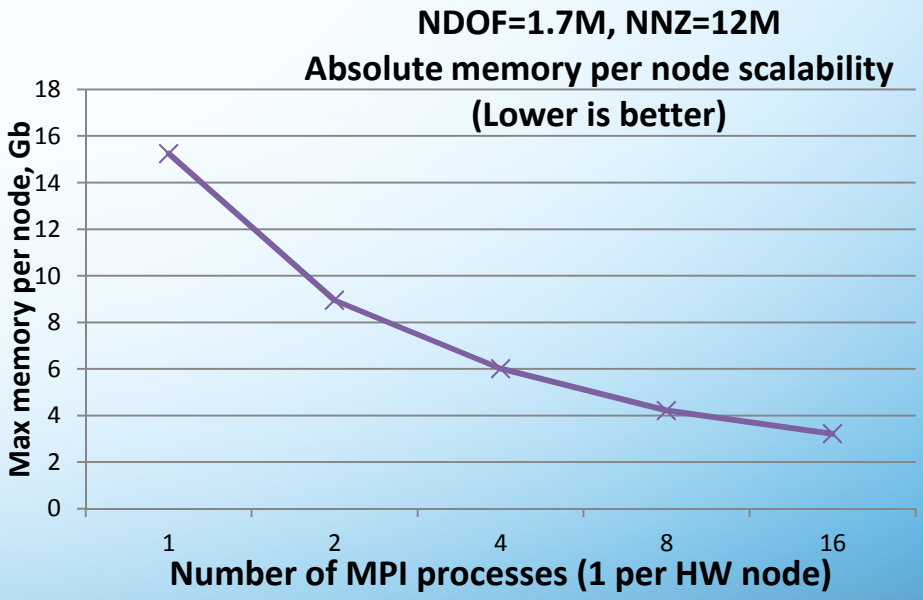
Experiments (scalability of time)



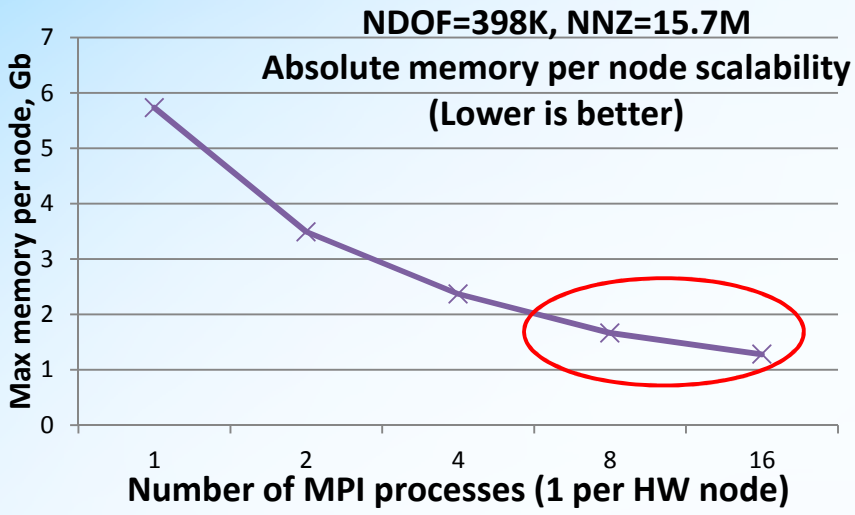


Experiments (scalability of memory)

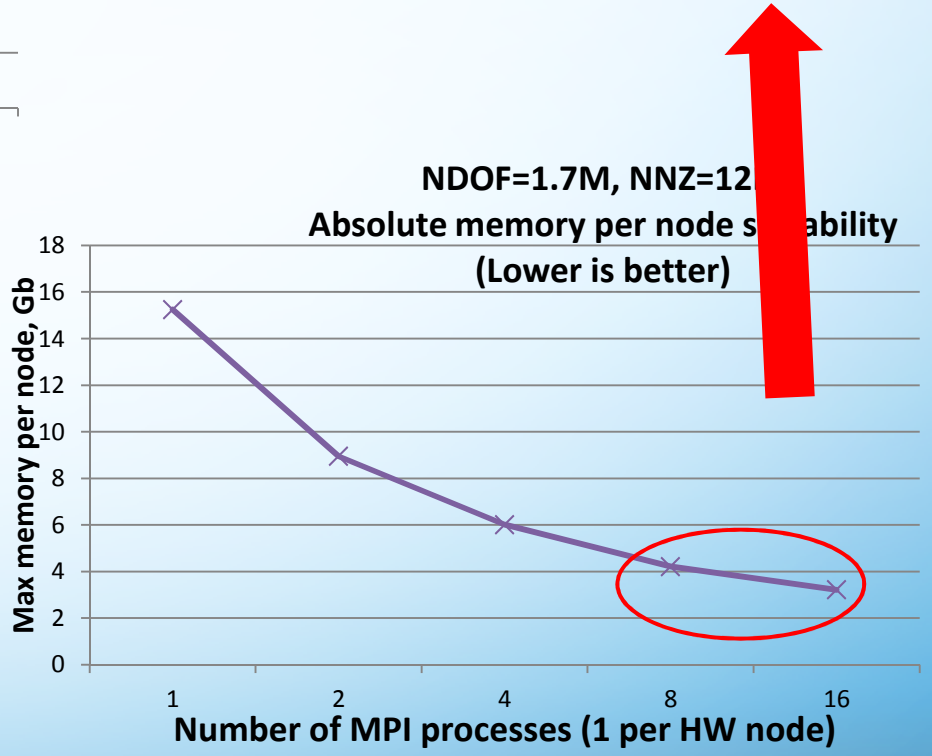




Experiments (scalability of memory)



Additional processes decrease memory size per host!!!

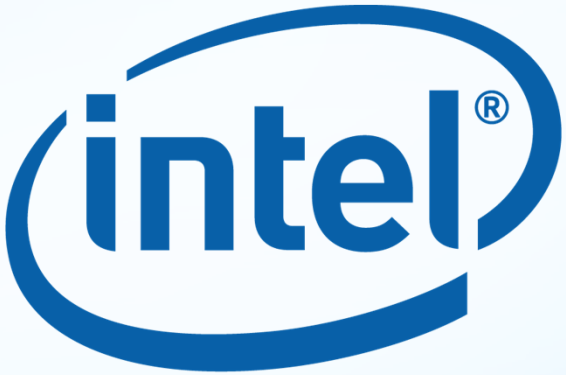


Conclusion

Intel® Direct Sparse Solver for Clusters based on Intel® MKL functionality results in

- Good scaling of computational time
- Good scaling of memory per node

Q & A



Software

Optimization Notice

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804