

## Cyclic Dependencies and Deadlock in Computer Networks (with historical anecdotes)

Greg Thorson (greg@thorsons.org)

### A Small Corner of Parallel Computing History

- My initial brush with cyclic dependencies came in the late 80s when we at Cray Research, Inc (not to be confused with Cray, Inc) were still completely focused on vector computers
- Eugene Brooks III was evangelizing the "Attack of the Killer Micros"
- 2 or 3 (depending on how you count) of us at Cray Research started looking into massively parallel interconnection of microprocessors.
- To add to the flavor, you must understand what heresy this was at a place like Cray Research.
- Disclaimers:
  - The references listed on the last slide are old, but so am I
  - Don't bother looking for more of my writings. I don't typically publish anything unless someone kicks me in the butt.

4/15/2009 Slide 2

### Cray Research, Inc. Y-MP Interconnect: Dedicated resources to and from memory

4/15/2009 Slide 3

### My First Encounter with Deadlock, 1989

- Simple simulated network used for research into massive parallelism
- Used the same network for processor requests and memory replies

4/15/2009 Slide 4

### First Encounter: Node 2 read from 0

- Single stream of references flowed very nicely

4/15/2009 Slide 5

### First Encounter: Node 2 read from 0 and vice-versa. This ground to a halt.

- Two streams in opposite directions quickly locked up and never flowed again

4/15/2009 Slide 6



## Strict Ordering: Examples

- Dimension-Order [1] in a 3-D Mesh
  - Example: network entry  $\langle x \ y \ z \rangle$  < network exit
- Dimension-Order in a 2-D Torus
  - Example: network entry  $\langle x \ vc0 \ \langle x \ vc1 \ \langle y \ vc0 \ \langle y \ vc1 \ \langle$  network exit
- Direction-Order [2] in a 3-D Mesh
  - Example: network entry  $\langle +x \ \langle +y \ \langle +z \ \langle -x \ \langle -y \ \langle -z \ \langle$  network exit
- Turn Model [3]: a more general set of constraints

4/15/2009 Slide 13



## Adaptive Routing [2] [4]

- Adaptive routing allows for turns that would normally be considered illegal
- Rules and resources must be provided to deal with back-pressure on illegal turns.
  - Can make illegal turn if NACK on a separate deterministic and cycle free set of resources when back pressure is encountered.
  - Can make illegal turn if a guaranteed sink for entire message exists on the other end of the link. This allows the message to get out of the way so that it does not create a dependence on the illegal turn
  - Can make illegal turn if a guaranteed cycle free path exists back into a cycle free set of resources (e.g. T3E)

4/15/2009 Slide 14



## Other Dependences: Protocol

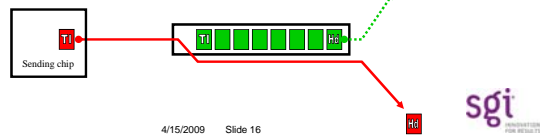
- Protocol: Message 1 is waiting for a message 2 to arrive before proceeding. Message 2 will not arrive because it is blocked behind message 1. For example, let's say you are waiting at a service counter at the store for change, but they have run out of change. If the person who is delivering the new supply of change has to wait in line behind you, there will be no progress.

4/15/2009 Slide 15



## Other Consideration: Arbitration Dependence

- Arbitration: A bad implementation can create an illegal dependence between resources.
- We had a DAMQ implementation that allowed the red packet to start passing because the green was blocked. In the implementation, once one packet started, the other had to wait even if the packet in progress stopped flowing.
- The sending chip did not use all the credits it had been given due to a startup threshold. That is, the red tail would never come until more slots emptied, but that would not happen, because the tail would not come across the link



4/15/2009 Slide 16



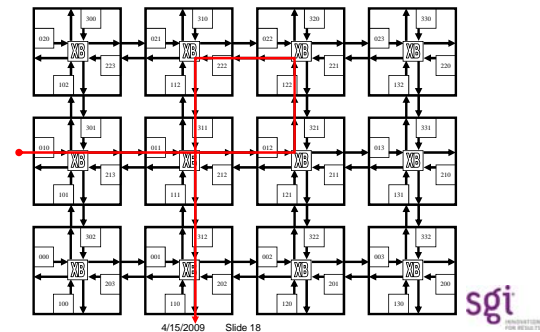
## Other Considerations

- Cyclic dependence can be the sum of many pieces of unrelated traffic that happen to share some of the same resources.
- Once you have the additional resources added to for breaking cycles, you can often balancing their use [5] to improve throughput.
- The average length of dependence chains can be increased by the choice of the cycle avoidance scheme. For example, direction-order is much more flexible than dimension-order routing, but there are added dependences between the + and - directions in each dimension. This can result in longer dependence chains that can impact the efficiency of the network.

4/15/2009 Slide 17



## Extra Long Dependence Chain Enabled by Direction-order Routing (e.g. $+x \ \langle +y \ \langle -x \ \langle -y$ )



4/15/2009 Slide 18



## Important safety tips

- Cyclic dependence can be the sum of many pieces of unrelated traffic that happen to share some of the same resources
- One designer may only implement half of a cycle, another may implement the other half. This may not be found until the two connect their equipment together.
- Never assume the other guy is doing the right thing.
- Never assume the other guy even understands cyclic dependence. For some reason people really have problems with this concept in practice.
- Use formal methods of validation wherever possible

4/15/2009 Slide 19



## References

- [1] Dally, William J. and Seitz, Charles L., "Deadlock Free Message Routing in Multiprocessor Interconnection Networks," IEEE Trans. on Computers, C-36(5):547-553, May, 1987.
- [2] Scott, Steven L. and Thorson, Gregory M. "The Cray T3E Network. Adaptive Routing in a High Performance 3D Torus", Hot Interconnects IV, Stanford University, August 1996.
- [3] C.J. Glass and L.M. Ni, "The Turn Model for Adaptive Routing," Proc. 19th Int'l Symp. Computer Architecture, vol. 20, no. 2, pp. 278-287, May 1992.
- [4] Duato, Jose, "A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks", IEEE Transactions on Parallel and Distributed Systems, v.4 n.12, pp. 1320-1331, December 1993.
- [5] Scott, Steven L. and Thorson, Greg, "Optimized Routing in the Cray T3D Network", Proceedings of the First International Workshop on Parallel Computer Routing and Communication, 1994, pp. 281-294.

4/15/2009 Slide 20

