CS 498
Hot Topics in High Performance Computing

Networks and Fault Tolerance

3. A Network-Centric View on HPC

Intro

- What did we learn in the last lecture
  - SMM vs. DMM architecture and programming
  - Systolic Arrays, Dataflow, Flynn's classification
    - Including architectural tradeoffs
  - A simple latency/bandwidth model
- What will we learn today
  - More about broadcasts
  - Optimality criteria
  - An asymptotically optimal algorithm

Why Broadcast?

- Broadcast is equivalent to reduction!
- Both are very important
  - Bcast is the central communication operation in HPL
  - (All)Reduce is most important
    - We've seen it in our compute pi example!
  - Algorithms can be used for any data-distribution problem!
    - E.g., streaming video (adjust optimal packet size)
- It's simple! (wait for scatter/gather)

Quick Example

- Simplest linear broadcast
  - One process has a data item to be distributed to all processes
- Sending s bytes to P processes:
  - $T(s) = P * (\alpha+\beta s) = \mathcal{O}(P)$
- Class question: Do you know a faster method to accomplish the same?

k-ary Tree Broadcast

- Origin process is the root of the tree, passes messages to k neighbors which pass them on
  - k=2 -> binary tree
- Class Question: What is the broadcast time in the simple latency/bandwidth model?

k-ary Tree Broadcast

- Origin process is the root of the tree, passes messages to k neighbors which pass them on
  - k=2 -> binary tree
- Class Question: What is the broadcast time in the simple latency/bandwidth model?
  - $T(s) = \lceil log_k(P) \rceil \cdot k \cdot (\alpha + \beta \cdot s) = \mathcal{O}(log(P))$
- Class Question: What is the optimal k?

k-ary Tree Broadcast

- Origin process is the root of the tree, passes messages to k neighbors which pass them on
  - k=2 -> binary tree
- Class Question: What is the broadcast time in the simple latency/bandwidth model?
  - $T(s) = \lceil log_k(P) \rceil \cdot k \cdot (\alpha + \beta \cdot s) = \mathcal{O}(log(P))$
- Class Question: What is the optimal k?

$$0 = \frac{ln(P) \cdot k}{ln(k)} \frac{d}{dk} = \frac{ln(P) ln(k) - ln(P)}{ln^2(k)} \rightarrow k = e = 2.71...$$

  - Independent of P, α, βs? Really?

Faster Trees?

- Class Question: Can we broadcast faster than in a ternary tree?

Faster Trees?

- Class Question: Can we broadcast faster than in a ternary tree?
  - Yes because each respective root is idle after sending three messages!
  - Those roots could keep sending!
  - Result is a k-nomial tree
    - For k=2, it's a binomial tree
- Class Question: What about the runtime?

Faster Trees?

- Class Question: What about the Runtime?
  - $T(s) = \lceil log_k(P) \rceil \cdot (k-1) \cdot (\alpha + \beta \cdot s) = \mathcal{O}(log(P))$
- Class Question: What is the optimal k here?

Faster Trees?

- Class Question: What about the Runtime?
  - $T(s) = \lceil log_k(P) \rceil \cdot (k-1) \cdot (\alpha + \beta \cdot s) = \mathcal{O}(log(P))$
- Class Question: What is the optimal k here?
  - T(s) d/dk has no minimum for k>1 and is monotonic, thus k_{opt}=2

Faster Trees?

- Class Question: What about the Runtime?
  - $T(s) = \lceil log_k(P) \rceil \cdot (k-1) \cdot (\alpha + \beta \cdot s) = \mathcal{O}(log(P))$
- Class Question: What is the optimal k here?
  - T(s) d/dk has no minimum for k>1 and is monotonic, thus $k_{opt}=2$
- Class Question: Can we broadcast faster than in a k-nomial tree?

Faster Trees?

- Class Question: What about the Runtime?
  - $T(s) = \lceil log_k(P) \rceil \cdot (k-1) \cdot (\alpha + \beta \cdot s) = \mathcal{O}(log(P))$
- Class Question: What is the optimal k here?
  - T(s) d/dk has no minimum for k>1 and is monotonic, thus $k_{opt}=2$
- Class Question: Can we broadcast faster than in a k-nomial tree?
  - $\mathcal{O}(log(P))$ is asymptotically optimal for s=1!
  - But what about large s?

Very Large Message Broadcast

- Extreme case (P small, s large): simple pipeline
  - Split message into segments of size z
  - Send segments from PE i to PE i+1
- Class Question: What is the runtime?

Very Large Message Broadcast

- Extreme case (P small, s large): simple pipeline
  - Split message into segments of size z
  - Send segments from PE i to PE i+1
- Class Question: What is the runtime?
  - $T(s) = (P-2+s/z)(\alpha + \beta z)$
- Class Question: Compare 2-nomial tree with simple pipeline for $\alpha=10$, $\beta=1$, $P=4$, $s=10^6$, and $z=10^5$

Very Large Message Broadcast

- Extreme case (P small, s large): simple pipeline
  - Split message into segments of size z
  - Send segments from PE i to PE i+1
- Class Question: What is the runtime?
  - $T(s) = (P-2+s/z)(\alpha + \beta z)$
- Class Question: Compare 2-nomial tree with simple pipeline for $\alpha=10$, $\beta=1$, $P=4$, $s=10^6$, and $z=10^5$
  - 2,000,020 vs. 1,200,120

Optimal Segment Size

- Class Question: What is the optimal z for given α, β, P, s?

Optimal Segment Size

- Class Question: What is the optimal z for given α, β, P, s?
  - Derive by z
  - $z_{opt} = \sqrt{\frac{s\alpha}{(P-2)\beta}}$
- Class Question: What is the time for simple pipeline for $\alpha=10$, $\beta=1$, $P=4$, $s=10^6$, and z_{opt}?

Optimal Segment Size

- Class Question: What is the optimal z for given α, β, P, s?
  - Derive by z
  - $z_{opt} = \sqrt{\frac{s\alpha}{(P-2)\beta}}$
- Class Question: What is the time for simple pipeline for $\alpha=10$, $\beta=1$, $P=4$, $s=10^6$, and z_{opt}?
  - 1,008,964

Lower Bounds

- Class Question: What is a simple lower bound on the broadcast time?

Lower Bounds

- Class Question: What is a simple lower bound on the broadcast time?
  - $T_{BC} \geq \min\{\lceil \log_2(P) \rceil \alpha, s\beta\}$
- Class Question: How close are the binomial tree for small messages and the pipeline for large messages?

Lower Bounds

- Class Question: What is a simple lower bound on the broadcast time?
  - $T_{BC} \geq \min\{\lceil \log_2(P) \rceil \alpha, s\beta\}$
- Class Question: How close are the binomial tree for small messages and the pipeline for large messages?
  - Bin. tree is a factor of $\log_2(P)$ slower in bandwidth
  - Pipeline is a factor of $P/\log_2(P)$ slower in latency

Towards an Optimal Algorithm

- Class Question: What can we do for intermediate message sizes?

Towards an Optimal Algorithm

- Class Question: What can we do for intermediate message sizes?
  - Combine pipeline and tree → pipelined tree
- Class Question: What is the runtime of the pipelined tree algorithm?

Towards an Optimal Algorithm

- Class Question: What can we do for intermediate message sizes?
  - Combine pipeline and tree → pipelined tree
- Class Question: What is the runtime of the pipelined tree algorithm?
  - $T = \left(\frac{s}{z} + \lceil \log_2 P \rceil - 1\right) \cdot 2 \cdot (\alpha + z\beta)$
- Class Question: What is the optimal z?

Towards an Optimal Algorithm

- Class Question: What can we do for intermediate message sizes?
  - Combine pipeline and tree → pipelined tree
- Class Question: What is the runtime of the pipelined tree algorithm?
  - $T = \left(\frac{s}{z} + \lceil \log_2 P \rceil - 1\right) \cdot 2 \cdot (\alpha + z\beta)$
- Class Question: What is the optimal z?
  - $z_{opt} = \sqrt{\frac{\alpha s}{\beta(\lceil \log_2 P \rceil - 1)}}$

Towards an Optimal Algorithm

- Class Question: What is the complexity of the pipelined tree with z_{opt} for small s, large P and for large s, constant P?

Towards an Optimal Algorithm

- Class Question: What is the complexity of the pipelined tree with z_{opt} for small s, large P and for large s, constant P?
  - Small messages, large P: s=1; z=1 (z<s), will give O(log P)
  - Large messages, constant P: assume α, β, P constant, will give asymptotically O(sβ)
  - Asymptotically constant for large P and s but bandwidth is off by a factor of 2!

Bandwidth-Optimal Broadcast

- Algorithms exist, e.g., Sanders et al. *"Full Bandwidth Broadcast, Reduction and Scan with Only Two Trees"*. 2007
  - Intuition: in binomial tree, all leaves (P/2!) only receive data and never send → wasted bandwidth
  - Send along two simultaneous binary trees where the leafs of one tree are inner nodes of the other
  - Construction needs to avoid endpoint congestion

SMM vs. DMM trivia

- Class Question: What do you think is the difference of the messaging characteristics between SMM and DMM machines?

SMM vs. DMM trivia

- Class Question: What do you think is the difference of the messaging characteristics between SMM and DMM machines?
  – SMM programming model results in smaller messages (single memory references)
    - High message rate!
  – DMM programming model allows to “pack” messages (larger data)
    - Low(er) message rate!

Open Problems

- Look for optimal parallel algorithms (even in simple models!)
  - And then wait for the more realistic models
  - Useful optimization targets are MPI collective operations
    - Broadcast/Reduce, Scatter/Gather, Alltoall, Allreduce, Allgather, Scan/Exscan
  - Implementations of those (check current MPI libraries ☺)