COMPUTER STRUCTURES FOR PARALLEL PROCESSING

G BELL

Rapporteur: A Petrie

Computer Structures for Parallel Processing

Gordon Bell
Stardent Computer Corporation
Sunnyvale, California

Special Purpose, Massively-Parallel, Multicomputers and SIMD Computers

Over the past 5 years, microprocessors have evolved more rapidly than in any past period using CMOS VLSI and RISC-based technologies. In 1991 microprocessors will be introduced that surpass the speed of mainframes.

Multicomputers or mCs[1], built from microprocessor-based computers interconnected either by fast, low-latency switches or in networks of workstations[2], super-supercomputer performance can be obtained. Similarly, SIMD computers[3] with a single instruction controlling a few thousand, floating point arithmetic chips in parallel also provides super-supercomputer performance. Both structures are inherently scaleable with no single element, except packaging as the bottleneck for expansion. *A Scaleable computer* is designed from a small number of basic components (i.e. computers, processors or processing elements, memories, switches, and cabinets) with no single component bottleneck, such that it can be incrementally expanded over its scaling range and deliver linear incremental performance for a well-defined set of applications.

Exploiting parallelism inherent in computer structures with 100s or 1000s of computers or processing elements is often not easy for even a particular application, let alone a workload comprised of a set of diverse applications. The programming paradigm is either a federated collection of message-passing processes or a Fortran-based, traditional vector processor (in the case of the SIMD structure).

---

[1]E.g. Intel, Meiko (and other Transputer-based companies), NCUBE.
[2]E.g. DEC, HP, IBM, Sun Microsystems
[3]E.g. ACT/ICL, MasPar, Thinking Machines, Wavetracer

Mainline, General Purpose Multiprocessors

Since the mid-60s, the shared memory, multiprocessor (mP) has been the mainline computer for general purpose, cost-effective, non-minimal computing nodes. All of the general purpose computer classes are implemented as multiprocessors: supers[1] for the highest performance; mainframes[2] for historic reasons; minis in order to become mainframes and to have higher performance than is available from micros; multi's[3] (for multiple microprocessors) as the mini replacement or as high availability transaction processing[4]; and PC and Workstation-based micros because it is difficult to build a uni-processor using modern, microprocessors.

Given that multiprocessors are now the standard, mainline computer structure, it is useful to understand: what the impediments have been, why build them versus other kinds of computers, why they're built as the mainline, and why the author bet on them for supercomputing in the mid 90's. In particular, it is important to understand the architecture and design issues that are particular to mPs.

Today's commercial multiprocessors, while non-scaleable, are implemented over a scaling range of an order of magnitude because they all rely on a central switch to interconnect processors and memories. The first scaleable mP was CMU's Cm* (c1975), and today, several groups are developing large multiprocessors that would provide massive parallelism in a general purpose computer as an alternative to the traditional *Cray-formula supercomputer* (fastest clock, multiple, scalar/vector processors interconnected to shared memory via a central, high bandwidth switch).

---

[1] E.g. Cray Computer, Cray Research, NEC
[2] IBM and IBM mainframe-compatible clones
[3] E.g. Arix, Digital, Encore, Sequent, Silicon Graphics
[4] E.g. Stratus

Measuring Performance

Determining performance is a challenge with few agreed upon methods, except specific (and expensive to obtain) customer benchmarks for individual programs and workloads. Peak performance parameters, problem scale, performance and turn-around time for either specific problems or workloads, and cost-effectiveness are often used in supporting *benchmarketing* claims that a computer is super or super-super. By some measure, just about every computer builder has declared that they have *the* supercomputer.

Progress

While the progress in exploiting parallelism with the various computer structures and their compilers for selected applications has been quite encouraging, the actual delivered performance indicates a need for improvement. Existing and proposed computer structures from the above three types will be examined together with the challenge facing machine builders, compiler writers and users.

User training is the single most important component for achieving the potential parallelism that the computers of the 1990s provide.

DISCUSSION

Rapporteur: Ann Petrie

Lecture One

No doubt influenced by the hard time that Mr Bell had given the Connection Machine coupled with the fact that it had been awarded the Gordon Bell prize, Professor Ibbett asked who it was that awarded this prize. Mr Bell replied that a committee administers the prize and that he had said to the committee to do what they liked and to make it flexible.

In view of Mr Bell's interest in scalable computers, Mr McCue asked whether people really did have scalable problems. Mr Bell said that they did and that practically all the problems for the Grand Challenge were scalable. If the grid size of a problem was halved the computation went up by eight.

Mr McCue said that people doing massive computations, such as the weather people, would try and get hold of the biggest computer that they could afford rather than buy a small one and extend it later. Was it sensible to sell scalable computers? Were there people who would start off with a computer costing 10,000 dollars and then upgrade it to one costing a million dollars? Mr Bell said that he was not making value judgements about scalable machines, but the financial constraints of computer manufacturing and the realities of the market meant that a company could not afford to sell a range of machines that differed by more than a scale factor of ten.

Lecture Two

Professor Ibbett asked if the DASH team were intending to perform experiments on the locality of the programs that were run. Mr Bell said that they were doing all sorts of instrumentation so he imagined that this was the case.

Professor Shepherd seemed not to be convinced that the hierarchical design was a good idea. Because it had two different types of interconnect, it was not obvious to him that the DASH machine would be easily programmable or that it would perform well. Mr Bell said that the two levels of interconnect did not affect the programming and that the behaviour of the DASH machine would depend on properties of the problem and on the algorithm being run. As the number of processes being used to solve the problem increased there would be an initial speed up but this would begin to fall as more traffic went over the interconnect rather than the bus.

Professor Shepherd said that the DASH machine was a solution looking for a problem. The team seemed to be saying "look how clever we are building this machine. Now let us see what it can do". Mr Bell replied that the DASH team, like himself, regarded hierarchical design as being the way to produce scalable machines and that they were driven by the desire to build a hierarchical machine.

After Mr Bell had shown some graphs showing how the DASH machine performed on several problems, Professor Ibbett said that, like all curves produced by manufacturers and designers to demonstrate performance, they proclaimed "This is the ideal. Look how well our machine does." The performance is bound to deteriorate as problems get larger since this is inherent in communication taking any length of time. Mr Bell said that all he believed in himself were the Hockney formulae for characterising machine performance. A comment was made that it was not a question of belief but of science.

Professor Shepherd said that there was no need for a hierarchy when you can have more processors on the bus. Why have a machine with 16 processors when you can have one

with 32. Mr Bell replied that too many processors led to a slower machine. Professor Shepherd asked what was the limit to the number of processors that you can have on a bus. Mr Bell said that, with current buses, there was an electrical limit of 16Mhz which allowed 10 taps into the bus. The fastest bus at the moment is the Stardent Bus which operates at 16MHz and can transfer data at a rate of 256 Mbytes/sec. You can put in more processors by having more than one processor to a tap but then you have to deal with an idiosyncratic cache. Professor Shepherd said that you don't change to a design like DASH until after you had exhausted the possibilities using a single bus. Mr Bell agreed, but the question was how did you do it?

Professor Randell said that the structure of DASH was similar to that of multiple linked Encores but, whereas Encore used buses at both levels of the hierarchy, DASH by using memory with a directory allows a network interconnect structure to be used in place of one of the buses. Why not use two interconnect structures. Mr Bell replied that the network interconnect introduces latency but that if you put in a crossbar switch you lost scalability. He added that you can get into arguments about scalability. With the DASH directory structure, for example, the size of the directory word has to increase as the number of clusters increases. A lot of work still needs to be done to optimise the performance of this structure.

Professor Shepherd said that the performance of DASH appeared to be determined by the latency of the interconnect and wondered in what way the machine was better than the BBN. Mr Bell replied that the designers made the leap of faith that problems would have locality and that one would be able to do prefetch.

In response to a comment that if one believed in locality then one may as well use a multi-computer, Mr Bell said that a multi-computer doesn't have the property of being able to do an arbitrarily large job mix and that one may not be able to fit each job onto one computer. The big advantage of a multi-processor over a multi-computer is that it can pretend to be a sequential processor with a single address space. This is a great advantage.

Professor Levy asked about the memory consistency of DASH and was told that the memory is weakly consistent as opposed to fully sequentially consistent (Nitzberg and Lo 1991). Dash doesn't synchronise caches after a write.

Mr Colloff suggested that if one was going for locality one could regard memory as being local or remote and for non-local memory one could broadcast asking who has it. Professor Tanenbaum thought that this was a good idea and said that you should never use anything that you are penalised for using. One knows that one has locality in a cluster. The problem is to know what happens when you go outside the cluster.

In reply to Professor Shepherd asking whether he was right in remembering that Mr Bell did not regard Mflops as a useful measure of performance, Professor Levy said that it was he who had said that Mips were not useful. Mr Bell said that the use of Mflops was fine for well defined applications. He himself likes the old metric developed at the National Physical Laboratory - the Whetstone - which he regards as the best metric for scalar work.

Professor Randell asked how Mr Bell thought that the maximum number of processors you could hang on a bus would change as the technology changes. Would it, for example, go down to two in five years. Mr Bell said that there was a strong argument that a better structure to use was a ring as a slotted ring gives more bandwidth.

Professor Atkinson asked whether it was reasonable to consider a single address space as machines got larger. Mr Bell replied that he wasn't planning on switching from having a single address space. Professor Levy said that the next generation of chips will have 64 bits and, consequently, we will not run out of address space.

Mr Colloff said that benchmarks tended to be small and therefore lived and ran in cache. As a result benchmarks don't give a proper idea of how a machine will perform for typical workloads. Caches keep getting bigger - Hewlett Packard, for example, have a 750 Kbyte cache. Mr Bell replied that many problems do cache well. Mr Colloff said that it varies with the application. In one investigation the operating system was found to cache well compared to a database.

Mr Bell mentioned his bet with Danny Hillis (the Bell-Hillis bet) that by the final quarter of 1995 traditional super-computers will deliver more computations (operations per month) that multiple data-stream machines. Professor Randell asked whether "super" would be measured by characteristic speed or by market share. Mr Bell said that the bet referred to the installed base and that the machines considered would sell for over a million dollars and would satisfy the Cray formula.

Professor Tanenbaum said that PCs would deliver more computational power overall. He then asked if Mr Bell thought that a Cray would still beat the fastest supercomputer. Mr Bell said that there would be some computer in 1995 that called itself a Cray but that, no doubt, someone would have lashed together lots of PCs to provide a lot of computing power. Rates measured in Teraflops were expected by 1995.

It was agreed that it would be difficult to decide who won the bet in terms of which machines qualified for inclusion. The Stardent is not a supercomputer and will not qualify. Mr Bell speculated that the CM3 (Connection Machine 3) would be a large scale multi-computer with a single address space and fast interconnect. It will be programmed as SIMD but shared memory will reduce explicit message passing. It will not be able to function like a multi-processor with one operating system and common work queue and this will be the end of this approach - a vector processor with its brains blown out.

Professor Randell said that the CM approach had originated with ICL. Mr Bell replied that no real money had been spent on the ICL machine and one of the nice things about it was that there were not many in existence. He added that ideas like that belonged to Universities who should keep them there so as not to contaminate users.