BLUE WATERS

SUSTAINED PETASCALE COMPUTING

Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues

Torsten Hoefler, William Gropp, Rajeev Thakur, and Jesper Larsson Träff

EuroMPI 2010, Stuttgart, Germany, Sept. 13th 2010

Imagine …

- … you're planning to construct a multi-million Dollar Supercomputer …
- … that consumes as much energy as a small [european] town …
- … to solve computational problems at an international scale and advance science to the next level …
- … with "hero-runs" of [insert verb here] scientific applications that cost $10k and more per run …

T. Hoefler et al. "Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues"

… and all you have (now) is …

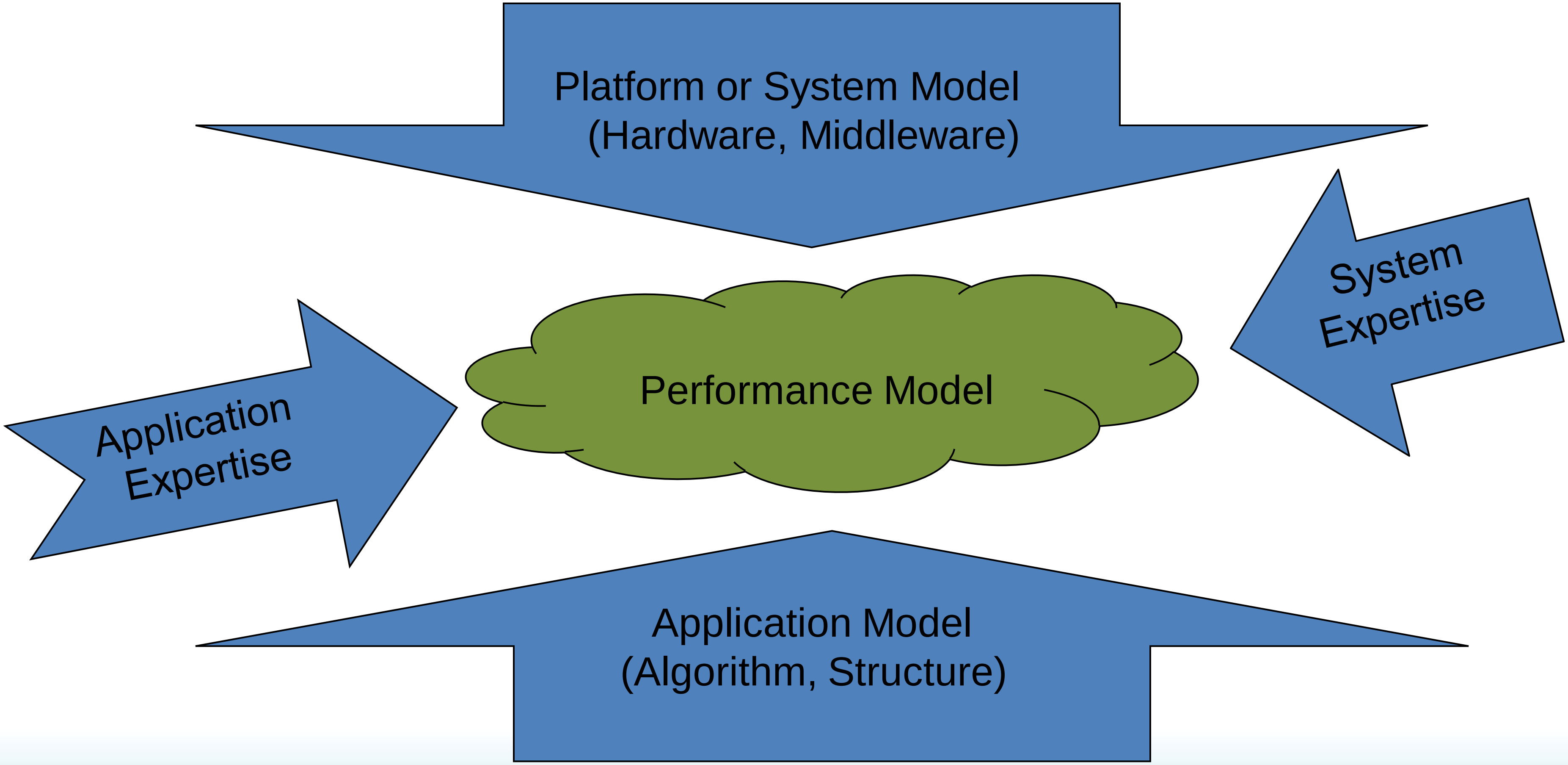
- … then you better plan ahead! (same for Exascale)

Application Performance Modeling

- Analytic expression for expected runtime
  - Often exists as *folklore* in people's minds
- Can estimate scalability of codes and algorithms
  - Folklore: Alltoall(v) is not scalable
- Helps to make design decisions
  - E.g., estimate trade-off between single-core performance and scalability
    - General principle: reduce communication with more/redundant computation

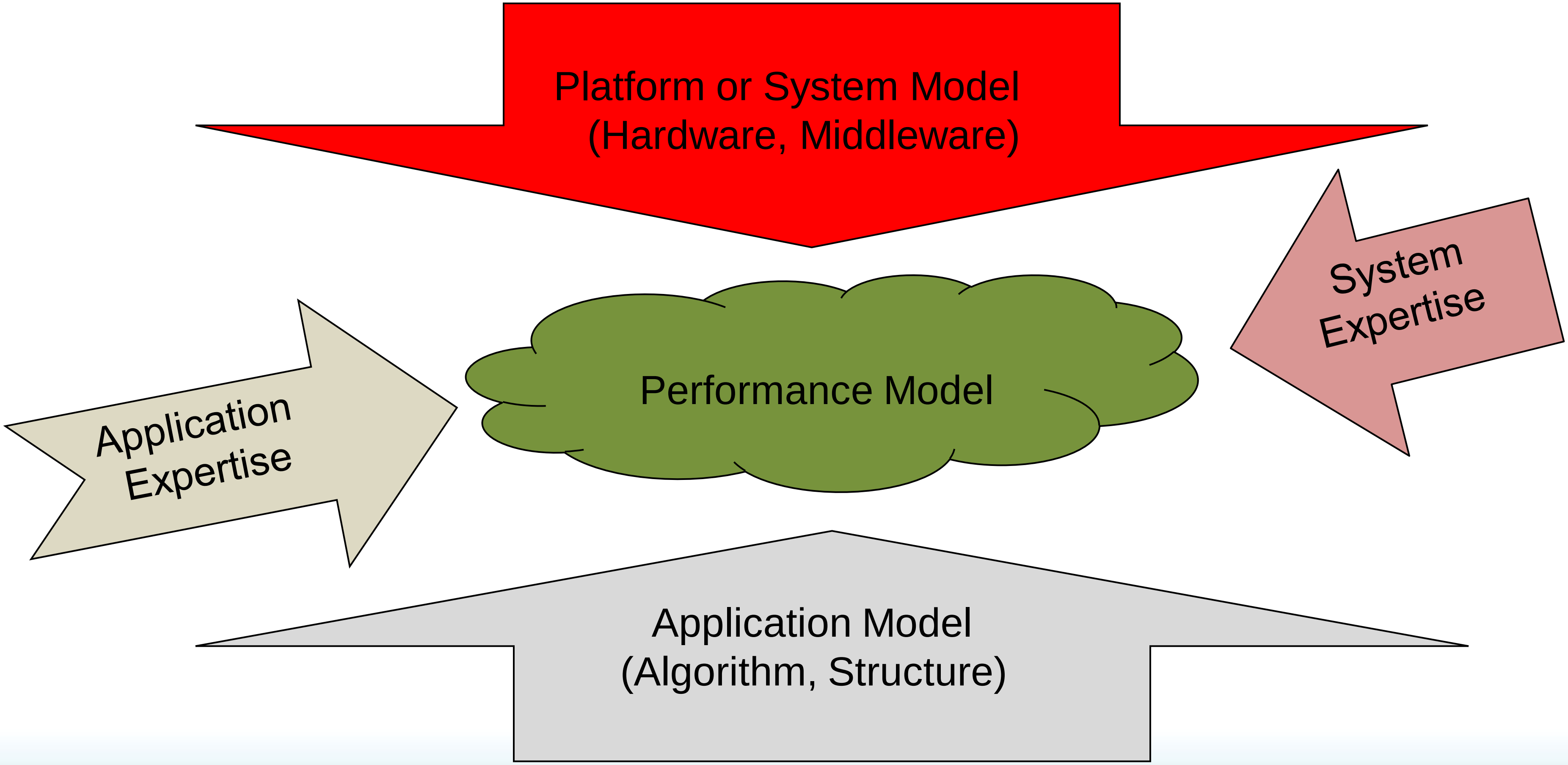
T. Hoefler et al. "Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues"

Performance Modeling from 10.000 Feet



T. Hoefler et al. “Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues”

MPI is part of the Platform Model



T. Hoefler et al. “Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues”

MPI Performance Models are Critical

- Folklore exists, e.g.,
  - Transmitting a message of size S = $\Theta(S)$
  - MPI_Alltoall on P processes = $\mathcal{O}(P)$
- Model inaccuracies often insignificant for small S,P!
  - Huge impact for large S, P
- E.g., application models assume MPI_Bcast or MPI_Allreduce = $\Theta(S \log(P)) \rightarrow$ wrong
  - A good implementation = $\Theta(S + \log(P))$

An Approach to Standardized Models

- Giving accurate estimates is a hard task
  - Users might see it as a contract
  - Vendors are hesitant to agree to or define contracts
- Many variables
  - CPU parameters (memory, speed, architecture, …)
  - Network parameters (latency, bw, topology, …)
  - Protocols (eager, rendezvous, …)
- We propose hierarchical modeling
  - Various levels of accuracy

T. Hoefler et al. "Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues"

The Four Levels of Accuracy

1. **Asymptotic**
   - Asymptotic scaling only, e.g., $\Theta(S + \log(P))$
2. **Dominant term exact**
   - Significant terms, e.g., $\beta S + \mathcal{O}(\log(P)) < 2\beta S + \mathcal{O}(\log(P))$
3. **Bounded (parameterized)**
   - Specify bounds, e.g., $\beta S + \log(P) < T < 2\beta S + \log(P)$
4. **Exact**
   - Exact model (if possible), e.g., $T_{BAR} = 0.95\mu s$

T. Hoefler et al. "Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues"

Current Point-to-Point Models

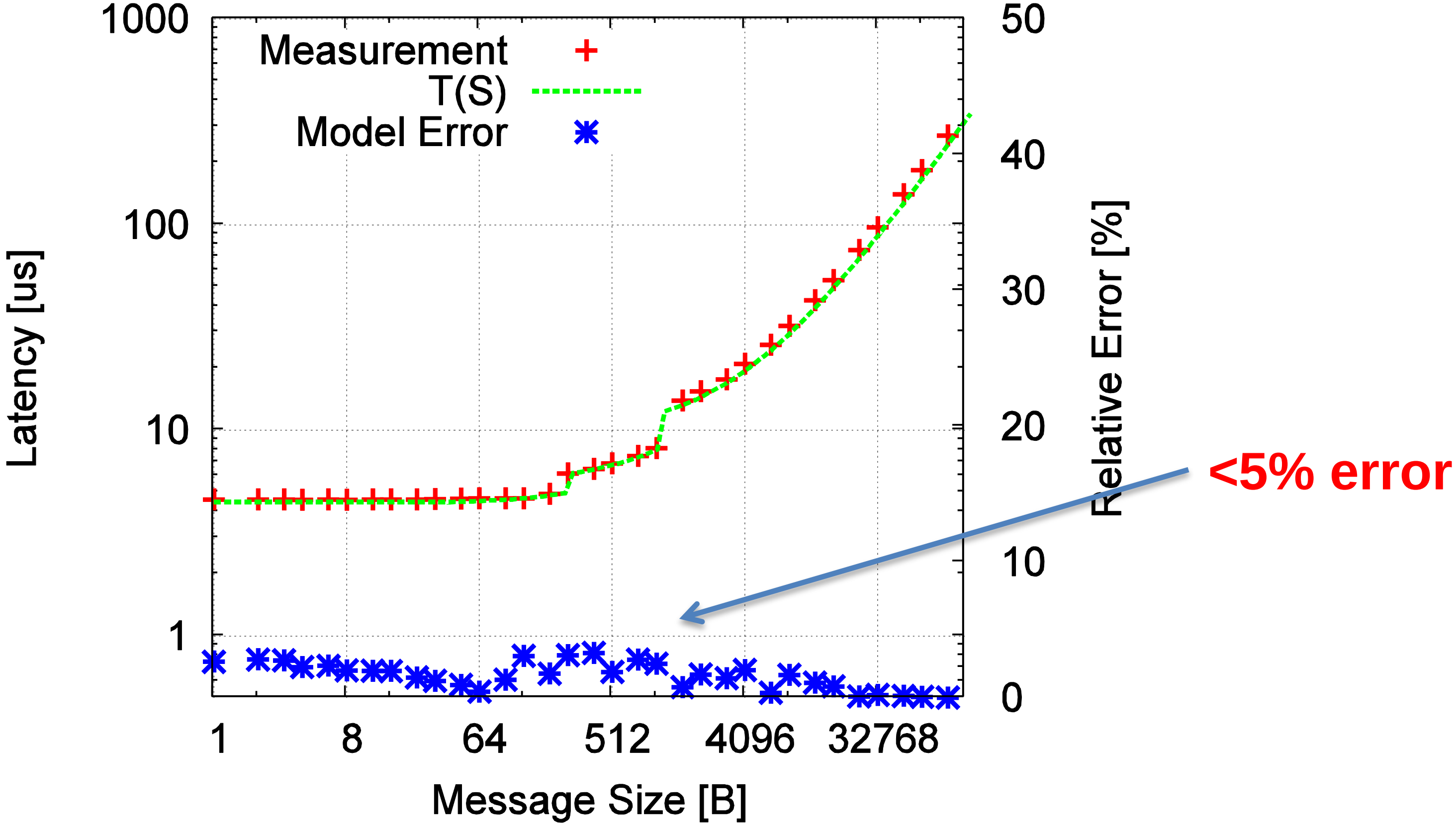
- Asymptotic (trivial): $\Theta(S)$
- Latency-bandwidth models: $T = \alpha + S\beta$
- Need to consider different protocol ranges
- Exact model for BG/P:

$$T(S) = \begin{cases} 4.5\mu s + 2.67ns/B \cdot S: & S \leq 256B \\ 5.7\mu s + 2.67ns/B \cdot S: & 256B < S \leq 1024B \\ 9.8\mu s + 2.67ns/B \cdot S: & 1024B < S \end{cases}$$

- Used Netgauge/logp benchmark
- Three ranges: small, eager, rendezvous

T. Hoefler et al. “Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues”

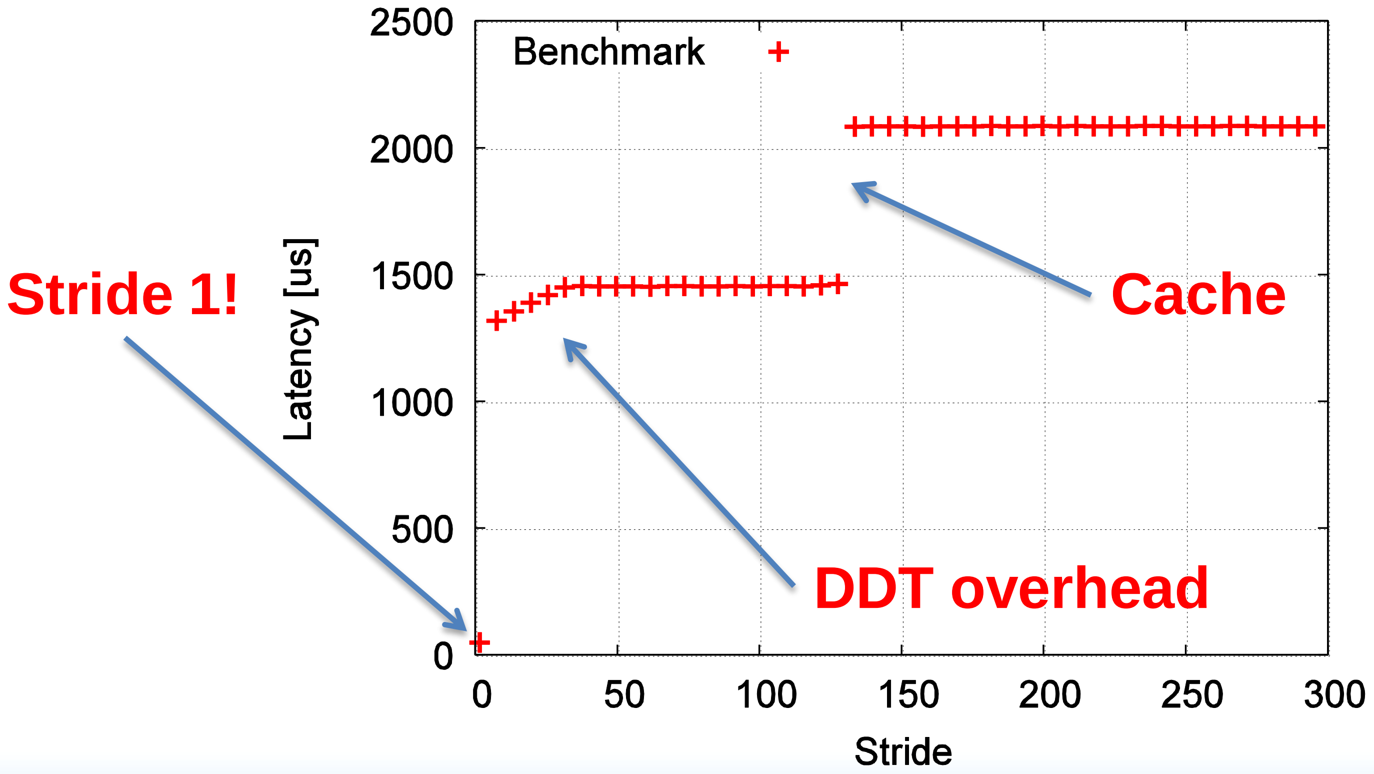
Model Accuracy



- Looks good, but there are problems!

The not-so-ideal (but realistic) Case I

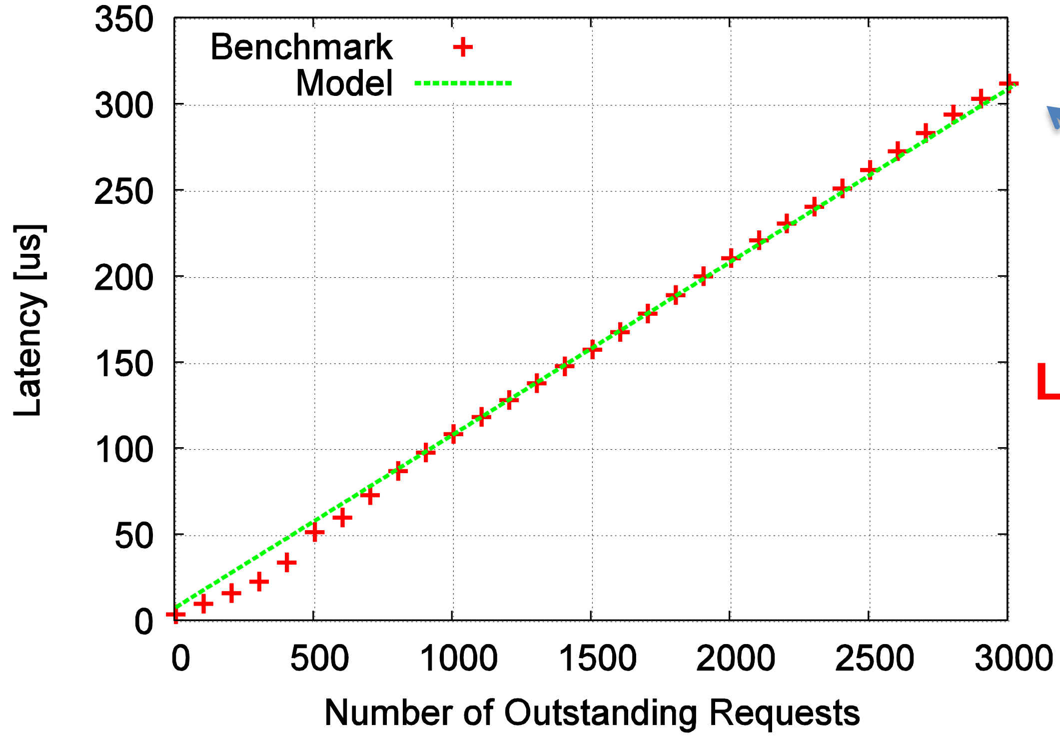
- Strided data-access (model assumed stride-1)



- Benchmark: Netgauge: one_one_dtype

The not-so-ideal (but realistic) Case II

- Matching queue overheads (very common)



Latency factor of 35!

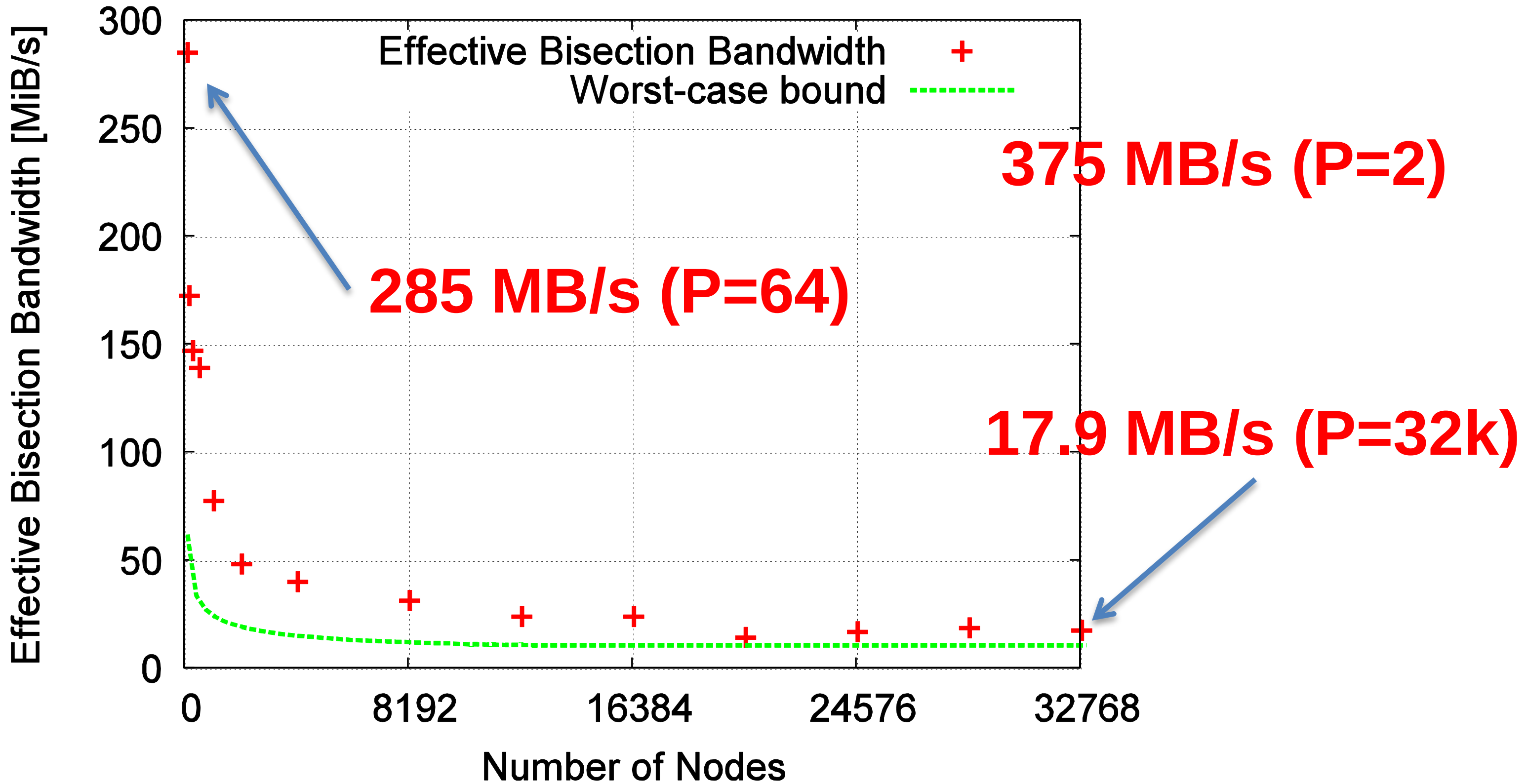
- R requests: $T_{match}(R) \leq 100ns \cdot R; \;\; T(13) \leq 1.3\mu s$
  - Benchmark: Netgauge/one_one_req_queue

The not-so-ideal (but realistic) Case III

- Congestion is often ignored
  - Very hard to determine but worst-case can be calculated
  - effective Bisection Bandwidth
    - Average bandwidth of a random perfect matching
- Upper bound is congestion-less (see model)
- Lower bound assumes worst-case mapping
  - Assume ideal adaptive routing (BG/P)
  - Congestion of $\mathcal{O}(\sqrt[3]{P})$ per link

T. Hoefler et al. “Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues”

Worst-case vs. Average Case Congestion



- Average seems to converge to worst-case (large P)
  - Benchmark: Netgauge/ebb

T. Hoefler et al. "Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues"

Bounded Model with Congestion

$$9.8\mu s + 2.67^{ns}/_{B} \cdot S \leq T(S,P) < 9.8\mu s + 2.67^{ns}/_{B} \cdot S \cdot {}^{3}/_{2}\sqrt[3]{P}$$

- Only considers congestion
  - Needs to add datatypes, matching queue, …
- Some parameters might be ignored
  - Typically application-specific
  - E.g., application only uses stride-1 access

Collective Communication

- Crucial for estimating application scalability
- Often simpler to use than p2p models
  - Matching queue, synchronization and cong. hidden
- Simple latency-bandwidth: $T = \alpha(P) + S \cdot \beta(P)$
  - Empirical parameterization, might be hard to use
- Build upon point-to-point models
  - E.g., LogGOPS model
  - More complex to derive (and often less precise)

T. Hoefler et al. "Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues"

Example: Small MPI_Bcast, MPI_Allreduce

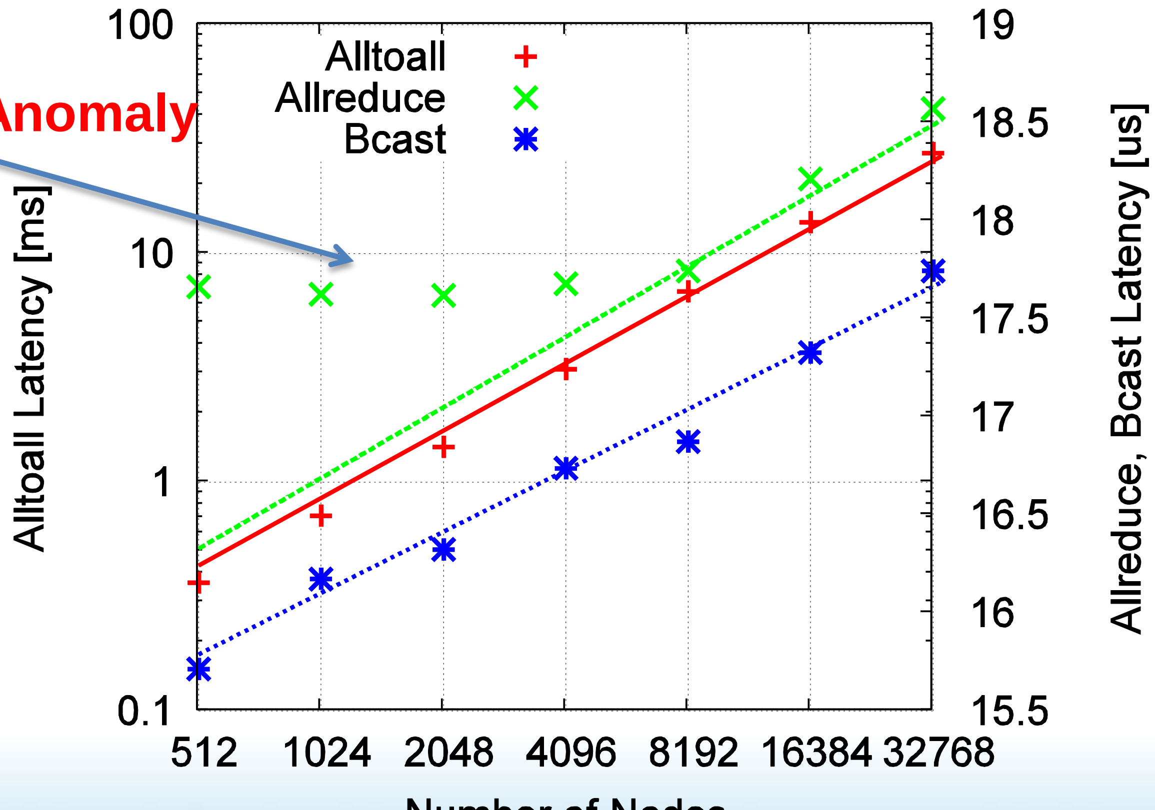
- Handled by Collective (Tree) network in BG/P
- MPI_Bcast: $T_{BC}(P,8) = \alpha_T + \beta_T^{BC} \log_2(P)$
  - α_T = startup overhead
  - β_T = cost per stage
  - Empirically determined: $\alpha_T = 13\mu s,\ \beta_T^{BC} = 0.31\mu s$
- MPI_Allreduce: $T_{ARE}(P,8) = \alpha_T^{SUM} + \beta_T^{SUM} \log_2(P)$
  - Empirically determined: $\beta_T^{SUM} = 0.37\mu s$

T. Hoefler et al. “Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues”

Small MPI_Alltoall

- Direct sends (**not** optimal!): $T_{A2A}(P,8) = \alpha + g(P-1)$

Performance Anomaly



T. Hoefler et al. “Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues”

Large MPI_Bcast

- Exact model for small MPI_Bcast
- Large bcast needs $T_{BC}(P,8)$ to reach all processes
  - Extend with bandwidth term
- Assume using all six torus links (2.67ns/B/link)
  - $T_{BC}(P,S) = \alpha_T + \beta_T^{BC}\log_2(P) + \frac{2.67}{6}ns/B \cdot S$
- Based on first-principles (documentation+LogGP)
  - No fit necessary
  - Still accurate for large S (see next slides)
    - Middle-ground can be covered by less accurate models

Large MPI_Alltoall

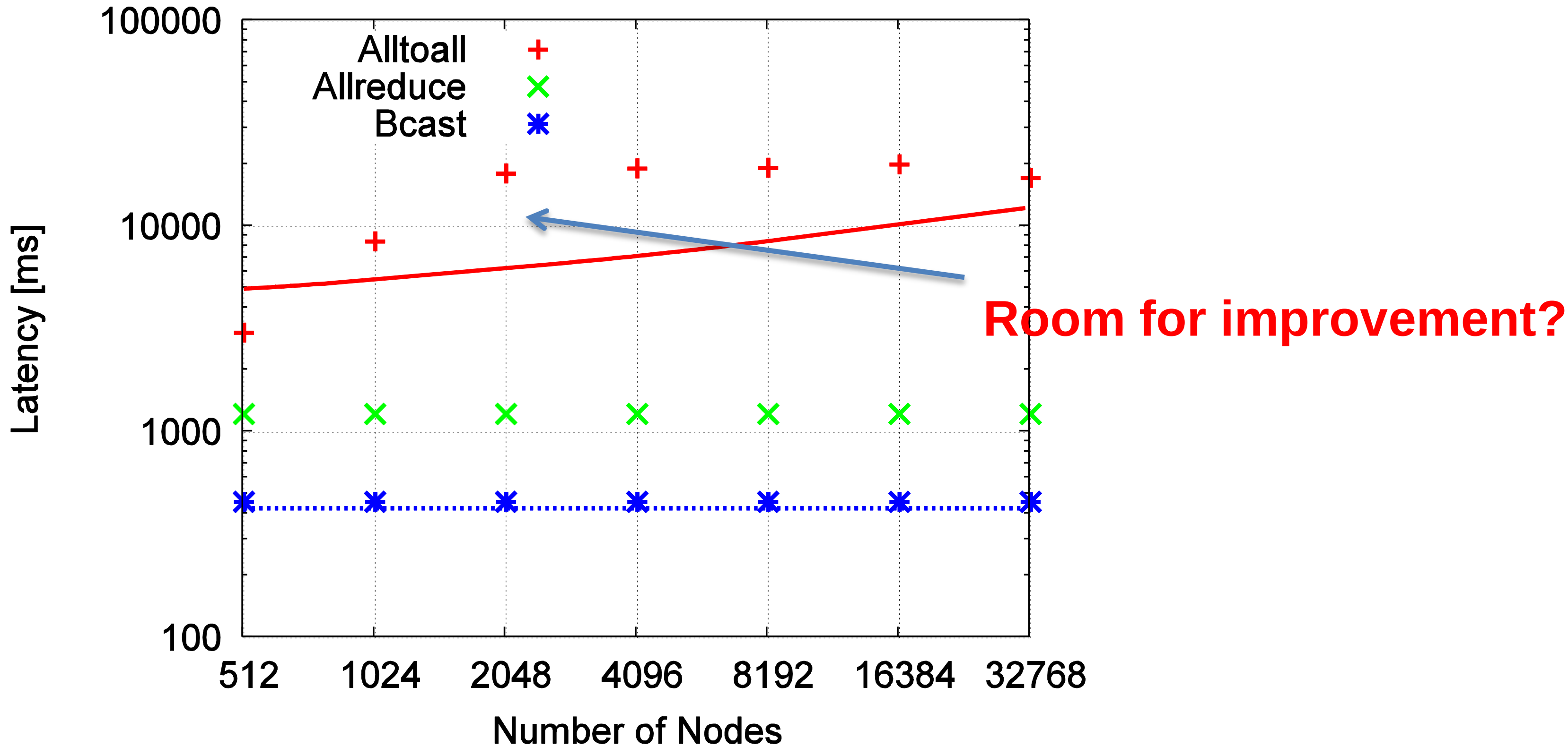
- Algorithm sends to all targets (random permutation)
  - Uses all six links, hits bisection bandwidth
  - Model: $T_{A2A} = (P-1)g + SG \cdot \max\{\frac{P-1}{6}, C(P)\}$
- Simple counting argument for C(P)
  - Assume k^3 processor grid, 1-d distance $d = \frac{k-1}{2}$
  - Total number of occupied links:

$$N(k) \leq k^3 \cdot 2\sum_{x=0}^{d} 2\sum_{y=0}^{d} 2\sum_{z=0}^{d}(x+y+z) = k^3 12d\,(d+1)^3 = \mathcal{O}(k^7)$$

  - Total number of links: $6k^3$
  - Congestion per link: $C(P) \leq \sqrt[3]{P}(\sqrt[3]{P}/2+1)^3 = \mathcal{O}(P\sqrt[3]{P})$
    - Model: $T_{A2A} \geq g(P-1) + SG\sqrt[3]{P}(\sqrt[3]{P}/2+1)^3$

T. Hoefler et al. "Toward Performance Models of MPI Implementations for Understanding Application Scaling Issues"

Results for Large Collectives



- No model for MPI_Allreduce (constant)
- Lower-bound for MPI_Alltoall holds

Takeaways, Questions & Discussion

- MPI Libraries should provide performance models!
  - Different levels allow accuracy/effort trade-off
    - Even black-box models work
  - Models greatly benefit application design
  - Models are useful to check performance consistency
  - Reasoning about large-scale
    - Is the current MPI performance sufficient?
  - Model design is a community effort!
    - More research needed!