ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Application-Centric Benchmarking and Modeling for Co-Design

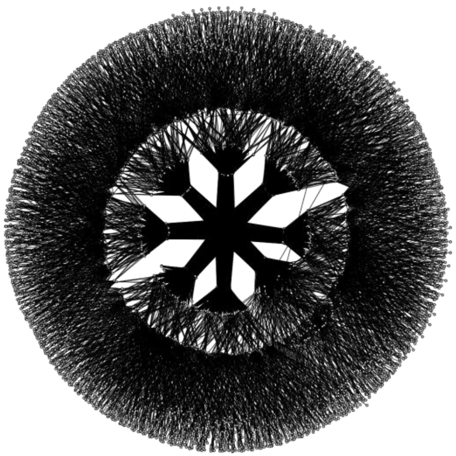
Torsten Hoefler

Exascale Applications and Software Conference

Edinburgh, Scotland, UK

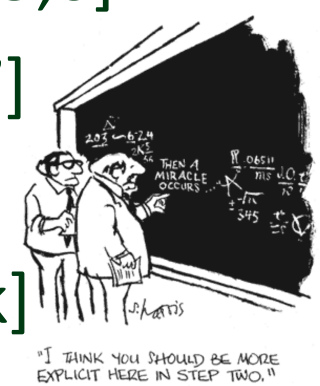
Why Modeling and Co-Design?

- Number of PEs grows exponentially
- Bottlenecks shift quickly
  - e.g., to data serialization
- Some aspects are over-engineered
  - At least for the “average application”
- How do we know what applications need?
  - At scale?



Performance Modeling

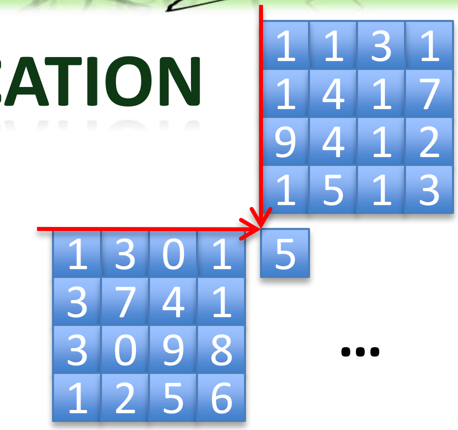
- Allows to:
  1. Predict performance at a different scale [1,2]
  2. Predict performance on a different machine [3,4]
  3. Predict performance for a different problem [5,6]
  4. Assess impact of network topology choices [7]
  5. Find performance bugs [to appear]
  6. Determine application requirements [this talk]



[1]: Zhai et al.:"Phantom: predicting performance of parallel applications on large-scale […]", PPoPP'10
[2]: Lee et al.: "Methods of inference and learning for performance modeling of parallel applications.", PPoPP'07
[3]: Marin, Mellor-Crummey: "Cross-architecture performance predictions for scientific applications […]", SIGMETRICS'04
[4]: Yang et al.: "Cross-platform performance prediction of parallel applications using partial execution.", SC'05
[5]: Hoefler et al.: "Performance modeling for systematic performance tuning", SC'11
[6]: Kerbyson et al.: "Predictive performance and scalability modeling of a large-scale application.", SC'01
[7]: Bauer et al.: "Performance Modeling and Comparative Analysis of the MILC Lattice QCD Application su3 rmd", CCGrid'12

Simplest Example – Matrix Multiplication

```
for(int i=0; i<N; ++i)
  for(int j=0; j<N; ++j)
    for(int k=0; k<N; ++k)
      C[i+j*N] += A[i+k*N] * B[k+j*N];
```

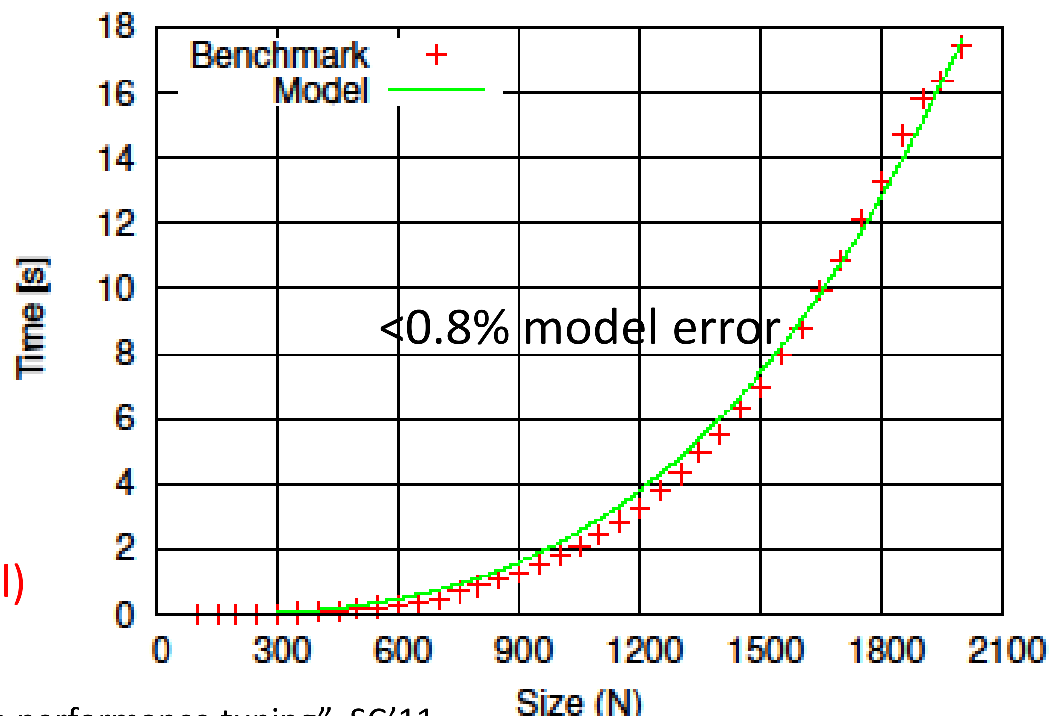


- Semi-Analytic Perf. Modeling [1]:

Algorithmic (analytic) Parameters

- T(N) = tN^3
  - t=2.2ns

Architectural (empirical) Parameters

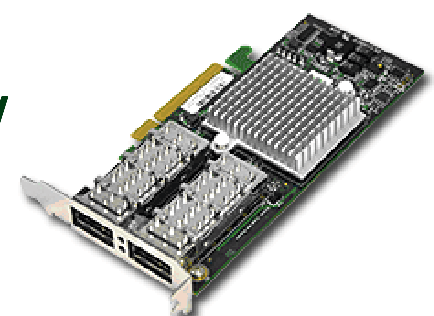


[1]: Hoefler et al.: "Performance modeling for systematic performance tuning", SC'11

Requirements Modeling

- Dual to performance modeling
  - What we **get** vs. what we **need**!
- Allows to model requirements of applications
  - E.g., FLOPs, Memory Bandwidth, …
  - Rules of thumb: "1 Byte/Word per FLOP"
  - Better: "balance principles" [1]

- Burton's interpretation of Little's law
  - concurrency = latency x bandwidth

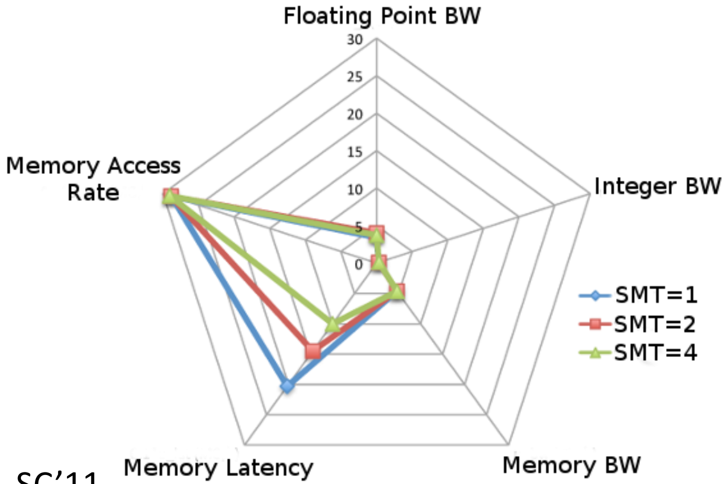


minimum needed — to hide latency — and keep system busy

[1]: Czechowski et al: "Balance Principles for Algorithm-Architecture Co-design", HotPar'11

Model Requirements

- System features
  - Memory/Network Latency, Bandwidth, Flops, …
- Requirements space of system X
  - Feature vector F defines limits
  - Each application run has a requirement vector R
  - In general R < F
  - Bottlenecks are easily identified



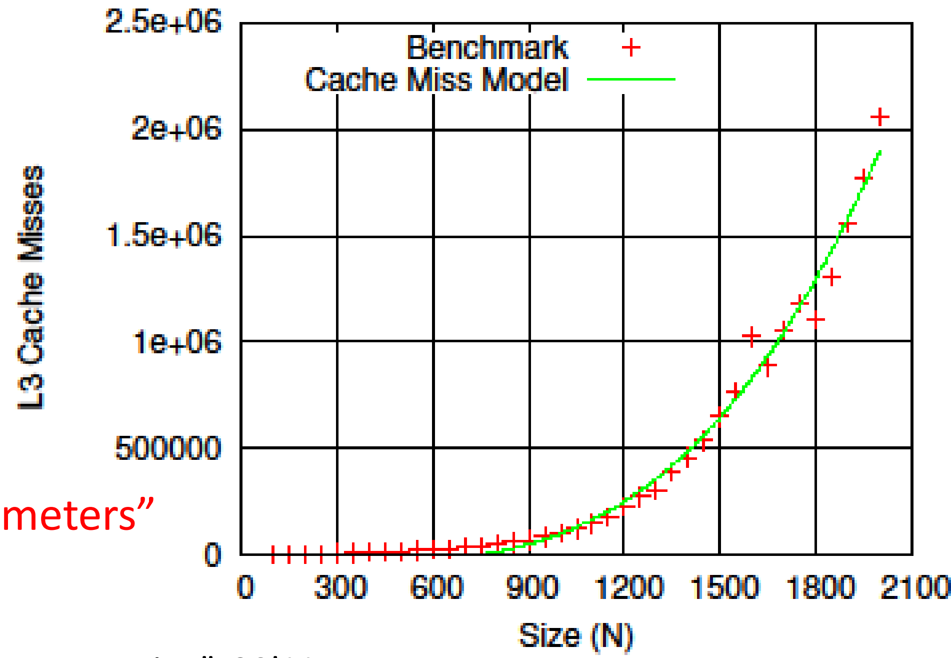
[1]: Hoefler et al.: "Performance modeling for systematic performance tuning", SC'11

Parametric Requirements Model

- Requirements vectors are not enough
  - Back to our simple example: Matrix Multiplication!
    - $2N^3$ FLOP/s
    - Memory bandwidth?
- LLC Misses:
  - $C(N) = aN^3 - bN^2$
  - a=3.8e-4
  - b=2.7e-1

"Algorithmic Parameters"

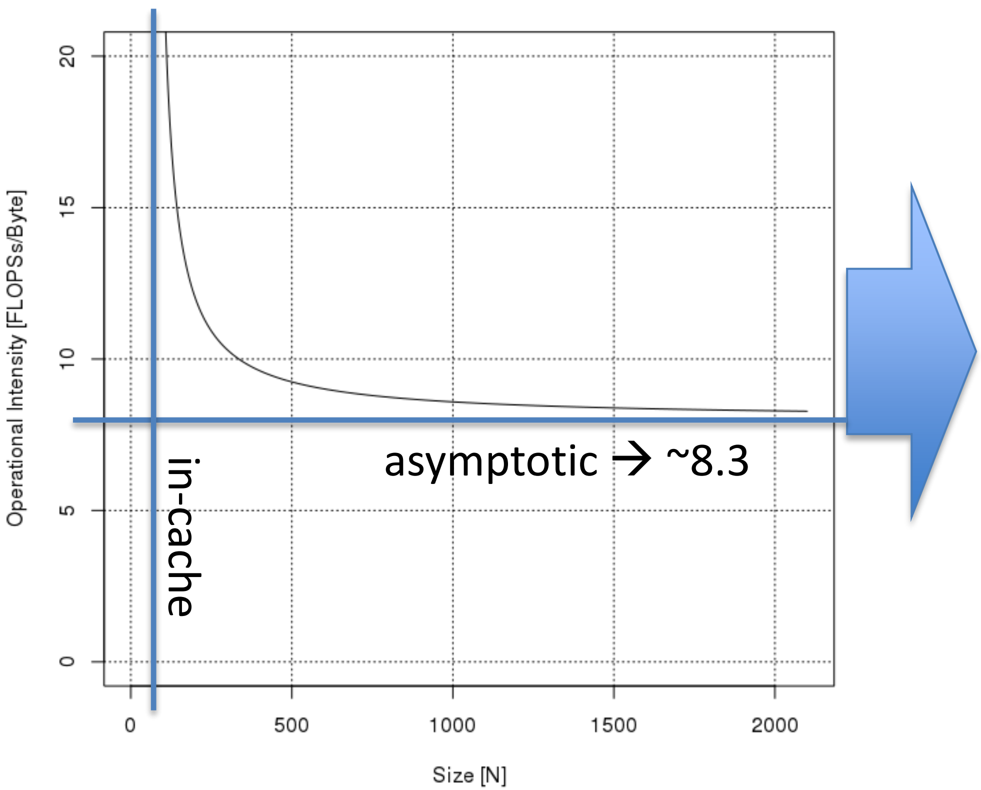
"Architectural Parameters"



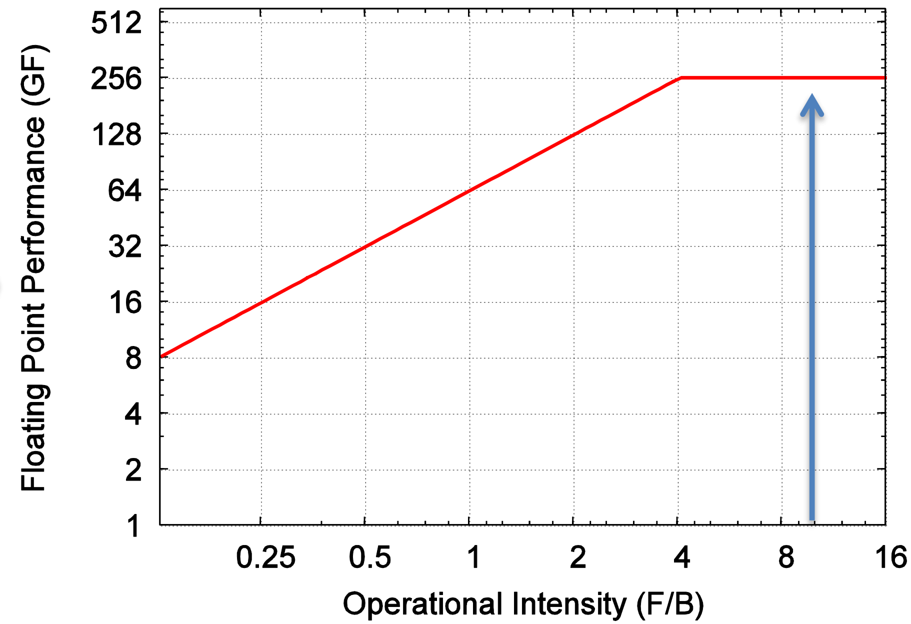
[1]: Hoefler et al.: "Performance modeling for systematic performance tuning", SC'11

How is the Balance?

Operational Intensity



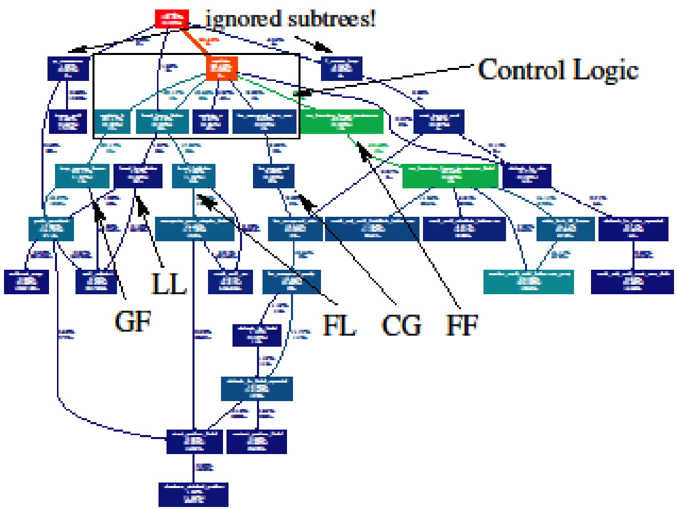
Roofline Visualization

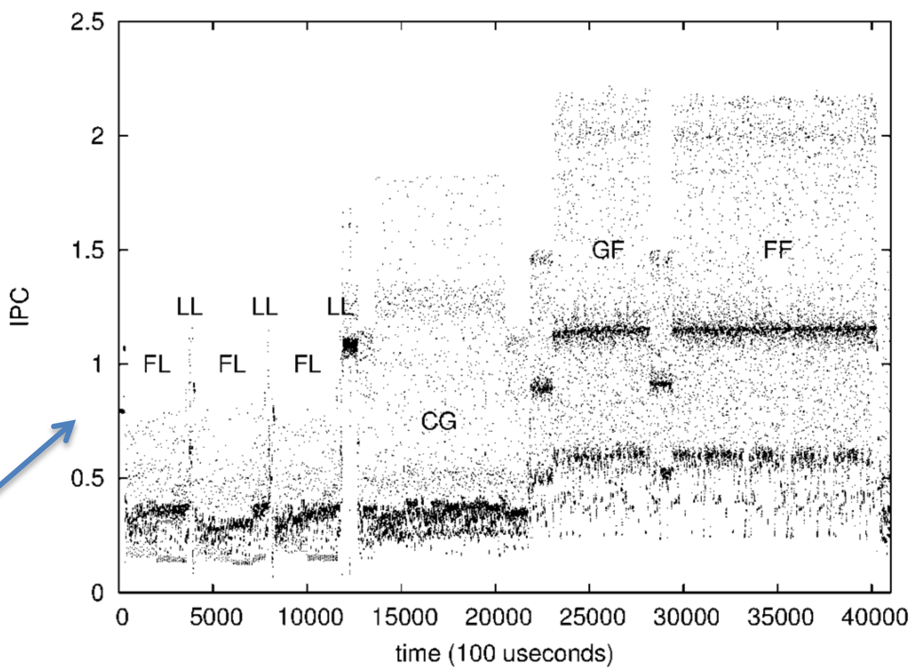


- MM needs < ½ memory bandwidth
  - Small problems even less

A REAL (SIMPLE) APPLICATION

- MILC – MIMD Lattice Computation, su3_rmd
- Well understood and modeled [1]
  - Five phases: FL, LL, CG, GF, FF

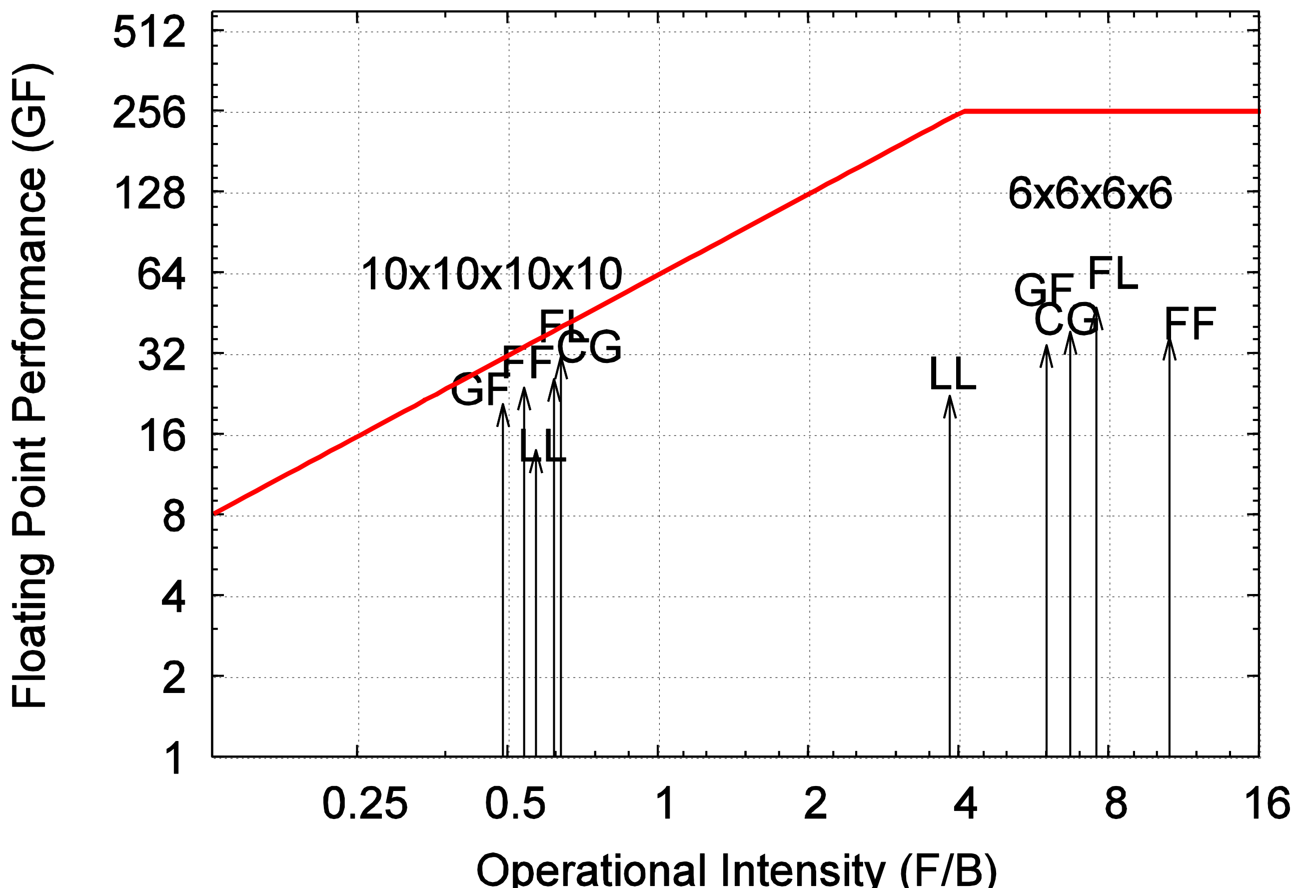




  - Phase detection can be automated

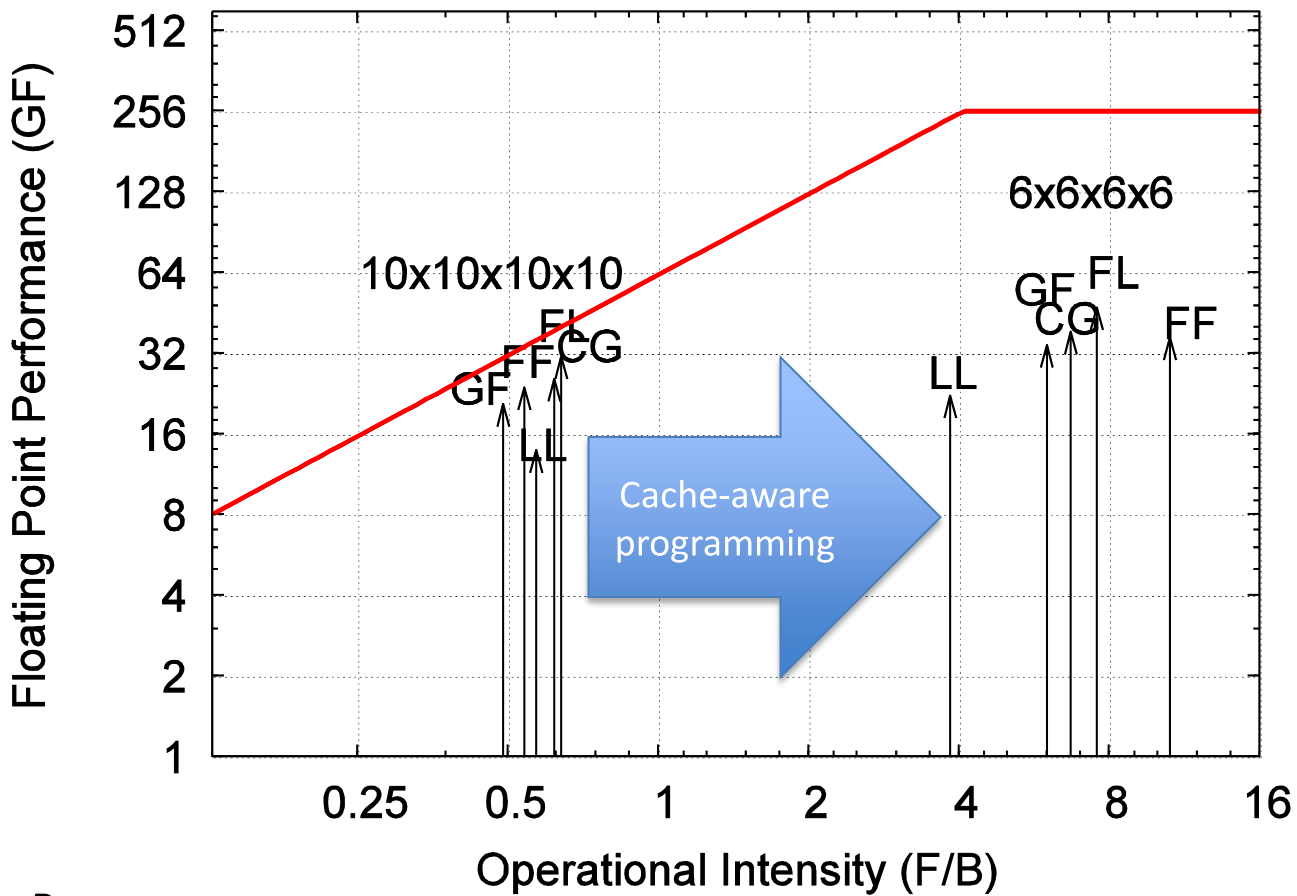
[1]: Bauer et al.: "Performance Modeling and Comparative Analysis of the MILC Lattice QCD Application su3 rmd", CCGrid'12

MILC Balance?



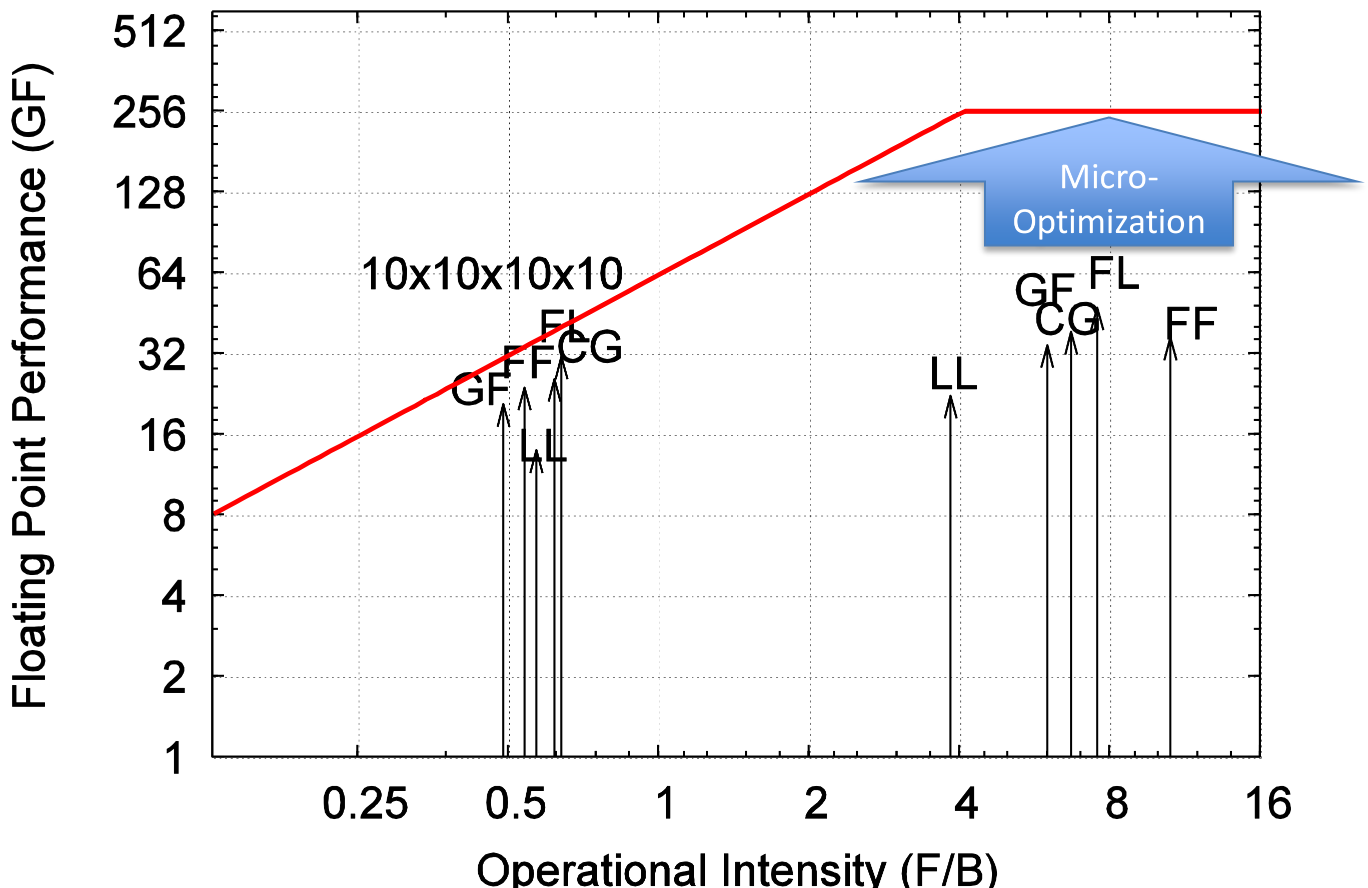
Thanks to Greg Bauer

MILC BALANCE?



Thanks to Greg Bauer

MILC BALANCE?



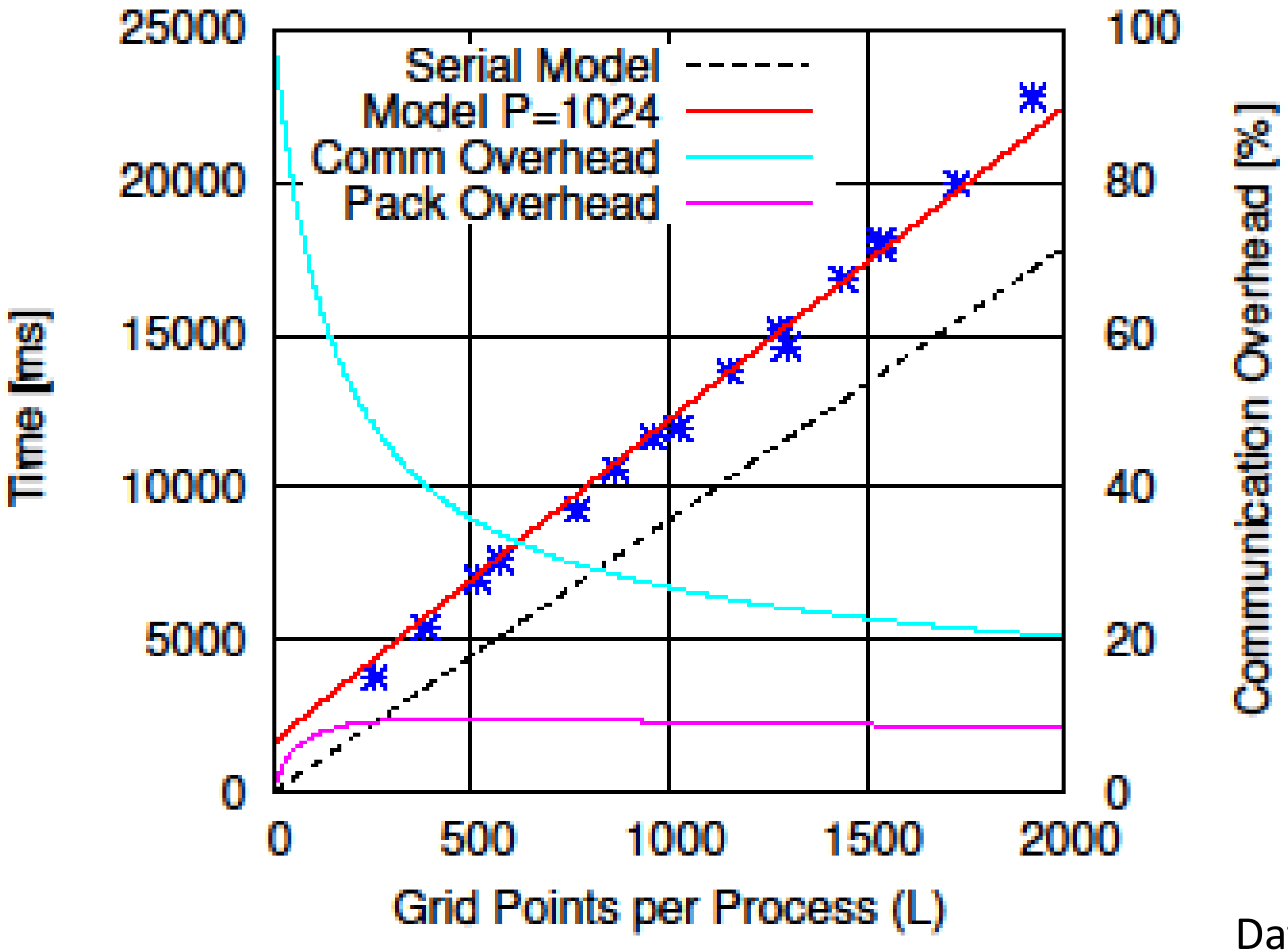
Thanks to Greg Bauer

Simple! What are the Pitfalls?

- Simple strategy for requirements modeling for co-design, isn't it?
  - Can be used by system designers and architects
- Issues:
  - Underestimated requirement functions
    - Working on it, e.g., bug finding!
  - Overlooked requirement dimensions
    - Misguided optimization/-benchmarking!
    - E.g.: Serialization overheads, Routing issues

[1]: Hoefler et al.: "Performance modeling for systematic performance tuning", SC'11

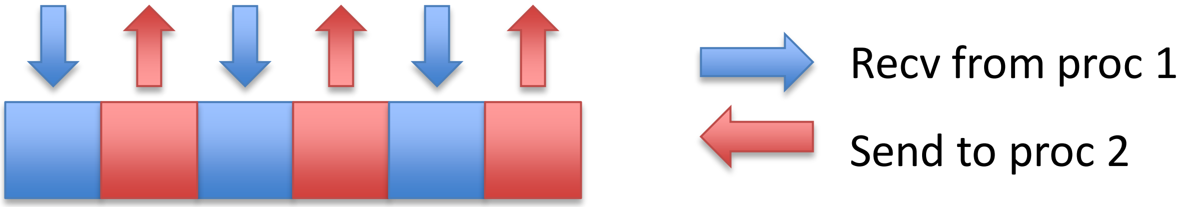
MILC: Full Performance Model



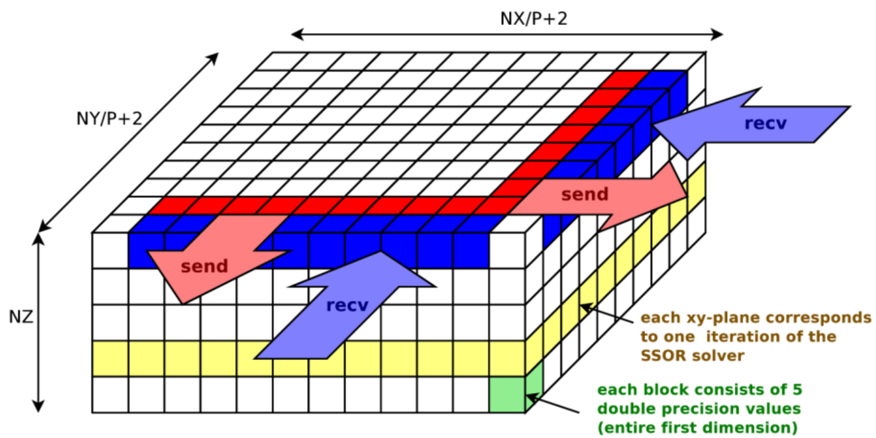
Data from POWER5

Missed Requirement I: Data Serialization

- Networks channels are serial!
- But you want to communicate this pattern:

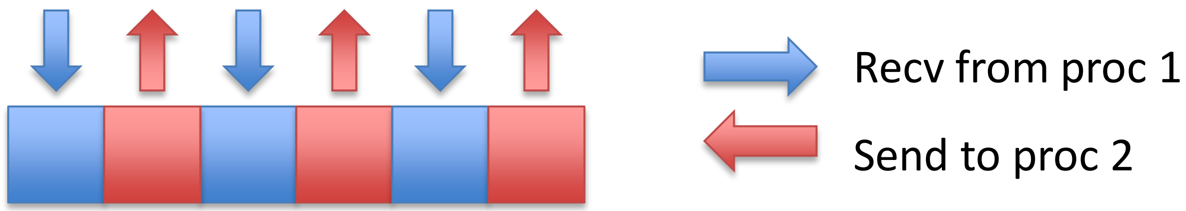


- Or a face- (Cartesian boundary-) exchange:



[1]: Schneider et al.: "Micro-Applications for Communication Data Access Patterns and MPI Datatypes", EuroMPI'12

Missed Requirement I: Data Serialization



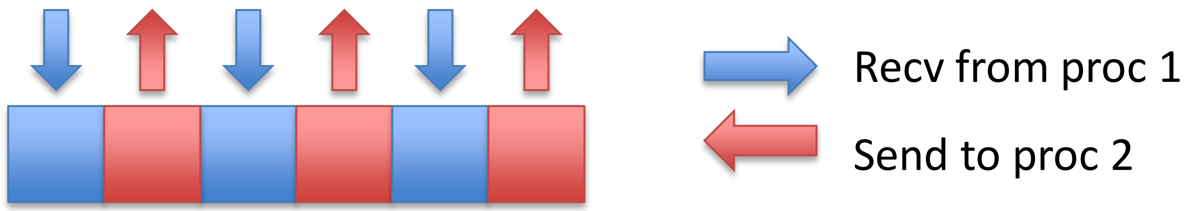
Manual Packing

```
sbuf = malloc(N/2*sizeof(int));
rbuf = malloc(N/2*sizeof(int));
for (i=1; i<N; i+=2) sbuf[i/2]=data[i];
MPI_Isend(sbuf, …);
MPI_Irecv(rbuf, …);
MPI_Waitall(…);
for (i=0; i<N; i+=2) data[i]=rbuf[i/2];
free(sbuf); free(rbuf);
```

Allocate extra send / recv Buffers

[1]: Schneider et al.: “Micro-Applications for Communication Data Access Patterns and MPI Datatypes”, EuroMPI’12

Missed Requirement I: Data Serialization



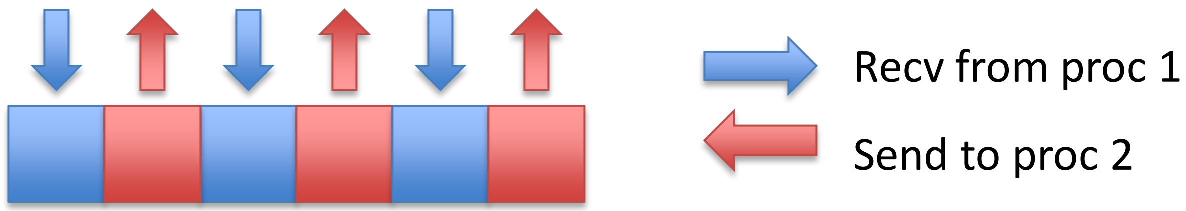
Manual Packing

```
sbuf = malloc(N/2*sizeof(int));
rbuf = malloc(N/2*sizeof(int));
for (i=1; i<N; i+=2) sbuf[i/2]=data[i];
MPI_Isend(sbuf, …);
MPI_Irecv(rbuf, …);
MPI_Waitall(…);
for (i=0; i<N; i+=2) data[i]=rbuf[i/2];
free(sbuf); free(rbuf);
```

Copy data to / from extra send / recv Buffers

[1]: Schneider et al.: "Micro-Applications for Communication Data Access Patterns and MPI Datatypes", EuroMPI'12

Missed Requirement I: Data Serialization



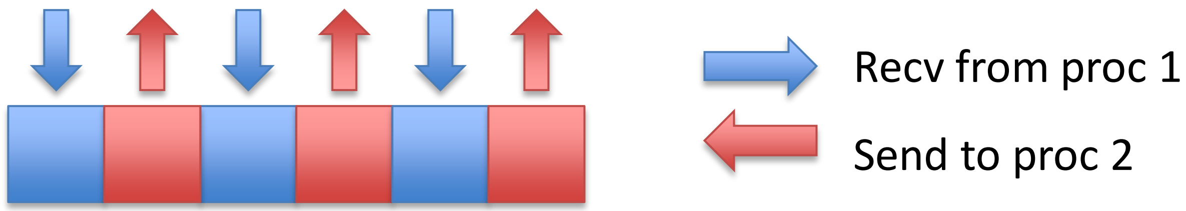
Manual Packing

```
sbuf = malloc(N/2*sizeof(int));
rbuf = malloc(N/2*sizeof(int));
for (i=1; i<N; i+=2) sbuf[i/2]=data[i];
MPI_Isend(sbuf, …);
MPI_Irecv(rbuf, …);
MPI_Waitall(…);
for (i=0; i<N; i+=2) data[i]=rbuf[i/2];
free(sbuf); free(rbuf);
```

Actual sending

[1]: Schneider et al.: “Micro-Applications for Communication Data Access Patterns and MPI Datatypes”, EuroMPI’12

Missed Requirement I: Data Serialization



Manual Packing

```
sbuf = malloc(N/2*sizeof(int));
rbuf = malloc(N/2*sizeof(int));
for (i=1; i<N; i+=2) sbuf[i/2]=data[i];
MPI_Isend(sbuf, …);
MPI_Irecv(rbuf, …);
MPI_Waitall(…);
for (i=0; i<N; i+=2) data[i]=rbuf[i/2];
free(sbuf);
free(rbuf);
```

MPI Datatypes

```
MPI_Datatype nt;
MPI_Type_vector(n/2, 1, 2, MPI_INT, &nt);
MPI_Type_commit(&nt);
MPI_Isend(&data[1], 1, nt, …);
MPI_Irecv(&data[0], 1, nt, …);
MPI_Waitall(…);
MPI_Type_free(&nt);
```

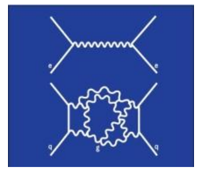
- No explicit copying
- Less code

[1]: Schneider et al.: "Micro-Applications for Communication Data Access Patterns and MPI Datatypes", EuroMPI'12

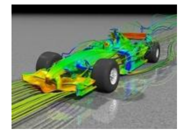
Serialization Access Patterns

Application Classes

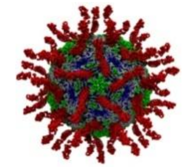
Atmospheric Science (WRF)



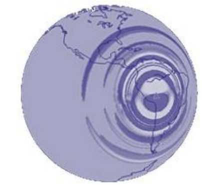
Quantumchro-Modynamics (MILC_su3)



Computational Fluiddynamics (NAS LU + MG)

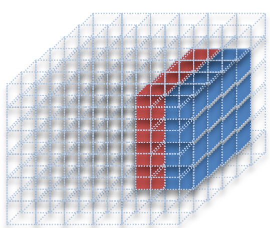


Molecular Dynamics (LAMMPS, MiniMD)

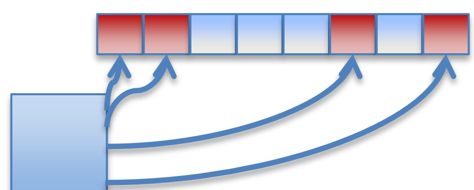


Geophysical Science (SPECFEM3D)

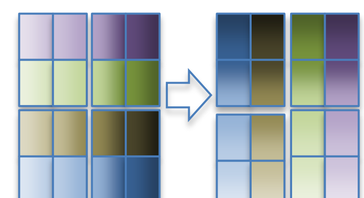
Access Patterns



Face Exchanges



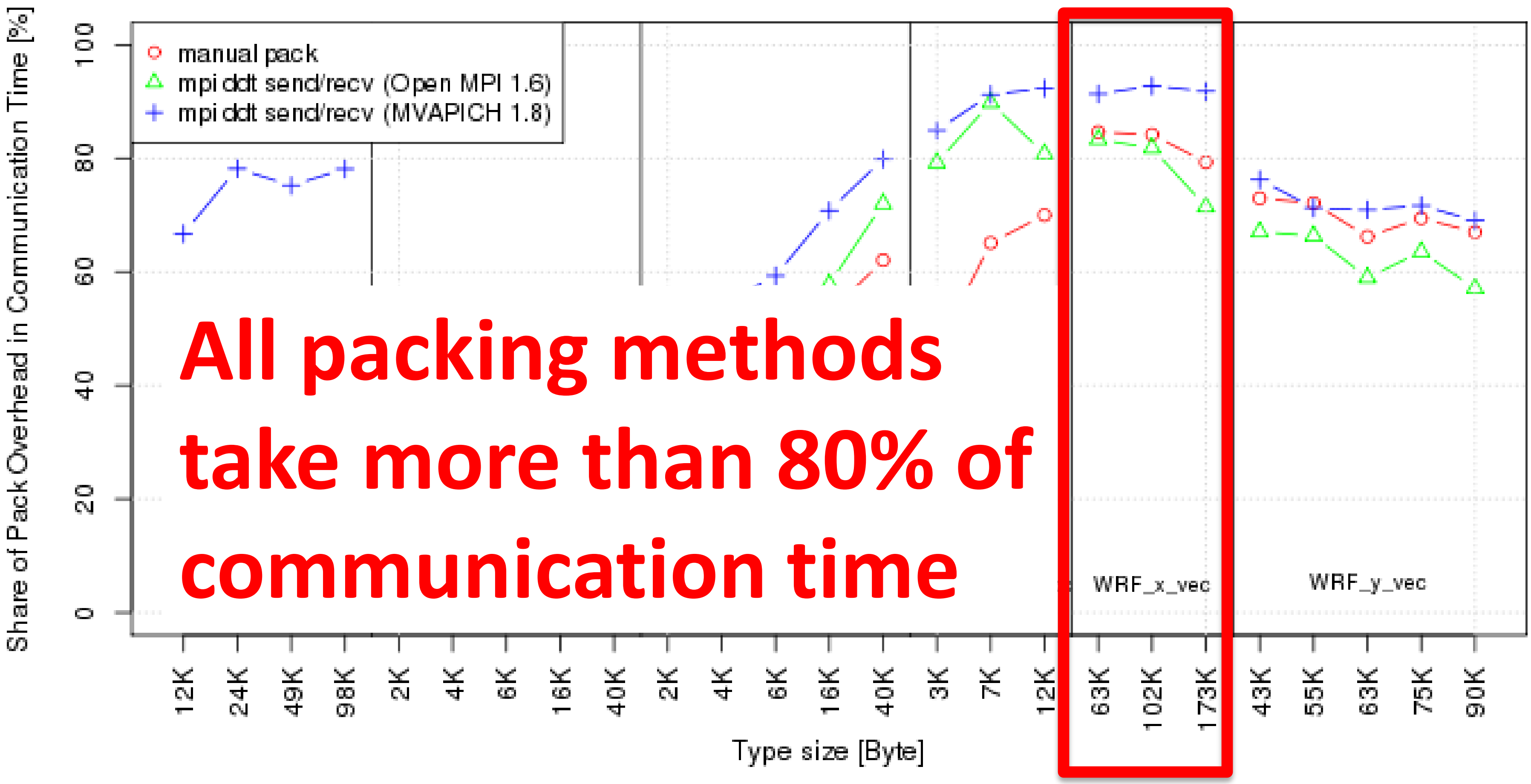
Unstructured Exchange



Matrix Transposition

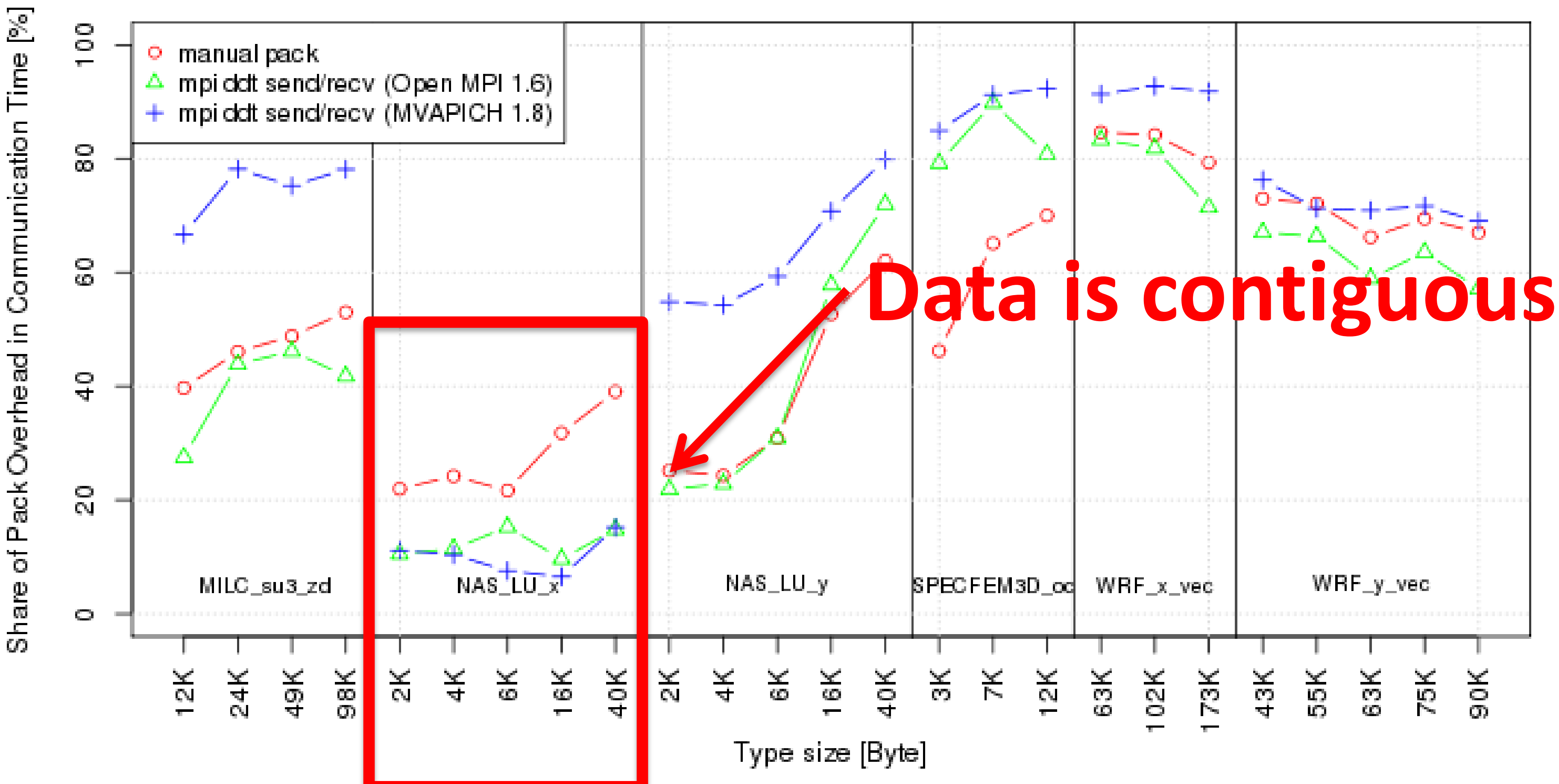
[1]: Schneider et al.: "Micro-Applications for Communication Data Access Patterns and MPI Datatypes", EuroMPI'12

Serialization Benchmark Results



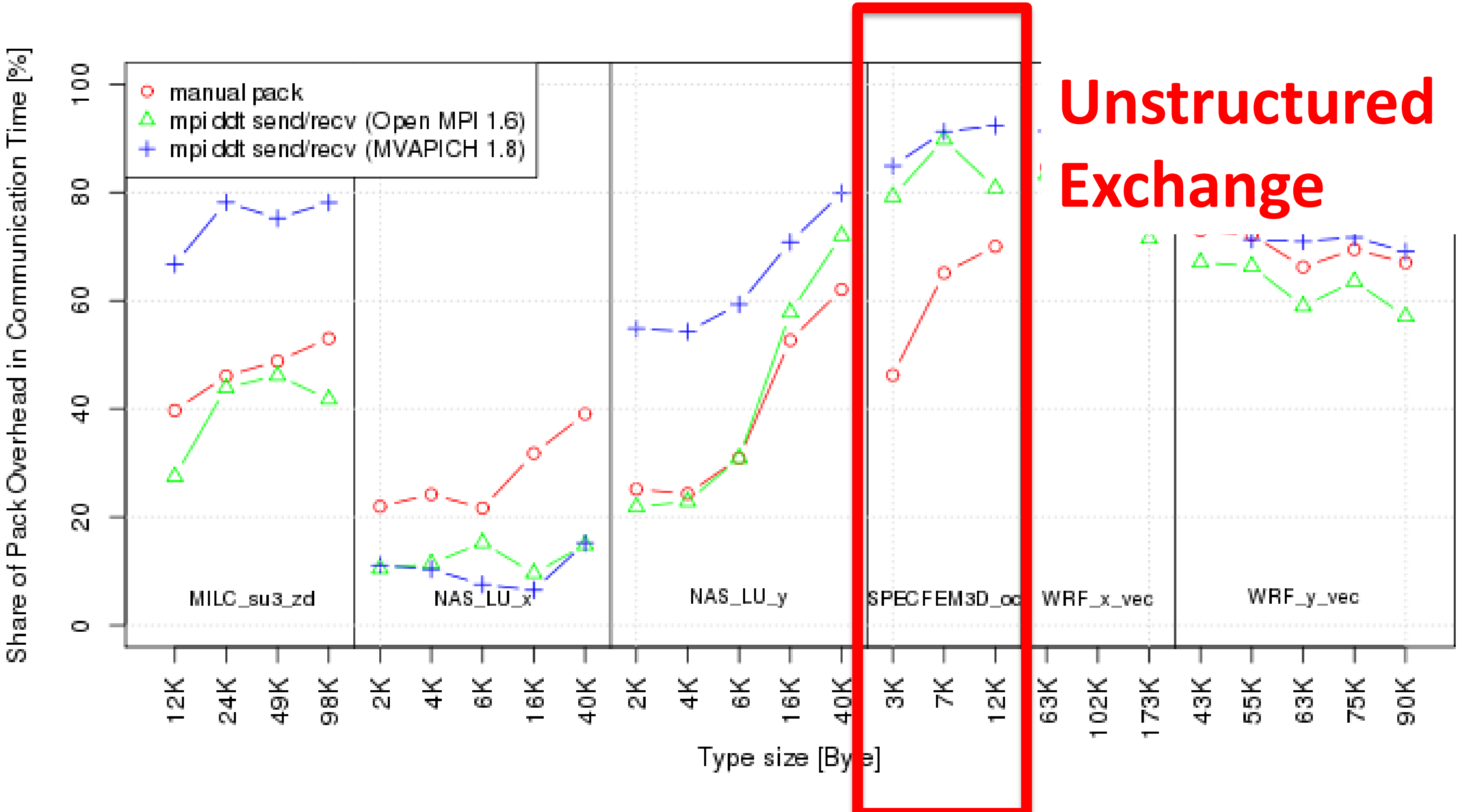
[1]: Schneider et al.: "Micro-Applications for Communication Data Access Patterns and MPI Datatypes", EuroMPI'12

Serialization Benchmark Results



[1]: Schneider et al.: "Micro-Applications for Communication Data Access Patterns and MPI Datatypes", EuroMPI'12

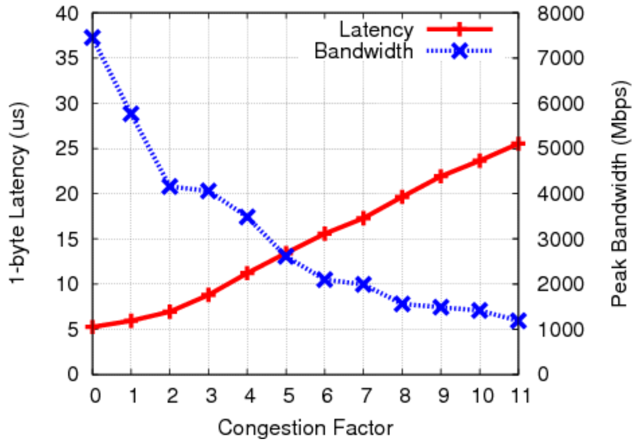
SERIALIZATION BENCHMARK RESULTS



[1]: Schneider et al.: "Micro-Applications for Communication Data Access Patterns and MPI Datatypes", EuroMPI'12

Missed Requirement II: Global Bandwidth

- Observed bandwidth < peak bandwidth
- Example [1]:
  - IB full bisection bandwidth fat tree
  - Effective bandwidth: 69% of peak
  - Reason: static routing



- Conjecture:
  - Routing needs to be co-designed [2]
    - With applications and topologies (complex topic)

[1]: Hoefler et al.: "Multistage Switches are not Crossbars: Effects of Static Routing in High-Performance Networks", Cluster'08
[2]: Prisacari et al.: "Bandwidth-optimal Alltoall Exchanges in Fat Tree Networks", ICS'13

CONCLUSIONS

- Co-design is becoming more important
- Requirements modeling is a simple strategy!
  - Not trivial, dangers:
  - Underestimating requirements function
  - Overlooking requirements dimension
- Tool support on its way
  - As automatic as possible
  - Collaboration with GRS, SPPEXA