Chita Das, Craig Stunkel, Topology, Routing Working Group

Das - I handle the NSF computer architecture program at NSF (I am a professor from Penn State)
Stunkel – I am with IBM in Deep Computing. We will begin with slides and then have some brainstorming. We will discuss today's high performance networks and exascale game changers (apps, technology, architecture), along with apparent trends.

- Many networks have been built (and many others have been proposed). We have fat tree/Clos and k-ary n-cubes dominate today – both provide global bandwidth, acceptable performance for traffic patterns, have simple routing rules, fault tolerance and are incrementally stable.
- Exascale application game changers – what have we learned from DOE apps to date? We need to understand the app requirements. For streaming we need high bandwidth but perhaps not low latency or fast switching. We also need unidirectional bandwidth.
- For more accurate physical models will nearest-neighbor communication be sufficient for many new apps?
- The bottom line: how many applications can utilize such systems?
- This morning we heard a plea to let us keep the illusion of a flat, uniform system.

Dally – What I heard is that the illusion gets in the way.
Comment – They want the illusion but it does not work.
Shalf – If you can make the illusion work then we are happy (but if not, do not fool us anymore).

- We heard that point to point is typically bandwidth bound and that collectives are typically latency bound. Both bandwidth and latency will become more challenging as technology scales. We hear about a few very important apps the need FFTs and that strong scaling will become important with flat CPU scaling (and will favor implicit methods).
- We heard that MPI will remain the most important programming model (and that most people don't program in MPI but instead to abstractions/libraries). How to handle heterogeneity? Fault tolerance is an important problem for the whole stack.
- Are there things you did not hear this morning in the apps space that we should identify now?

Comment – More shared memory applications (essentially systems that do not use distributed memory but have a large, flat memory that the processors access).
Stunkel – Coherency is difficult in hardware across a large system. There is also a language question.
Dally – People generally like global address spaces because they give the impression of coherence. Having illusions is a way to guarantee that you will be unhappy.
Comment – MPI will remain the most important programming model - but as multicore systems enlarge that is basically a large shared memory processor (which is not ideal for MPI) and I do not think that MPI will remain the most important programming model.
Heroux – MPI will remain the most important inter-node programming model. Keeping the illusion of a flat, uniform system – the point is that MPI is just as good as using OpenMP. Let us seek to extend the life of an MPI programming application.
Comment – I think there are two modes and both are important (it would be nice to have an interconnect that can do both).
Shalf – If we have a hierarchical interconnect we do not have the resources or the tools to deal with that. If we want global memory then we are stuck with a uniformed, flat view.
Dally – Uniformly flat means uniformly slow.
Winkler – I think the flat view does a lot of damage and I would prefer that we have explicit recognition of exposing the non-uniformity to the app layer. If someone prefers (for whatever valid reason) to ignore that and program against a flat model then they have accepted a performance hit right away. Let us explicitly recognize it.
Heroux – Some apps would like to be able to view the network topology as flat – an MPI process could reach any other MPI process with non-uniform cost.

Comment – There are incremental codes and, perhaps, also disruptive inventions that would allow other things to happen. To the extent that we can separate the short term and the long term and pursue both concurrently I think we can have a less disruptive outcome.
Shalf – These bullets are problems in general (and not necessarily to be solved by topology and routing).
Comment – The same can be said about resilience. The illusion that you have a reliable system can be damaging (so we need to explicitly deal with that, as well).

- What do we expect from the technology (ten years down the road)? Tens of billions of transistors per chip (huge performance, off-chip bandwidth will not increase as fast), integrated optics/CMPS-compatible photonics (will it be cheap?), 3D stacking, phase change memory, proximity comm8unication for high radix routers, power and technology implications.

Comment – For years the government has driven the apps they care about. We see today that things change – we see GPUs pushed by consumers and - although it is not MPI or what we like to see - because it is cheap and available people will try to use it. We need to disconnect from the thinking that we have to continue the past. If we look at commercial apps that are out there with different models of computation and programming and, if they can lead to exascale sooner than our current way, then I think that is a reasonable approach and we should consider changing everything. All of the investment we heard about this morning may not scale to exascale.

- What are the new apps or technologies that will be relevant?

Dally – On the technology question – it is easy to fixate on a cool new technology. When I do the final analysis what matters is the energy per bit for each level of the hierarchy. Power limit on chip is what counts. I do not care whether we are talking about optical, electrical or RF but each technology needs to be graded by picojoule per bit.
Comment – I have worked in this area for some time and the most capable interconnect I ever saw was a superconductivity operating at 20 GB/s (and that was five years ago). The power situation is leading to renewed interest in superconductivity and the industry is being rebuilt. I think superconductivity could be a game changer because of that.
Dally – At what temperature does that operate (and how much energy does it take to get it there)?
Comment – They used to have to cool superconductive devices – there are now efficient freezers that can be plugged into the wall and they do not consume that much energy.
Dally – The laws of thermodynamics say they cannot do better than delta two over t.
Yates – I think we should consider resiliency and reliability issues. We are building today's systems out of commodity parts which have some error detection - but that needs to be increased by an order of magnitude.
Stunkel – I agree.
Shalf – There is a study about silent errors in I/O that is very disturbing. There is no end-to-end detection of errors and they accumulate at a rate of about 1 TB/month. NCSA has a system that is down because they discovered that. Our MPI passing situations do not carry a signature end-to-end.
Comment – Given the amount of data we are talking about, the bit error rate has to go away and that is going to be hard. Silent data corruption is a worry to all of us.

- Exascale architecture game changers (continued) – switch radix growing, many-core chips will require on-chip networks (does this create a hierarchy of networks? How do internal and external networks cooperate and complement each other? Will the on-chip networks be hierarchical?), commodity switches don't natively support torus (ring) routing.

Comment – Beyond the notion of hierarchy of networks on chip and off chip there is the notion of heterogeneity in one or both of the spaces (heterogeneous multicores, heterogeneous networks with multiple networks for different functions with different latencies, for example). We should be thinking what the various implications are of having heterogeneity for these various purposes.
Comment – I think the key question is tied to the programming model. We need to know where we want to go.
Stunkel – We began with discussion of the programming model for that reason.

Comment – And that is why we talked about exposing the heterogeneity.
Dally – You want to build the most efficient communication you can, given a certain budget, and allow people to extract that with different programming models. MPI has actually led to a warping of the communication network. You want to build the best network you can and allow people to extract that. Given a certain power budget you want to allow a certain bandwidth and then connect into an off-chip network that has much lower bandwidth. You want to have people extract what they want.
Comment – I think we have had a philosophy of build first and figure out how to program later. I think we need to start the other way around – go all the way to the device people and ask them to invent something to do what is needed (take the optics guys, as an example). In my opinion we should derive specs from a programming model and then see what can be built.
Shalf – The important lesson from MPI is that the assumption of flatness has enabled portability for several generations of machines. The issue is we have too many choices in programming models right now. The challenge of the programming model is not that one does not exist that understands hierarchical programming networks – the problem is we have not selected one.
Daly – Is there a program underway to down select?
Shalf – I participated in such an effort (to determine what was ultimately subsumed under MPICH).
Dally – That was in 1994-1995. Is it happening again?
Shalf – Our backs are not up against a wall yet (as they were back then). Then we took a toy piece of our code and ported it to various places, and, rapidly, the choices were down selected to MPI. The fact is that we will re-engineer codes - but we need a target. And, some things are inherently non-portable. Our assumption of flatness has enabled some portability at a cost of performance. We need a layer that informs about runtime and topology – it is partly a programming model issue. Also, the Intel people tell me that even if it is a homogeneous system it is going to look heterogeneous (due to non uniformity in clock frequencies, power management and other items).
Comment – In the telecommunication world it was a commercial need – it made sense to use optics for long distances versus electronics. Today there is no need to use optics (yet) – because electronics is working fine and is cost effective. The government should let the market forces define the next direction; money drives everything.
Comment – HPC systems are not easy to program. What is cost effective?
Comment – I am talking only about developing technology for long-haul communication.
Yates – We do not have a parallel programming model – my suggestion is to concentrate on the design requirements of the apps and design your hardware to meet those. Then we will come up with a programming model that works. Consider data flow machines as an example of a machine based on a programming model – there are many footnotes in history (this is not a successful approach).
Comment – The Von Neumann programming model is a successful programming model (I am not talking about languages).
Yates – For sequential programming the Von Neumann model works well (but not for parallel programming).
Stunkel – Turning now to the reality that commodity switches do not natively support torus (ring) routing. The switch element radix is typically growing. Switch elements are getting larger (more I/O is available) – this typically translates to less hops, but it is difficult to grow both port bandwidth and radix on a single chip.
Comment – The issue is fewer hops and power.

- With more I/O you could have more ports or fatter ports (and industry is going to more ports).

Question – As we get higher radix on these nodes and systems with larger node counts are we going to be able to wire these things? (That is, physically construct the machines without errors).
Comment – There are levels of integration.
Dally – With fat trees you may not care which cable you plug in where.

Comment – High radix switches are good - but in a generation or so we may begin to approach a power limit (in terms of what the chip can sustain). Even if optics kick in you will eat a fair amount of power and I am concerned there are limits lurking that could provide a significant constraint.
Comment – There are limits but we are talking a shift in the equilibrium point.
Dally – Every generation of technology has been limited by how much power can be put on the die - but the reality is that we have seen 100X improvement over the past ten years. People are now shipping 10 GB and 20 GB will be available shortly; the energy per bit is also coming down due to more efficient design.
Comment – If you go to high radix switches at low bandwidth you need to think differently.
Stunkel – You have to think differently about the switches and the processor – that is a good point. On routing, what is the necessity of moving from simple oblivious routing to sophisticated routing schemes? What is the best technique, performance-wise? What is the energy involved? How to handle QoS, fault tolerance?

- What are the exascale apps requirements that might occur in this area?

Dally – If you go to a high radix network because of the efficiency you end up with huge load imbalances due to pathological traffic patterns– to get around that you need global adaptivity.
Comment – The reality is that there are trade-offs between performance, energy and fault tolerance. We need to be able to actively trade these things, as we need to.
Stunkel – In some networks the node and the routing element are inseparable (so if you lose the node you lose the routing element). What are the challenges we are going to face? Based on discussion this morning I think the list includes network size, performance requirements, power consumption, reliability, performance-power-reliability tradeoffs, need for better performance monitors and tools, and questions such as is dynamic adaptation possible, what is the role of the network interface? Let us focus our discussion on the Top 10 challenges:

- Parallel programming and programming for resilience (not related to this discussion).
- Power consumption
- Global bandwidth
- How to handle small messages and collectives– handled by the NIC panel?
- Placement/embedding
- How to leverage commodity (or get our ideas into commodity production)? It will be a hard solution for supercomputing if industry does not go for what we need.

Dally – That is a somewhat defeatist attitude.
Comment – It is good to think in terms of what we need - but it is also useful to think about what others need.
Dally – I think that NRE does that. You are talking about changing the NRE number.
Comment – Perhaps this fits under sustainability – it would be helpful to get to a higher volume level in order to make it feasible to manufacture the devices that supercomputing needs.
Shalf – So either you lower the NRE or you share the burden.
Stunkel – There are some factors here that are in our favor.
Yates – The answer to the question about NRE cost for everything we have discussed is high ($15M is not high NRE these days).
Dally – All of these numbers are low. Medium should be $100M; high should be $1B.
Stunkel – Will collective communication become more or less important?
Comment – If we not attend to it then Amdahl's Law says it will be the problem (and, if we do, then everything else will be the problem).
Yates – Collective is important to everything that we do. You also have to be able to do arithmetic reductions.
Dally – Because short messages are so painful people rely on MPI and arithmetic reductions. We found no advantage to wiring this into the network. If you drive the overhead and granularity of messaging down you do not need to have collectives in the network. What you really need are

RMOs so you can send a message over and have it atomically modify something on the receiving node. Barriers are probably the one thing you want to do in the network.
Shalf – I agree that we use collectives because they are more efficient for small messages. However the roundtrip time for "joins" is becoming significant. If we can get a one-cycle response for the processor implementing the join that is OK. Collectives were designed to implement patterns that commonly occur.
Das – We are looking at very large systems – how many nodes?
Comment – 100K
Comment – 1 million
Dally – By 2015 we are talking 500 cores in a socket and 64K sockets.
Das – So we are talking 64K-100K sockets.
Shalf – It depends on how much money is available. The per-socket cost of these machines has not changed dramatically.
Dally – For a typical machine in a data center today you consume the price of the machine in the first 18 months paying the power bill. COTS is a false economy.

END OF WORKING GROUP SESSION.