Powerful Computing: A Way Forward?

J.H. Davenport (editor)
J.H.Davenport@bath.ac.uk

February 22, 2007

Many academics, largely scientists and engineers but also social scientists and management researchers, find that they need more compute power than is provided “on their desk”. If a machine twice as fast is required, then buying one is generally feasible, but this approach rapidly runs into the buffers of price, power consumption, air (or water!) cooling, and sheer availability. The future lies in more, rather than faster, computers, so-called “parallelism” [3]. The good news is that parallelism is getting more affordable [7], but upgrade routes, and associated financing, become ever more critical.

Should the University of Bath be investing in such a parallel facility, probably of the scale indicated in figure 2 (page 5)?

1 Scientific Case

The motivations of the academics can be thought of as two-fold.

Results: where the prime motivator is the answer. One example would be the historical astronomer who wishes to know where the moon was 4000 years ago in order to match a lunar eclipse with Babylonian records, and hence establish a sound reference point for Babylonian chronology.

Methodology: the researcher knows that the family of questions is important, and he wishes to improve our knowledge of ways of answering them, even though he has no “real-world” instance at hand. The University of Bath does not design aeroplanes, but researchers may well wish to model the aircraft’s noise production, with a view to others using these tools to design quieter aircraft. Sometimes the issue is how to parallelise the problem at all, sometimes it is how to parallelise it more efficiently, and sometimes the issue is how to solve the problem at all, since parallelism is clearly necessary.

However, comparatively few academics are motivated by purely one or the other: the motivation is generally some mixture of the above.

Anecdotal evidence says that "what access will I have to powerful computing facilities" is being asked more by potential recruits (academic staff, but also research students) in some subjects.

A wide range of problems can be suitable for parallel computing, with a wide range of characteristics. Some of these are listed here.

Memory. Sometimes minimal memory is required. Sometimes the computation splits into chunks, each of which requires a significant, but not outrageous, amount of memory. For example, we may be interested in matrix-vector multiplications $\mathbf{M} \cdot \mathbf{v}$, where $\mathbf{M}$ is fixed, and we have several different $\mathbf{v}$. If each processor contains a few rows of $\mathbf{M}$, and is given $\mathbf{v}$, then it can produce the corresponding elements of $\mathbf{M} \cdot \mathbf{v}$. Sometimes very large amounts of memory, with essentially random access, are required, to the point where the issue could be described as "memory parallelism" rather than "processing parallelism".

Coupling. The problem must be split into various chunks to run on the different processors. How coupled are these chunks? How often must they speak to each other, how much must they speak to each other, and how critical is any delay? Some problems are almost totally uncoupled, or very loosely coupled: if I have 10,000 haystacks, 10,000 workers, and wish to search for a needle, then I can give each worker a haystack, and the only communication required is when one worker shouts "I have it".

At the other extreme are algorithms that are completely coupled: every step depends on all the previous ones. Here no parallelism is possible, short of re-designing the algorithm. However, even if this is the case at a high level, the steps themselves may be parallelisable. Thus, if we wish to compute $\mathbf{M} \cdot \mathbf{v}$, $(\mathbf{M} \cdot \mathbf{v}) \cdot \mathbf{v}, \ldots$ each step depends on the previous, but the individual steps can be parallelised as described above. Coupling tends to split into two aspects.

Bandwidth This denotes the speed at which data can be transferred between processors in the system. In a grid system, it depends on the network used, and other uses for that network. In a cluster, it depends on the architecture used: on Skein (appendix A.1) it is 2Gb/sec (at most). In both cases, the limits are generally the *total* bandwidth: so on Skein, A can talk to B at 2Gb/sec, or A to B at 1Gb and C to D at 1 Gb, or four pairs at 0.5Gb .... In a special system, fancier mechanisms can be used, and bandwidth figures are often "point-to-point" rather than total.

Latency This describes how long it takes between A deciding to send a datum to B, and B receiving it. An Ethernet packet has a minimum length of 64 words, which is 0.5μs even on 1Gb Ethernet. This is a cable-imposed limit: typically the hardware and software at both ends will impose much more delay: 25μs is the minimum measured by Netpipe[1]. Infiniband [2,

[1] http://www.scl.ameslab.gov/netpipe/.

Infiniband] quotes 2.6μs. In general we have

$$\text{Grid} \gg \text{Cluster} \gg \text{Special}.$$

High latency can be tolerated in some applications [6], and provided by some implementations [9].

Specialist hardware and associated software is generally needed to achieve very low latency. If we plan to go down the more "commodity" route, we should note that "Gigabit Ethernet is still much cheaper than 10-Gb Ethernet. Some cards offer an upgrade path to 10-Gb Ethernet" [4]. Probably we ought to be careful not to lock ourselves too firmly into 1Gb at this stage of evolution.

2 Teaching

> We believe that good university teaching needs the invigorating stimulus of active research, and that the disciplined approach needed to make a new topic teachable can feed back to influence the future course of research. [8]

There is already some teaching of parallelism in the M.Sc. in Modern Applications of Mathematics. What is now the Department of Computer Science used to teach some aspects of parallel and distributed systems, and would like, indeed need, to do so again in the frameworks of the M.Comp. and the new portfolio M.Sc. in Computing recently presented to Executive. This should cover various aspects, with a "compare and contrast" flavour, and the presence of both cycle-stealing and a more tightly coupled system would be a distinct advantage.

With the growth in 'e-Science', other Departments would probably also wish to teach it, or have it available for projects, typically Masters (integrated or postgraduate) level. Course teaching tends inevitably to be more towards the "Methodology" end of the Results/Methodology spectrum presented above, whereas project use might be more varied. There would also be substantial scope in integrated Ph.D.s, Eng.D.s etc. Teaching's needs are slightly different, or, to be more accurate, have a different priority order, as in Figure 1. Common use of a such system might lead to some economies in teaching, though this would be a minor factor.

3 Types of Parallel Systems

For the purposes of argument, we can distinguish three kinds of parallel computer, though the boundaries are inevitably blurred.

Grid This term is used[2] to denote loosely coupled computers that can "collaborate" on given problems. It could be argued that the original ARPAnet

[2] Abused? Hyped??

	Research	**Teaching**
System Architecture	Fastest	Modern
Processor Speed	Fastest	Sufficiently fast
Number of Processors	Greatest	Enough
Reliability	Total Uptime	c/w Deadlines
Scheduling Automation	Preferable	Necessary
Documentation	WARSAF	Necessary

Figure 1: Relative Needs of Teaching and Research
(WARSAF = "What Are Research Students/Assistants For")

was "a grid before its time". The computers in the grid need not be at all similar, in hardware or in software, though a user may choose only to use those that satisfy certain conditions. These systems are suited to very loosely coupled problems, especially those where there is some redundancy. A particular kind of this arrangement is cycle-stealing networks (see appendix B).

Cluster "A computer cluster is a group of loosely coupled computers that work together closely so that in many respects they can be viewed as though they are a single computer. The components of a cluster are commonly, but not always, connected to each other through fast local area networks. Clusters are usually deployed to improve performance and/or availability over that provided by a single computer, while typically being much more cost-effective than single computers of comparable speed or availability."[2, `Computing_cluster`] The computers themselves are often "off-the-shelf" machines (such use is often called "Beowulf" —see appendix D, and the networking is generally a private Ethernet, loosely connected to the wider world. The computers are "similar", in the sense of having the same software and operating system, same architecture, but possibly differences in speed or memory (though such differences are at best irritating). A more modern technology, apparently common in the States[3], is the 'lambda', where such computers are connected by optical fibre.

It appears that there are at least five Beowulf clusters on campus, "belonging" to departments, research groups or small teams. These have been financed out of a mixture of research grants and departmental funds, are looked after by a mixture of technical staff, RAs and research students, and generally have no clear future: some are out of maintenance.

Specialist We use this phrase to denote machines or systems that are built using equipment which is not designed for general-purpose use. The boundary line between this and clusters is vague. Higher-end facilities, such as the National Facilities described in appendix A, are of this nature.

[3] JHD/RAA are trying to find out more. See also `http://www.dolphinics.com/products/hardware/d200.html` for one example.

Option	A	B	C	D	E	F
Cost (£K)	100	100	250	250	500	500
Networking	Gb	I	Gb	I	Gb	I
Nodes	32	24	72	48	164	128
$R_{\max}$ (Tflop)	0.75	0.765	1.685	1.531	3.838	4.085
Cost (£/hour)	25	25	40	40	60	60
Cost (£/CPU hour)	0.19	0.26	0.14	0.21	0.09	0.12

Figure 2: Power/cost

4 What can we buy?

This section focuses on clusters, as we already have the machines for a Condor-like solution. The details are from a quote from Dell, clearly we would need to do a proper OJEC[28]-style tender, and **there is no commitment to use this technology**.

Networking `Gb` means Gigabit Ethernet. It seems that we are not quite at the right time for 10Gb Ethernet at this level. We should probably explore upgrade options, though.

`I` means an Infiniband[4] switch, capable of 30Gb, and lower latency, e g. 2.6μs.

Nodes Each with 2 2.66GHz Woodcrest chips[5] and 4Gb memory.

$R_{\max}$ Linpack benchmark figures, as used in the Top500[14].

Cost is a rough figure assuming 27/7/44 running, depreciation over three years (see appendix C), a systems manager and some allowance for maintenance and running costs.

Out of interest, option F appears to be one quarter of number 27 in the Top500[14].

A Available Facilities

A.1 `skein.bath.ac.uk`

This machine was bought four years ago out of an EPSRC grant, and is now out of warranty/maintenance. When purchased, it would probably have been described as "specialist/cluster" in the terminology of section 3. Today it would probably be described as "cluster". It consists of:

[4] [2, Infiniband]; `www.infinibandta.org/about`

[5] Each Woodcrest chip is dual core, so the number of CPUs is four times the number of nodes. However, this architecture means that each node is like two pairs of identical Siamese twins: the two CPUs on a single Woodcrest are very tightly coupled. Some other manufacturers, AMD in particular, offer true 'quad' chips, i.e. Siamese quadruplets.

30 Sunblade 1000 with 2 × 900[6] MHz processors and 2GB memory, linked by low-latency 2Gb networks;

4 Sunfire V880[7] with 8 × 900 MHz processors and 16GB memory, linked by 1Gb Ethernet (to each other and to the Sunblade complex);

totalling 92 × 900 MHz processors and 124GB memory.

Max Norton writes[8] as follows,

> Skein initially used myrinet[9] to interconnect all nodes. On 15th July 2005 we started using the gigabit for MPI traffic on the Sunfire V880s when the failing myrinet cards could no longer be replaced under warranty. Benchmarks showed no loss in performance once the Sunfires used gigabit and as far as the users were concerned skein hadn't changed, they would not have made the conscious decision to use myrinet over gigabit.
>
> The decision that was usually made was whether to run larger jobs on the Sunfires, utilising shared memory between up to 8 processes/jobs on a node. I think this decision has diluted over time due to the job scheduler, its use of backfill would cause smaller jobs to run first, so the decision now seems to be to use whatever resources allow the quickest job execution.

A.2 HPCx

The main national facility currently is **HPCx**[10]. This has 160 compute nodes, each containing 16 processors, with a peak performance of 6.8 Gflops[11] each. This gives the complex a peak performance of 17.4Tflops[12]. Each node has 32GB of main memory[13], totalling 5TB. Based on Linpack benchmarks, on which it scores 12.9Tflops, HPCx is rated[14] the 43rd most powerful computer in the world.

[6] The same sort of processor of which `amos` has 4.

[7] Part of the proposal to use HEFCE money to replace Skein was that we would use two or more of the 880s to replace existing cpu servers, at least for the next year or so. although they are over four years old they are still faster and newer than Mary and Midge.

[8] `200702201012.44198.M.B.Norton@bath.ac.uk`; JHD's notes.

[9] [2, Myrinet], `http://www/myricom.com/myrinet/overview`.

[10] Prof. Bird was on the Management Board that procured this system, and has been heavily involved with earlier national computers.

[11] Gflop = thousand million floating-point operations per second. A powerful PC would have a *theoretical maximum* performance of around 10 Gflops.

[12] Tflop = million million floating-point operations per second

[13] A reasonably powerful modern PC would have 1 GB, and `amos`, the 'largest' BUCS machine, has 16. A "32-bit" application cannot make use of more than 4GB.

[14] `http://www.top500.org/`.

A.3 HECToR

The new national facility will be HECToR[15], with a revised [5] in-service date of October 2007. It is initially expected to be 60Tflops, moving to 200Tflops within a couple of years[16]. The procurement runs for six years, with various substantial interim “refreshers”. This is a lesson which UoB should learn — a commitment to the facility beyond the expected currency of the initial hardware is required.

B Cycle Scavenging

One way of obtaining parallelism is to utilise a number of otherwise idle PCs/ workstations, without formally “commandeering” them. This technique first rose to prominence in [6], but has since become widespread, either within an institution or more widely, including the SETI project[17][1]. Such jobs are sometimes termed “embarrassimgly parallel” [2].

One of the more common implementations of this approach, generally known as “cycle scavenging”, is CONDOR[18]. Activities at Cardiff are described in `http://www.nesc.ac.uk/esi/events/438/CondorActivities/Condor_Cardiff.pdf`, and Prof. Parker[19] and his group have used the UCL configuration. There would be little technical difficulty in setting up such a configuration to work on the LLC PCs and the BUCS teaching labs. **However**, such a decision might preempt either the SUMS review of BUCS, or a decision to move to ‘thin client’[20] models in the Library for energy reasons. However, once Executive has moved on these matters, a Condor[21] decision can, and should, be made urgently, and without prejudice to any other decisions, since the type of paralleism provided is significantly different from that proposed elsewhere in this paper.

It would also be possible to use otherwise idle workstations on staff desks. This would require a great deal of set-up unless such machines also used a “common boot” system as is used in the Library etc. Moving towards such a system would be a significant decision, but would have substantial savings in terms of configuration maintenance, whether or not the machines were used as a Condor pool[22].

[15] `http://www.epsrc.ac.uk/ResearchFunding/FacilitiesAndServices/HighPerformanceComputing/HECToR/ProjectStatus/ProjectStatusAt1December2006.htm`.

[16] The current (Nov. 2006) record is 128Tflops.

[17] Now claiming to have 247Tflops, though the basis of the calculation is not clear.

[18] Presumably so called because it does fly, and is a remarkable scavenger.

[19] **Steve**: any more details?

[20] Thin clients could still be used in a cycle-stealing mode, but the details of the implementation would inevitably differ.

[21] Or another mechanism: a proper evaluation of the field should be carried out, but Condor does appear to be the “market leader”.

[22] Further investigation would need to be carried out on the implications of Condor-style pooling for Active Directory and home working.

C Finance

Traditionally[23], finance is divided into capital and recurrent, with, for a project of this nature, recurrent being staff costs[24], running costs (electricity and air/ water conditioning) and maintenance (hardware and software).

But, as noted in section A.3, it is necessary to commit to the existence of a facility beyond the time at which the equipment originally purchased could be regarded as "state-of-the-art". Indeed, it would probably be intellectually dishonest to engage a research student based on such a facility unless such a commitment was in place. Hence the vision has to be changed.

A better vision is of "initial capital" I, being enough to purchase the machine at Y0. Recurrent should have added to it a (substantial) depreciation amount, earmarked for the replacement/upgrade[25] of the machine, enough to write off I over three years[26]. After two years, enough should have accumulated for a substantial upgrade, and again after four. After six years, there should be (about) enough for a replacement. Other scenarios, such as complete replacement after four years, could also be supported within this general framework. More detailed costing needs to be done nearer the decision point.

Advice needs to be taken from Finance on the options possible and rules to be followed, especially if we want to define this as a "special facility" in terms of FEC[27]. UoB currently has two such facilities, both in Physics, and the system seems to be working well. Apparently there are no rigid definitions of what constitutes an FEC "special facility', and an explicit and expensive parallel computing operation should probably be constituted as one from the start.

On the purchasing front, the advice from Supplies is that we are certainly within the scope of OJEC[28]. It is possible, and strongly recommended, that we go for a two-stage tender, i.e.:

1. Issue call for proposals in OJEC (this call would probably include a requirement to quote existing benchmarks, either the Linpack[14] or the HPC Challenge[29]);

2. Receive proposals;

3. Issue tender documents to the shortlist (about 3);

   Almost certainly invite the short-list to present/discuss;

[23] This view has been the explicit view of research councils in the past, colouring the academics' mentality, and very largely the practice of funding councils, with earmarked "capital funding".

[24] For a system manager and any other associated people, not the RAs etc. doing the actual research

[25] Here is not the place to discuss the Ship of Delos paradox.

[26] Ideally two, but JHD thinks the accountants would have problems with this, whatever the scientific validity.

[27] Full Economic Costing.

[28] Official Journal of the European Community.

[29] Used as part of the HECToR procurmenet: see `http://www.epsrc.ac.uk/ResearchFunding/FacilitiesAndServices/HighPerformanceComputing/HECToR/HECToRBenchmarkCodes.htm` or `http://icl.cs.utk.edu/hpcc`.

4. Receive final tender documents and decide.

We must[30] leave 30 *calendar* days between 1 and 2, and 35 between 3 and 4. This adds up to at least $2\frac{1}{2}$ months, more in practice.

D Beowulf

> Beowulf is a design for high-performance parallel computing clusters on inexpensive personal computer hardware. Originally developed by Thomas L. Sterling and Donald Becker at NASA, Beowulf systems are now deployed worldwide, chiefly in support of scientific computing.
>
> A Beowulf cluster is a group of usually identical PC computers running a Free and Open Source Software (FOSS) Unix-like operating system, such as Linux or BSD. They are networked into a small TCP/IP LAN, and have libraries and programs installed which allow processing to be shared among them.
>
> There is no particular piece of software that defines a cluster as a Beowulf. Commonly used parallel processing libraries include MPI (Message Passing Interface) and PVM (Parallel Virtual Machine). Both of these permit the programmer to divide a task among a group of networked computers, and recollect the results of processing. It is a common misconception that any software will run faster on a Beowulf. The software must be re-written to take advantage of the cluster, and specifically have multiple non-dependent parallel computations involved in its execution.

[2, `Beowulf`]

E Questionnaire circulated

```
Questionnaire to help the Powerful Computing Working Group recommend a
campus Strategy. Open Meeting at 12.15 7 March 2007 in BICS Seminar
Room (1West 3.6) to discuss this, but please fill in even if you can't
come. Return to J.H.Davenport@bath.ac.uk. One questionnaire per PI, please.



1) Assuming the University buys ONE of the options listed in Figure 2
   (page 5 of http://staff.bath.ac.uk/masjhd/Powerful.pdf), how many
   thousand FUNDED (EPSRC etc.) CPU hours/year do you think you would bid
   for to the funder?   ....,
   (100-250 pounds per 1000 CPUh)
```

[30] This assumes electronic publication. Otherwise the figures are 35 and 40. We cannot ‘mix-n-match’ the two.

If the choice matters, please give the figures for each scenario.

A	B	C	D	E	F
...	...	...	...	...	...

If your usage would be radically uneven across a grant, please still quote the average, and give details below.

ASSUME your grant is funded: we will multiply the total by the University's hit ratio.

2) How many CPUs would you want to use in a typical production job? ......

3) On a scale of 0-5, how important is fast interconnect to you: ... ?
(0 = carrier pigeons would do, 1= Gigabit Ether is fine, 5=need Infininet)

4) Do you have requirements for significant amount of disc traffic DURING the computation? If so, please give details.

5) Any other comments?

Name(s): ....
Funder: .... (principal one only)

Queries to J.H.Davenport (x6181)

References

[1] D.P. Anderson et al. SETI@home: An experiment in public-resource computing. *Comm. ACM*, 45:56–61, 2002.

[2] Anonymous. Wikipedia. http://www.wikipedia.org, 2007.

[3] G. Cooperman. Parallel Computing is Required if Moore's Law Continues. http://www.ccs.neu.edu/home/gene/research.html#moore, 2005.

[4] G. Cooperman. Miscellaneous. *Private Communication*, 2007.

[5] M.C. Dewar. HECToR. *Private Communication*, 2007.

[6] A.K. Lenstra and M.S. Manasse. Factoring by Electronic Mail. In J.-J. Quisquater and J. Vandewalle, editors, *Proceedings EUROCRYPT '89 Springer Lecture Notes in Computer Science vol. 434*, pages 355–371, 1990.

[7] M. Ripeanu. A note on the Zipf distribution of Top500 supercomuters, and why Grid computing has the wind in its sails. http://www.ece.ubc.ca/ matei/PAPERS/zipf-argument.pdf, 2007.

[8] University of Bath. Mission Statement. http://www.bath.ac.uk/internal/staff/intro/mission.html, 2003.

[9] D Waitzman. Standard for the transmission of IP datagrams on avian carriers. *ftp://ftp.rfc-editor.org/in-notes/rfc1149.txt*, 1990.