*** TMC CONFIDENTIAL --- DO NOT DISTRIBUTE ***

Notes on the DR

Steve Heller and Mark Bromley
10 March 92

Distribution:

Mark Bromley
Dick Clayton
Dave Douglas
Carl Feynman
Rolf Fiebrich
Steve Heller
Charles Leiserson
Bradley Kuszmaul
Jon Wade
Shaw Yang

*** TMC CONFIDENTIAL --- DO NOT DISTRIBUTE ***

Cascading Degradation in the Data Router

Steve Heller and Mark Bromley
10 March 92

A group of about a half a dozen people have been studying and discussing an apparent weakness of the current DR implementation called cascading degradation. The bottom line is that the current implementation is not scalable with respect to random communication. There appear to be solutions within the domain of the current abstract DR architecture that involve changing some implementation decisions.

The study group includes: Mark Bromley, Carl Feynman, Steve Heller, Charles Leiserson, Bradley Kuszmaul, Jon Wade, Shaw Yang. We met on Friday, 6 March 92, and this report is a summary of our current understanding of the situation.

Outline:

1. Random and Non-Random Communication and Bisections

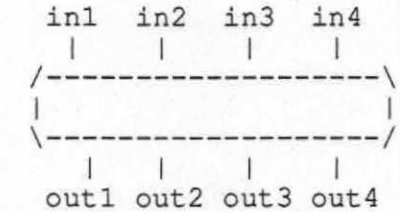
Each node has 40MB/s bandwidth into the router. After two undoubled levels, the share of the bisection owned by each node is 10MB/s. This continues all the way to the top. There are usually two words of header information associated with each packet, and up to four data words, bringing the usable bandwidth (or "utilization") down to 6.7MB/s. On top of that, for random routing there is a hit due to congestion which, if we assume is fixed at 70%, takes us to 4.7MB/s, just under the five we claim.

"Random Communication" does not include grid communication, for example, but does include most other global communication.

As we'll see, we actually achieve the 6.7 number on non-congestive patterns, the "speed of light" for the router. We'll also see that the hit due to congestion can be much worse, and degrades with larger and larger machines.

2. The Theoretical Problem

To a first approximation, each node above the undoubled/doubled boundary can be thought of as a four in and four out switch.



Each input wants a particular output, and if more than one input wants an

output only one gets through. This contention is a function of the message pattern. Even if the utilization in is 100%, the utilization out is in general somewhat degraded. In fact for random patterns the utilization is 1 - (1 - 1/4)^4 = 68%. One level down the utilization in is only 68%, and the utilization out drops from there. In the limit (many levels) we are basically looking for the fixed point of the equation:

u = 1 - (1 - u/4)

which is zero. This says that bigger and bigger networks will have worse and worse utilization for random communication. To get the bandwidth available per node, multiply the utilization by 6.7MB/s.

Taking into account a little more detail than above, the model predicts the following utilizations. The little more detail treats the top node differently, and takes into account the fact that messages "turn around" from below as well as come from the four sources above.

Machine size	Utilization	Bandwidth/node
16 nodes	100%	6.7 MB/s
64 nodes	70%	4.7 MB/s
256 nodes	64%	4.3 MB/s
1K nodes	53%	3.6 MB/s
4K nodes	44%	2.9 MB/s

According to Charles' modeling, utilization falls of as 1/log n.

3. The Problem in Practice

Experiments were run comparing a contention free communication pattern called exchange (or simply X) to a random communication pattern (R). X flips the high bit of the self address, sending data between the low and high processors in a kind of block exchange. X achieves about 6.7 MB/s, ie it is pushing against the hardware bandwidth. The following rates were measured using the LANL machine.

Machine size	X rate	R rate	R/X (measured)	Model (above)
16 nodes	6.65	6.65	100%	100%
32 nodes	6.65	6.55	98%	
64 nodes	6.65	5.02	75%	70%
128 nodes	6.65	4.65	69%	
256 nodes	6.64	3.74	56%	64%
512 nodes	6.63	3.34	50%	
1K nodes	6.60	2.93	44%	53%

The model doesn't match all that well numerically, but it definitely shows the same trend. In fact, reality appears worse than theory for all but the small machines.

4. Possible solutions

Solution 1) Do nothing.

We are only going to build machines so big before the next generation --- live with what we have. There are some non-congestive patterns which get full utilization, including all grid communication. This set must be explored to understand what can be done in a carefully orchestrated communication dance. I/O might be doable in this mode, as might global transposes. If degradation was only 1/4 or 1/3 total, it might not be worth worrying about this right away. For a factor of two, however, it

gains in importance --- this needs additional study. [Note: the software cost is nontrivial.]

Solution 2) Buffers --- respin "drop in" chips

Adding buffers to a DR chip allows messages that contend for outputs and lose to make it out of the inputs. This effectively increases the number of sources the outputs have to choose from the cycle after any congestion.

The algebraic story is completely different; the following equation, which models the situation with buffers, has a positive fixed point.

$u = 1 - b(1 - u/4)^4$

b is represents getting a message from a buffer. A positive fixed point is *very* important: it means that we can only lose so much.

Doing a simple simulation of a single chip with buffers and random input, the story is way different. NB This is not a simulation of the whole router but rather a single chip at different levels, taking into account both input utilization and buffering.

Machine size	Utilization with buffers				
	0	1	2	3	4
64 nodes	69%	75%	79%	81%	84%
256 nodes	66%	73%	78%	81%	83%
1K nodes	56%	63%	70%	73%	76%
4K nodes	49%	59%	66%	70%	74%

[Technical notes: 1) this was a simulation as opposed to a probabilistic model, so the numbers in column 0 are slightly different. 2) There is not as much degradation going from 64 to 256 nodes. This is because 20% of the massages are still turning around at the next to top nodes. This is not true at lower nodes, though.]

THE ABOVE IS AN ABSTRACT MODEL. We expect reality to follow the same trends, but the model is not accurate enough to predict actual numbers.

It is possible to respin the DR chip and add buffers in such a way that the new chips can be "dropped in" to the slots for the old chips, even intermixing new and current chips and still deriving benefit. It may make sense to use the current chips at levels one and two, and the new chips at levels above. Although the percentage of DR chips at level three and above grows as the machine gets bigger, it's less than half for all practical purposes. The column labelled "high %age" corresponds to the fraction of DR chips that would need replacement if we only replaced chips above level two.

	high chips	total chips	high %age
256 nodes	64	256	25%
1K nodes	384	1152	38%
4K nodes	2048	5120	40%
16K nodes	10240	22528	45%

Not only must we understand how many buffers to use, but there is an issue of where to put them. In addition to the shared buffers described above, each output may have several buffers, or we might want a combination of shared and private buffers. This needs additional study.

Respun chips will be able to take advantage of a much newer, denser, and cost effective technology, .8 micron low power versus 1.0 micron high power, and will cost less: Shaw estimates the chip cost will be less than

half the current chip price both now and in the future. Also, a power dissipation concern can be addressed.

Solution 3) Buffers and Clock --- respin chips and pump clock

It may be possible to increase the clock by as much as fifty percent as we respin the chips, but it may affect wires and cables. It may not.

Solution 4) Dilated fat-tree --- respin chips and boards

There is a theory that by changing the topology slightly into a dilated binary fat-tree tree, we can achieve even better throughput. This will involve respinning chips and the DR boards (again, can be just above level two) and will use twice as many DR chips (only at levels replaced) and have twice the latency. The cables should not need alteration.

Solution 5) Multi fat-tree --- respin chips, boards, and wires

There is a theory that a multi fat-tree will perform better than anything else known to man. A multi fat-tree is a serious evolutionary step in the router architecture. New chips, new boards, new wires, ...

5. How Did This Happen?

Two buffers were in the original design, which was simulated in many ways. This original design, however, did not include doubling. Perhaps the reasoning for not simulating the doubling is that one can argue that it only improves the situation.

An engineering decision was made when the DR was cranked out that the buffers would be dropped. Space on the chip was tight, and the schedule was pressing. Including the buffers would have extended the chip's schedule several months. The current chip implementation was deemed the best choice given the understanding of the engineering tradeoffs at that time. It was believed that this would decrease utilization from 83% to 70% for most routing patterns, a 16% decrease --- not bad. Unfortunately, the model was wrong --- cascading effects were not appreciated. In retrospect, this is not a surprising conclusion as cascading effects on the buffered version are not very dramatic, and they are limited in the limit.

An RTL simulation of the final design was done. The main emphasis of this effort was to study the tradeoff between input and output fifo sizes. It was at too low level to study design issues however, like the kind of buffering we discuss here.

High level modeling was done for the final design, but nothing that could be termed architectural simulation. There was no vehicle for attacking the design and exploring its weaknesses, just some checking of things that were felt to be important.

If the modelling were correct and complete enough, we might not have made this mistake. But it was neither correct nor complete enough. The problem of cascading degradation is the *third* major issue we have come across that might have been understood two or three years ago.

The first issue was the injection/ejection policy. Should more emphasis be placed on pushing messages into the network, or pulling them out? This was studied at the model level, and it was concluded that a push heavy strategy was best: push repeatedly until you can't push any more, then pull once. This turned out to be wrong. When we realized that this might be an issue (due to poor performance on some patterns) we changed the code to a fair

strategy (where the opportunities to push and pull are equal) and we saw dramatic improvements in *some* patterns. Now we are exploring various push favored strategies the further improve *some* patterns. Experimenting now requires assembly level programming and is extremely vulnerable to implementation details. These issues could have been and still can be studied at a higher level. This is not a solved problem, especially in light of a possible new NI.

The second issue is a phenomenon whereby inserting barrier synchronizations can improve communication performance significantly in *some* patterns. This is surprising and counterintuitive. There is a theory and some evidence. This phenomenon may be controllable in the current system, and solvable with a new NI. To compound matters, there appears to be an interaction between barrier synchronization and the injection/ejection strategy.

The third major issue (to date) is the topic of this note, cascading degradation.

Not only did we miss these issues (and probably others) early in the design cycle, but they are difficult to study now due to the absence of an architectural simulator. Whatever comes out of the current problem, we all felt that we need to build an architectural simulator to study current problems and future solutions.

6. How Do We Keep It From Happening Again?

We need to build an architectural simulator to study these and future problems as well as design issues for the current and future machines. The simulator need not be built all at once to address the issues at hand. The issues related to building a simulator may be the topic of a separate note and is not explored here.

The importance of modeling should not be minimized. But modeling should not be viewed as a replacement for simulation.

7. Effects on NI/DMA

There is a project underway to respin the NI to facilitate a faster interface to the network. We need to understand the relationship of the current router to any proposed engine, and if we consider changing the DR, we should do it in conjunction with any changes to the NI. As an example, it probably makes sense to expand the input and output fifos to handle larger messages (say 64 bytes, up from the current 20). There are issues of the relationship of the DR and appropriate size of the fifos in light of the NI. Also, both the injection/ejection strategy and the barrier issues are intimately related to the NI design. It would be a shame to miss the opportunity to study and possibly put these issues to rest with the new design.