Golden Nuggets of Discovery
CUG 2011 • Fairbanks, Alaska • May 23-26

Prospects for truly asynchronous communication with pure MPI and hybrid MPI/OpenMP on current supercomputing platforms

Georg Hager[1], Gerald Schubert[2], Thomas Schoenemeyer[3], Gerhard Wellein[1,4]

[1]Erlangen Regional Computing Center (RRZE), Germany
[2]Institute of Physics, University of Greifswald, Germany
[3]Swiss National Supercomputing Centre (CSCS), Manno, Switzerland
[4]Department for Computer Science, Friedrich-Alexander-University Erlangen-Nuremberg, Germany

Cray User Group Meeting, May 23-26, 2011, Fairbanks, AK

CSCS
Swiss National Supercomputing Centre

KONWIHR II

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Agenda

- **MPI nonblocking != asynchronous**

- **Options for really asynchronous communication**
  - MPI does it ok
  - Separate explicit communication thread

- **Example: Sparse matrix-vector multiply (spMVM)**
  - Motivation and properties
  - Node performance model
  - Distributed-memory parallelization
  - Hiding communication: “vector mode” vs. “task mode”

- **Results**
  - XE6 vs. Westmere EP InfiniBand cluster

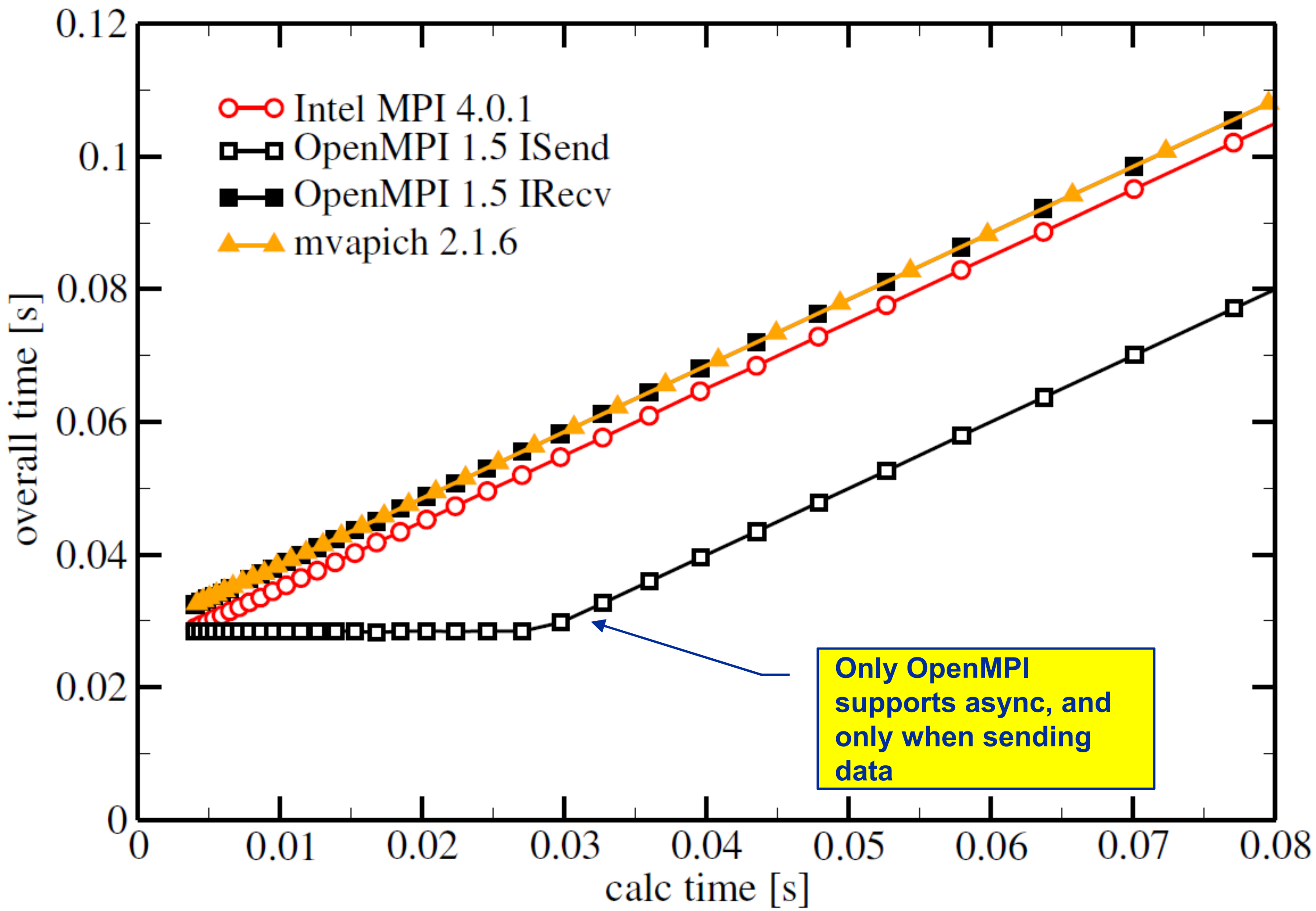
MPI nonblocking point-to-point communication

- Is nonblocking automatically asynchronous? → Simple benchmark:

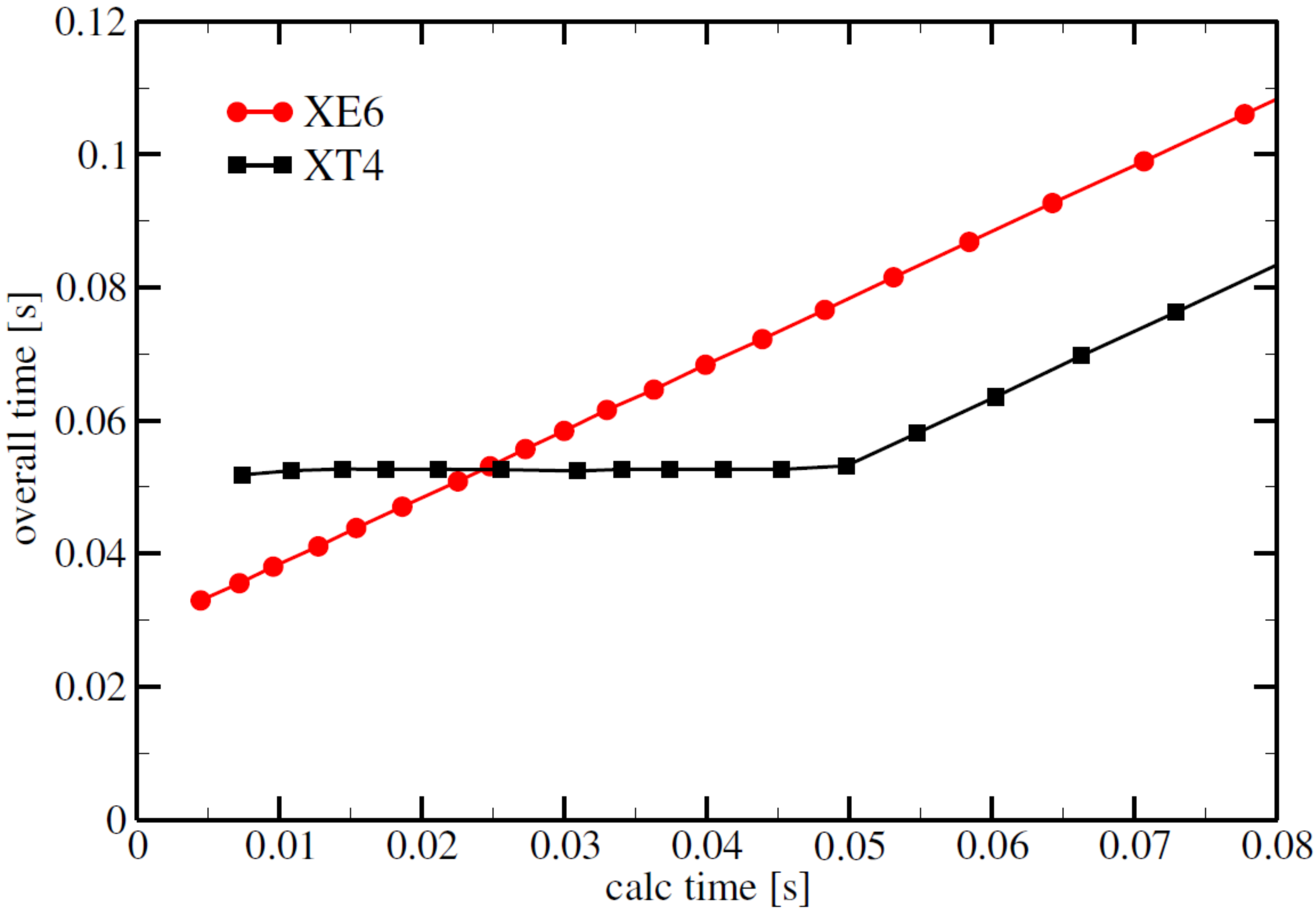
```
if(rank==0) {
  stime = MPI_Wtime();
  MPI_Irecv(rbuf,mcount,MPI_DOUBLE,1,0,
     MPI_COMM_WORLD,&req);
  do_work(calctime);
  MPI_Wait(req, &status);
  etime = MPI_Wtime();
  cout << calctime << " " << etime-stime << endl;
} else {
  MPI_Send(sbuf,mcount,MPI_DOUBLE,0,0,
     MPI_COMM_WORLD);
}
```

- For low calctime, execution time is constant if async works!
- Benchmark: 80 MByte message size, in-register workload (do_work)
- Generally no intranode async supported!

- Internode results for Westmere cluster (QDR-IB)



- **Internode results for Cray XT4 and XE6**



MPI nonblocking – results and consequences

- **Asynchronous nonblocking MPI does not work in general for large messages**

- **Consequences**
  - If we need async, check if it works
  - If it doesn't, perform comm/calc overlap manually

- **Comm/calc overlap: Options with MPI and MPI/OpenMP**
  - Nonblocking MPI
  - Sacrifice one thread for communication
    - Compute performance impact?
    - Where/how to run? Threads vs. processes?
    - Can SMT be of any use?

- **Case study: Sparse matrix-vector multiply (spMVM)**

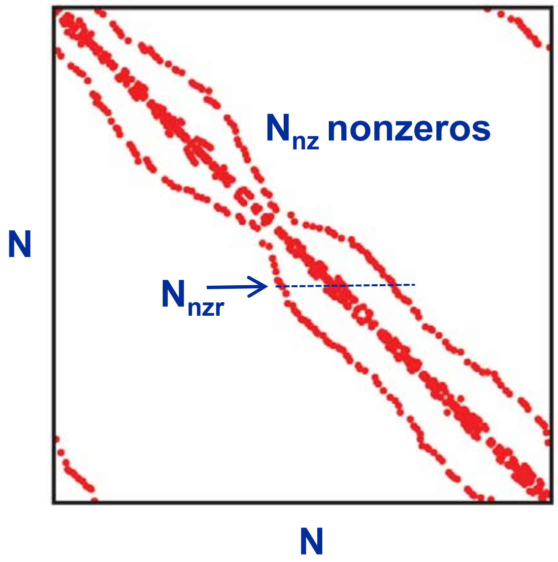
- **Why spMVM?**
  → **Dominant operation in many algorithms/applications**

- **Physics applications:**
  - Ground state phase diagram Holstein-Hubbard model
  - Physics at the Dirac point in Graphene
  - Anderson localization in disordered systems
  - Quantum dynamics on percolative lattices
- **Algorithms:**
  - Lanczos – extremal eigenvalues
  - JADA – degenerate & inner eigenvalues
  - KPM – spectral properties
  - Chebyshev time evolution

- **Fraction of total time spent in SpMVM: 85 – 99.99%**

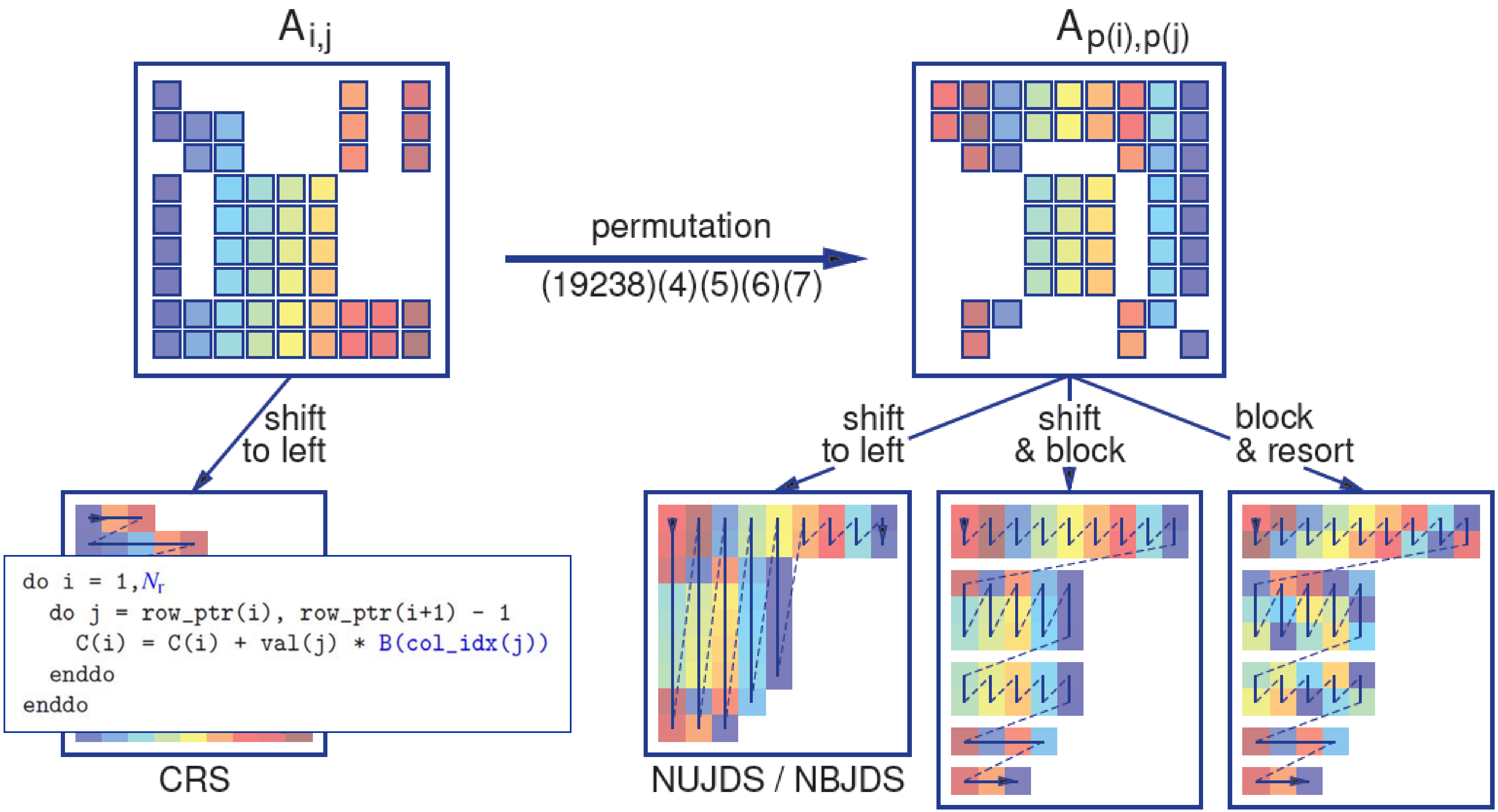
Sparse MVM properties

- **"Sparse" matrix ≅ N_{nz} grows slower than quadratically with N**
  - N_{nzr} = avg. # nonzeros per row
- **A different sparsity pattern ("fingerprint") for each problem**
- **Performance of spMVM c = A·b**
  - Always memory-bound for large N (see later)
  - Usage of memory BW divided between nonzeros and RHS vector
  - Sparsity pattern has strong impact
  - Storage format, too



- **Storage formats**
  - Compressed Row Storage (CRS): Best for modern cache-based µP
  - Jagged Diagonals Storage (JDS): Best for vector(-like) architectures
  - Special formats exploit specific matrix properties

A quick glance on CRS and JDS variants…




```
do i = 1,Nr
  do j = row_ptr(i), row_ptr(i+1) - 1
    C(i) = C(i) + val(j) * B(col_idx(j))
  enddo
enddo
```

G. Schubert, G. Hager and H. Fehske: *Performance limitations for sparse matrix-vector multiplications on current multicore environments.* In: S. Wagner et al., High Performance Computing in Science and Engineering, Garching/Munich 2009. Springer, ISBN 978-3642138713 (2010), 13–26. DOI: 10.1007/978-3-642-13872-0_2, Preprint: arXiv:0910.4836.

SpMVM node performance model

High Performance Computing

- **Concentrate on double precision CRS:**

```
do i = 1,N_r
  do j = row_ptr(i), row_ptr(i+1) - 1
    C(i) = C(i) + val(j) * B(col_idx(j))
  enddo
enddo
```

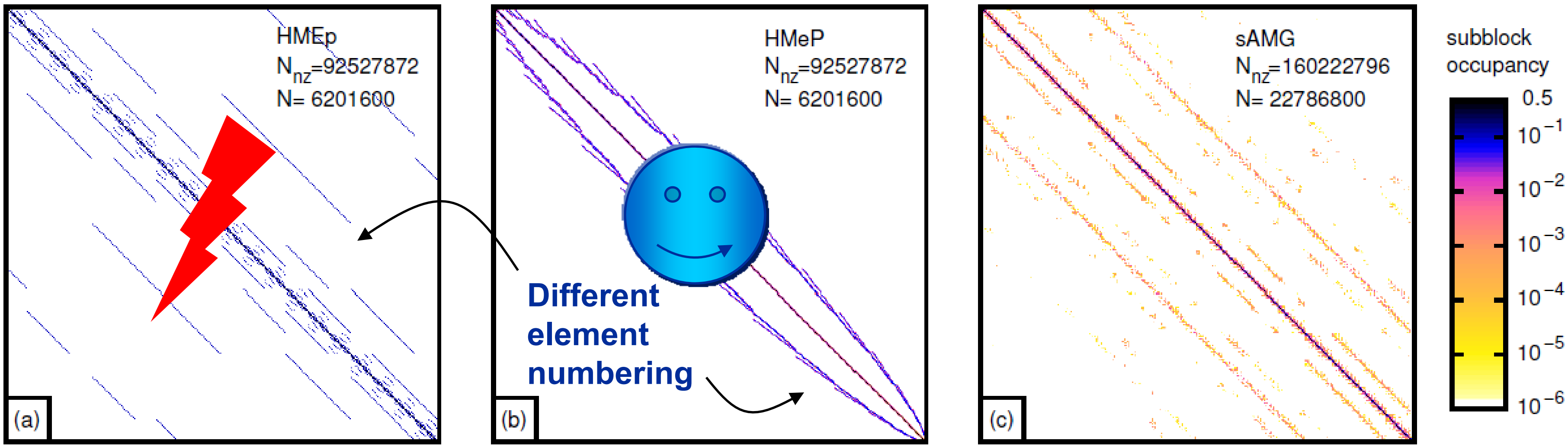
- **DP CRS code balance:**

$$B_{\mathrm{CRS}} = \left(\frac{12 + 24/N_{\mathrm{nzr}} + \kappa}{2}\right) \frac{\mathrm{bytes}}{\mathrm{flop}} = \left(6 + \frac{12}{N_{\mathrm{nzr}}} + \frac{\kappa}{2}\right) \frac{\mathrm{bytes}}{\mathrm{flop}} .$$

  - κ quantifies extra traffic for loading RHS more than once
  - Predicted Performance = streamBW/B_{CRS}
  - Determine κ by measuring performance and actual memory BW

- **Matrices in our test cases: $N_{nzr} \approx 7 \ldots 15$ → RHS and LHS do matter!**
  - HM: Hostein-Hubbard Model, 6-site lattice, 6 electrons, 15 phonons
  - sAMG: Adaptive Multigrid method, irregular discretization of Poisson stencil on car geometry
  - Considered Reverse Cuthill-McKee (RCM) transformation, but no gain

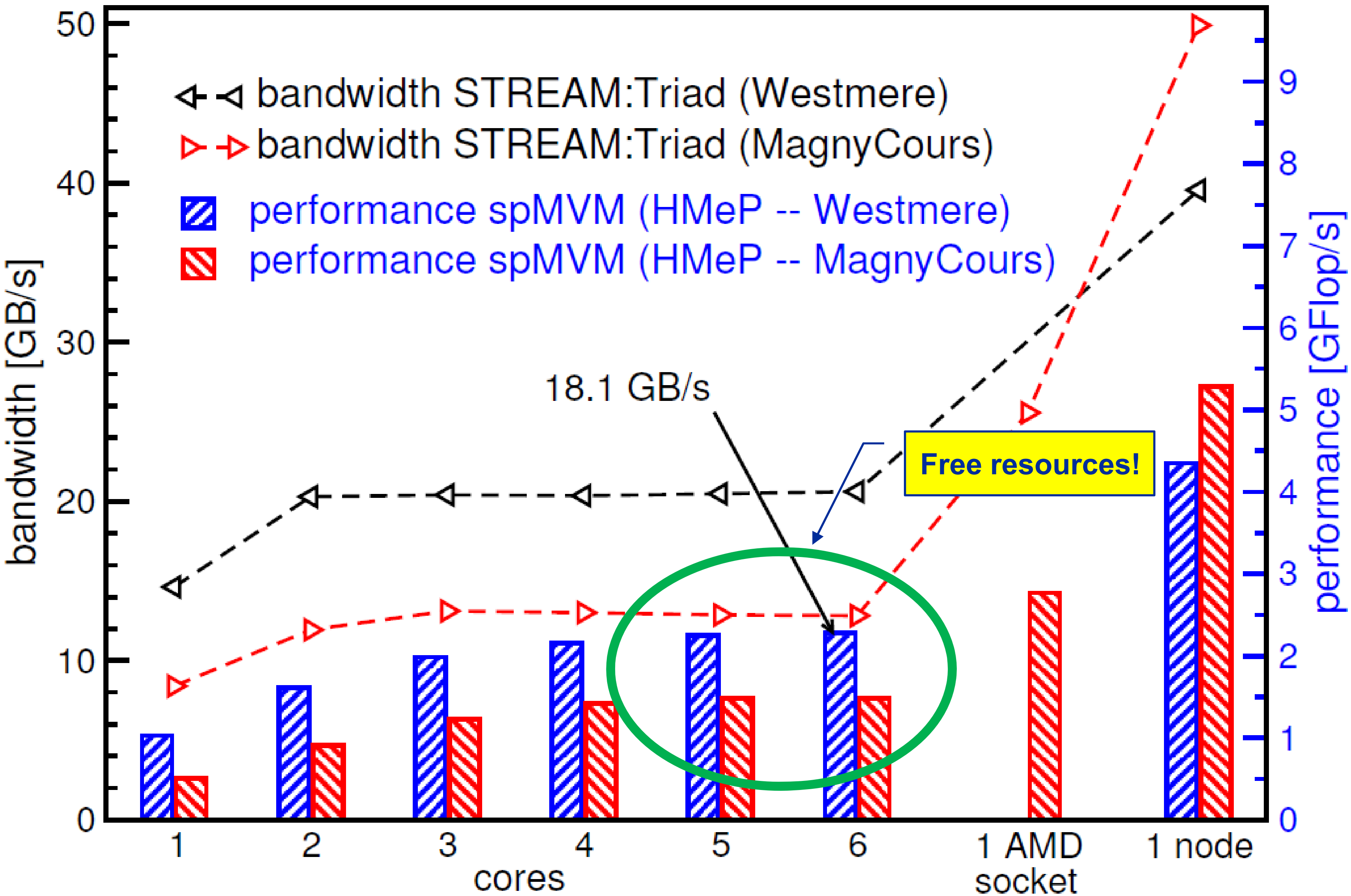
Test matrices: Sparsity patterns



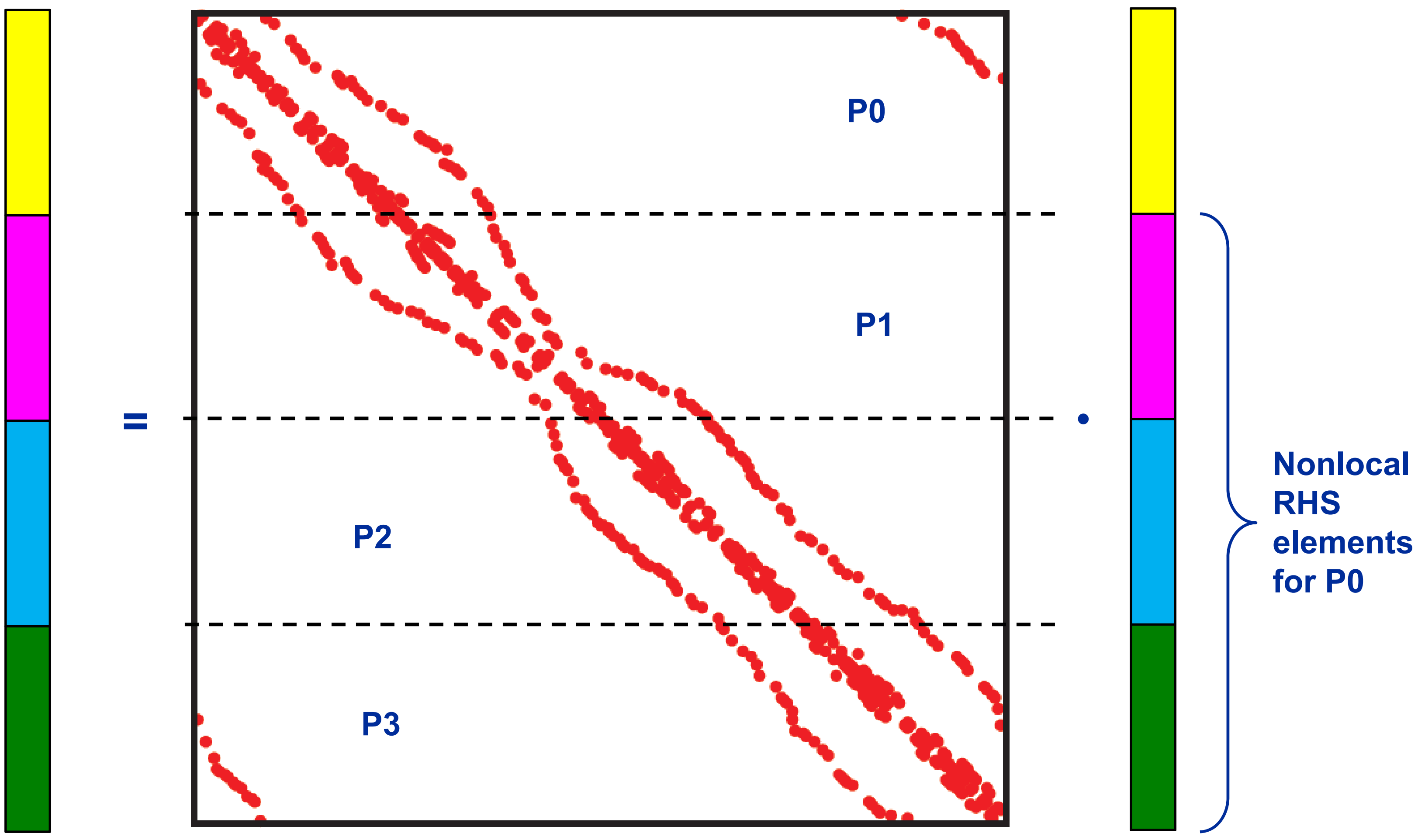
- **HMeP: RHS loaded six times from memory**
  - → about 33% of BW goes into RHS

- **Special formats that exploit features of the sparsity pattern are not considered here**

Node-level performance for HMeP: Westmere EP vs. Cray XE6 (Magny Cours)



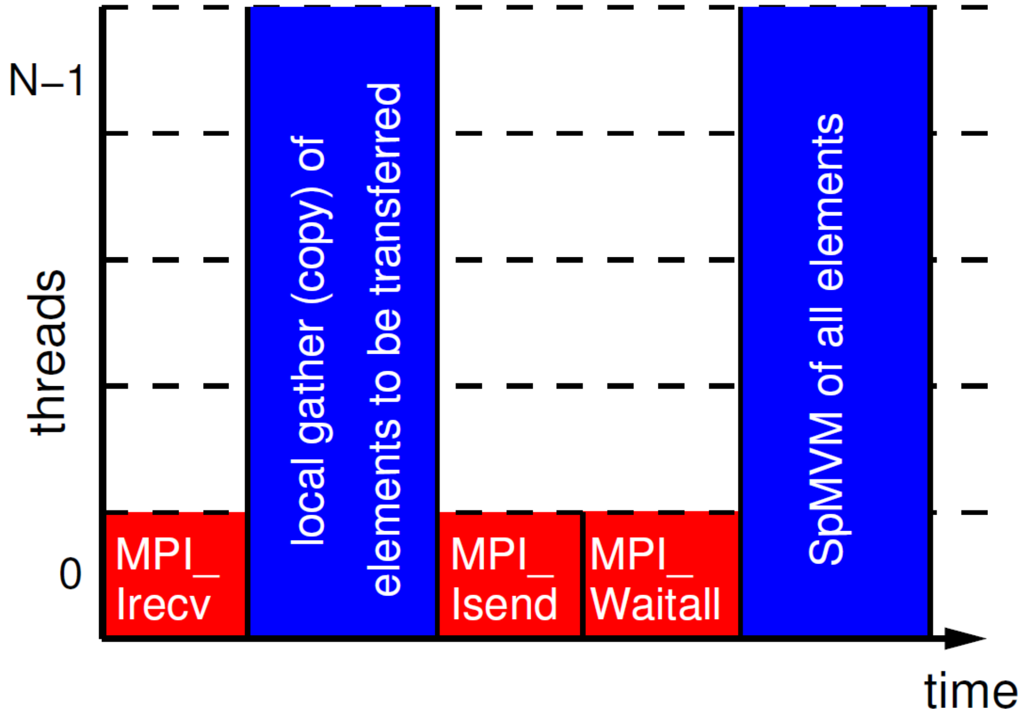
Distributed-memory parallelization of spMVM



Distributed-memory parallelization of spMVM

- **Variant 1: "Vector mode" without overlap**

- **Standard concept for "hybrid MPI+OpenMP"**
- **Multithreaded computation (all threads)**
- **Communication only outside of computation**



- **Benefit of threaded MPI process only due to message aggregation and (probably) better load balancing**

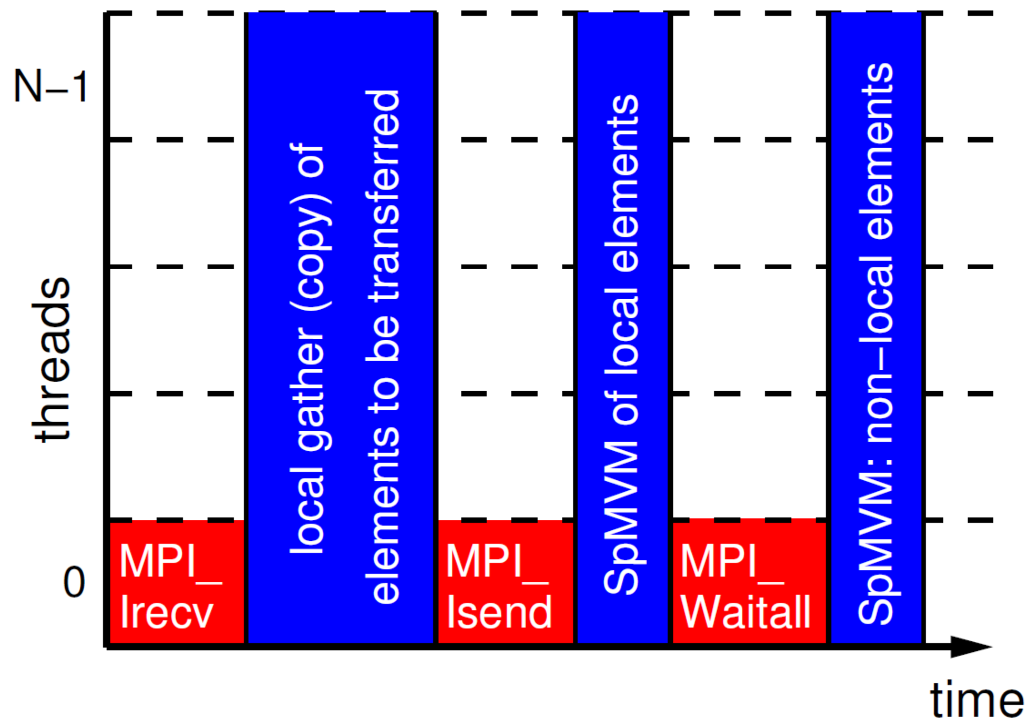
G. Hager, G. Jost, and R. Rabenseifner: *Communication Characteristics and Hybrid MPI/OpenMP Parallel Programming on Clusters of Multi-core SMP Nodes.* In: Proceedings of the Cray Users Group Conference 2009 (CUG 2009), Atlanta, GA, USA, May 4-7, 2009. PDF

Distributed-memory parallelization of spMVM

- **Variant 2: “Vector mode” with naïve overlap (“good faith hybrid”)**

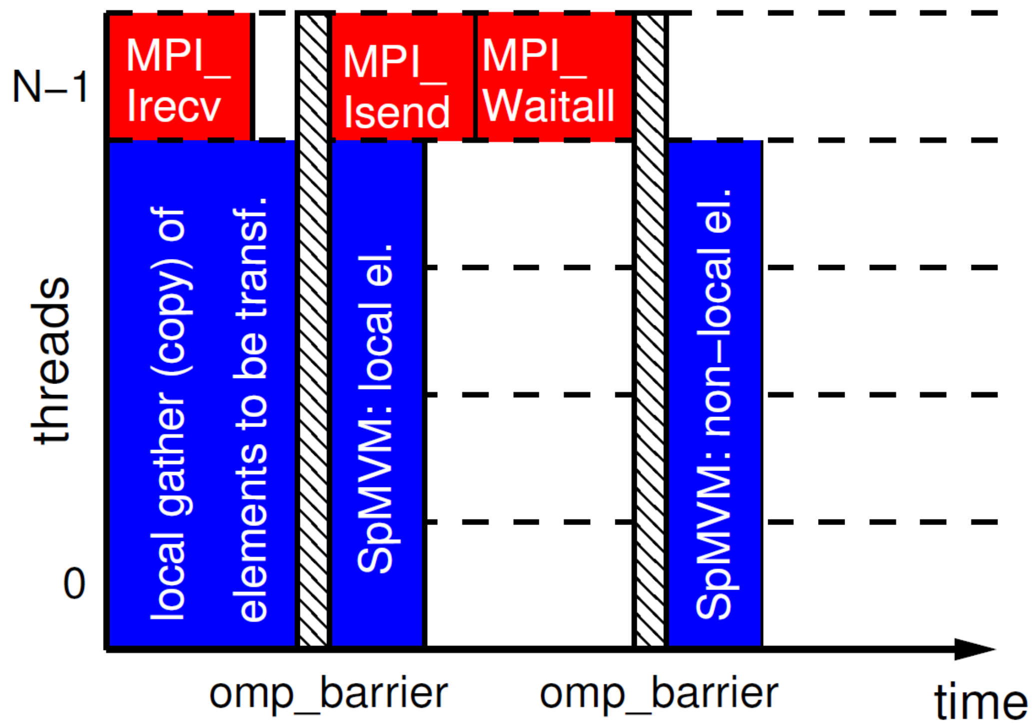
- **Relies on MPI to support async nonblocking PtP**
- **Multithreaded computation (all threads)**

- **Still simple programming**
- **Drawback: Result vector is written twice to memory**
  - modified performance model



- **Variant 3: "Task mode" with dedicated communication thread**

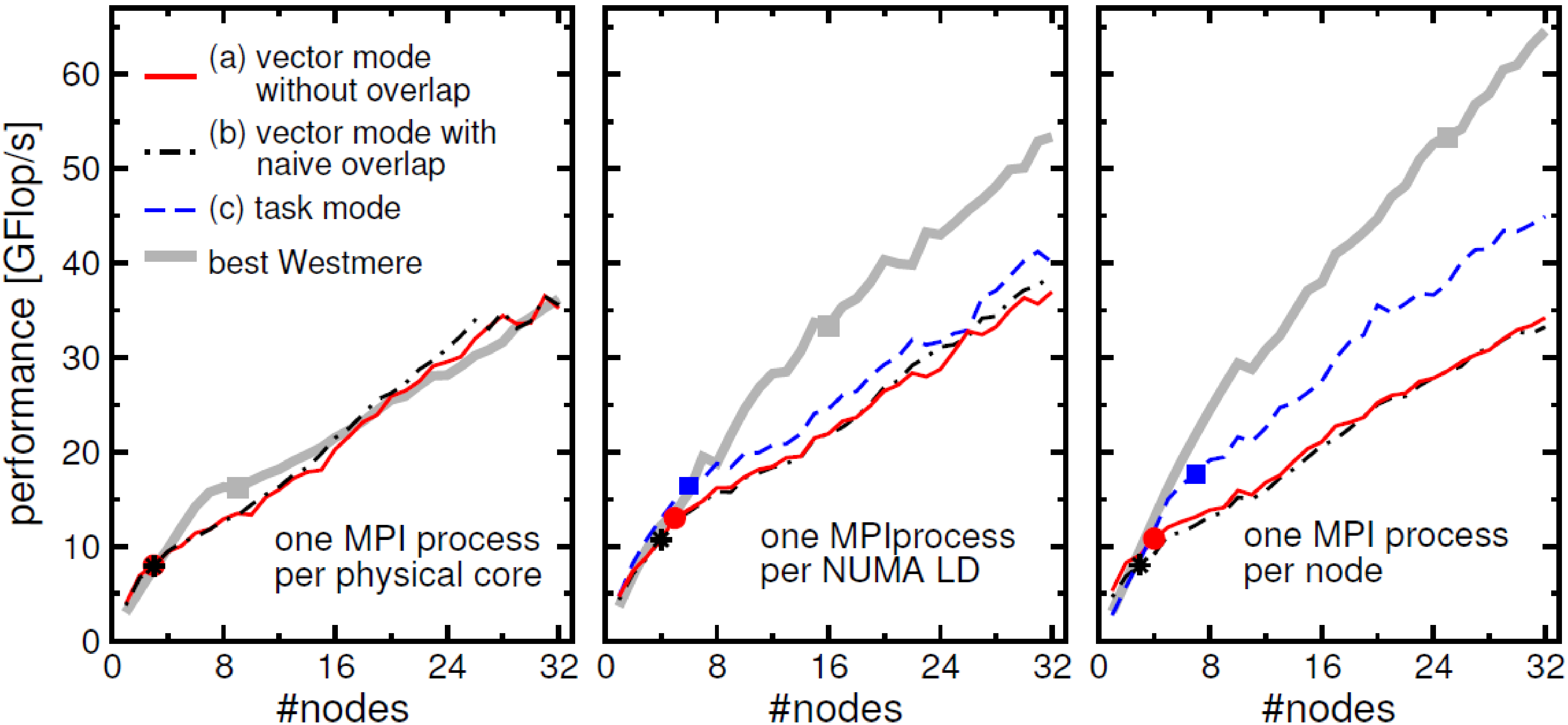
- **Explicit overlap**
- **One thread missing in team of compute threads**
  - But that doesn't hurt here…
- **More complex**
- **Drawbacks**
  - Result vector is written twice to memory
  - No simple OpenMP worksharing (manual, tasking)



R. Rabenseifner and G. Wellein: *Communication and Optimization Aspects of Parallel Programming Models on Hybrid Architectures.* International Journal of High Performance Computing Applications **17**, 49-62, February 2003. DOI:10.1177/1094342003017001005

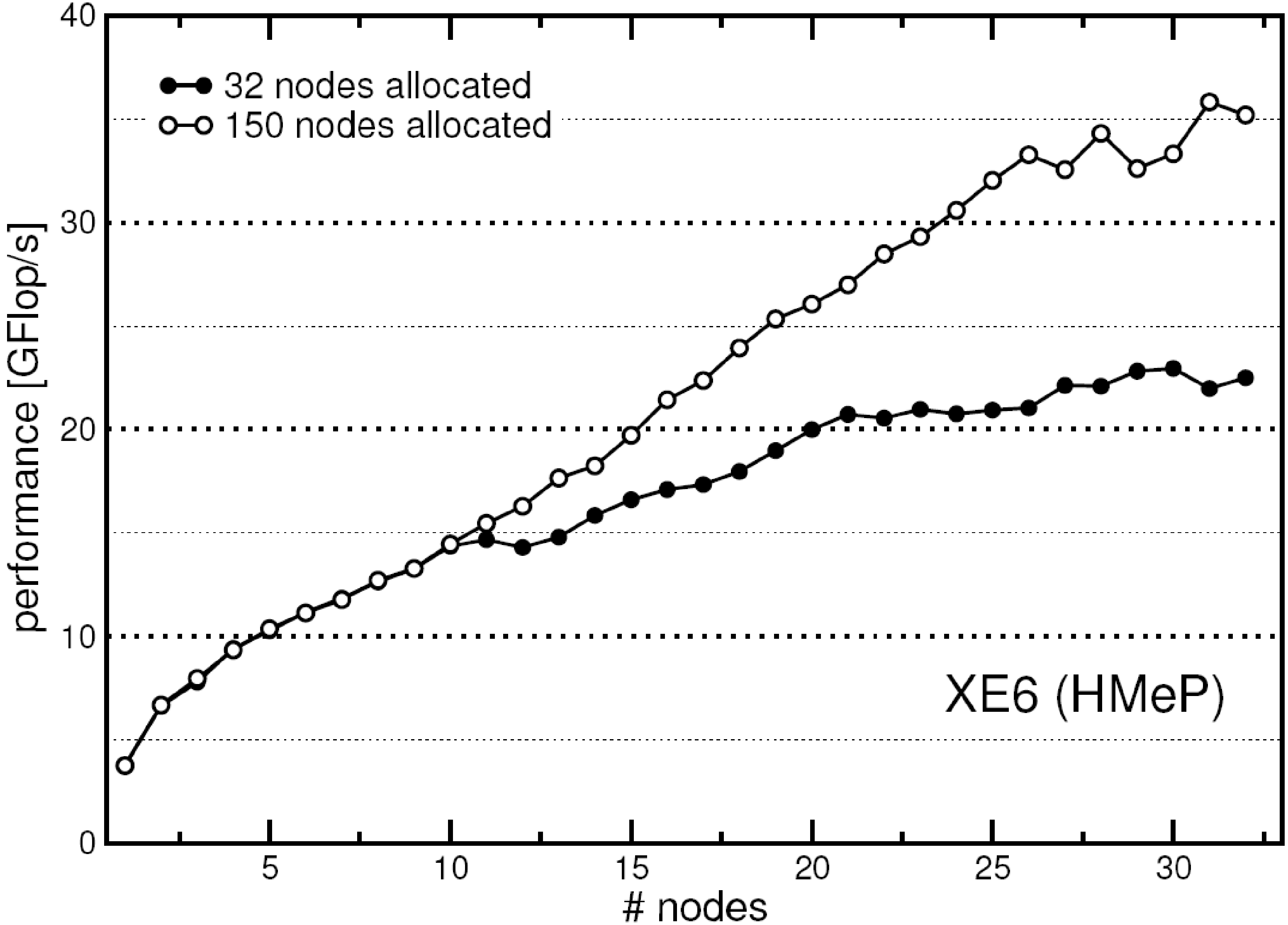
Results HMeP

High Performance Computing

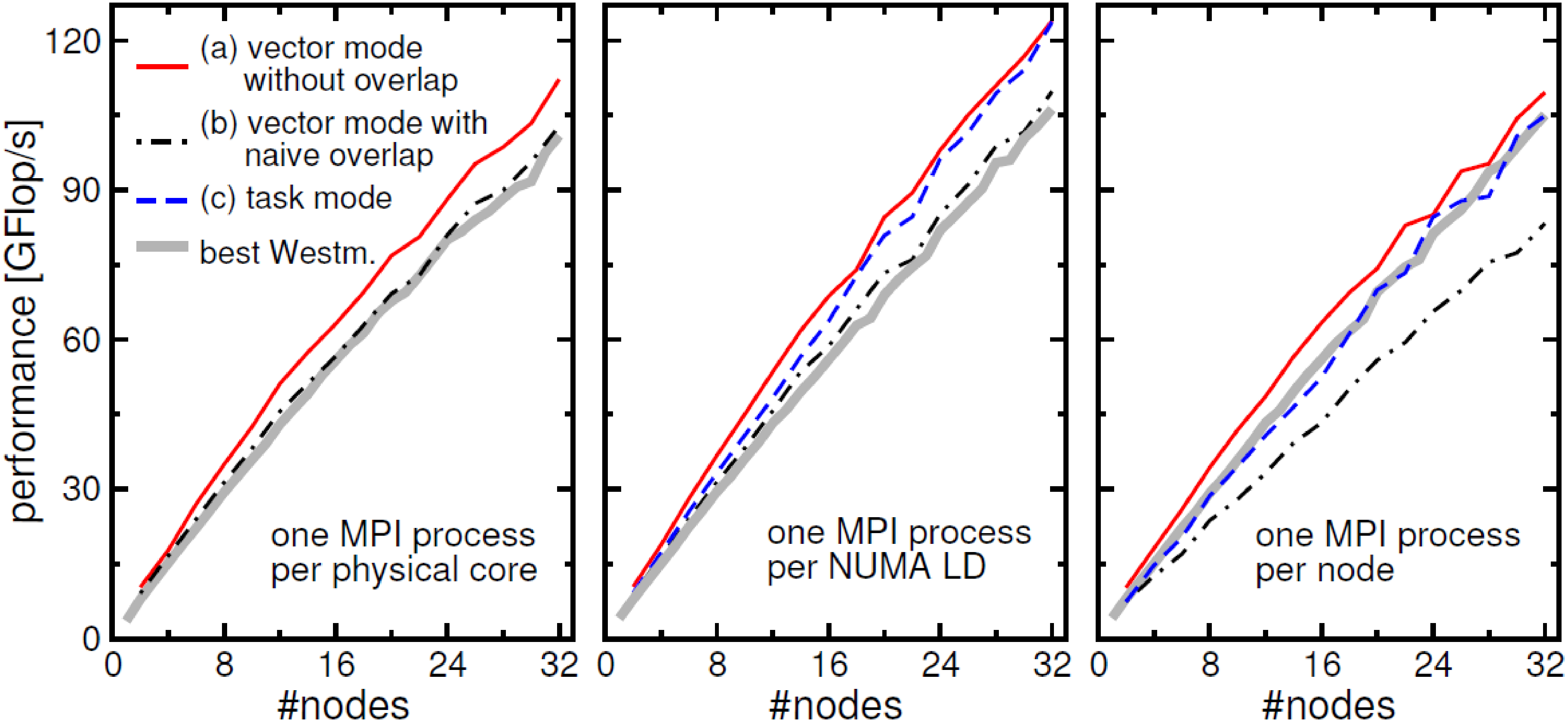


- **Dominated by communication and load imbalance**
- **Single-node Cray performance cannot be maintained beyond a few nodes**
- **Task mode pays off esp. with one process (24 threads) per node**
- **Task mode overlap (over-)compensates additional LHS traffic**

XE6 influence of machine load (pure MPI)



Results sAMG



- **Much less communication-bound**
- **XE5 outperforms Westmere cluster, can maintain good node performance**
- **One process per ccNUMA domain is best, but pure MPI is also ok**
- **If pure MPI is good enough, don't bother going hybrid!**

Conclusions

- Do not rely on asynchronous MPI progress
- Simple "vector mode" hybrid MPI+OpenMP parallelization is not good enough if communication is a real problem
- Sparse MVM leaves resources (cores) free for use by communication threads
- "Task mode" hybrid can truly hide communication and overcompensate penalty from additional memory traffic in spMVM
  - (Not shown here: Comm thread can share a core with comp thread via SMT and still be asynchronous)
- If pure MPI scales ok and maintains its node performance according to the node-level performance model, don't bother going hybrid