CLUSTER COMPUTING: WHEN MANY HANDS MAKE LIGHT WORK

ŚAUNAK SEN[1]
UNIVERSITY OF CALIFORNIA SAN FRANCISCO
sen@biostat.ucsf.edu

Introduction Many computations in biomedical research such as simulations, bootstrapping, database searches (such as BLAST), and many Monte Carlo algorithms are embarassingly parallel. This means that the computation can be split up into smaller computations; each of those calculations can be performed in parallel threads that do not need to interact with each other. Computations with this feature can be easily distributed, (that is, run on different computer processors), with a gain in speed that is approximately proportional to the number of processors. In this note we introduce some of the concepts behind distributed computing, examples where they have been used, and lay out scenarios where they may be useful for biomedical researchers in the future.

The basic idea The idea of distributing work to gain time has been around for ages. History is replete with such examples – just think of the construction of the pyramids, or cotton picking in plantations, and battalions rushing into battle – you get the idea. In nature, the idea has been around even longer – a great example is the beehive. Many modern scientific techniquies such as microarrays, and the 96-well plate also distribute jobs to gain speed. In the early twentieth century when a computer was a person, who computed, computing jobs were distributed, for example, in tabulating census data. The idea of cluster computing has gained prominence lately with the proliferation of low-cost computer processors. It is now recognized that for some problems it is cheaper to distribute the work to many processors, than to build a single customized supercomputer to do the same job.

A typical cluster A typical distributed computing cluster will consist of several computers, called "nodes". One of the nodes is designated as the "head"[2] node. This is a computer that is connected to every other computer in the cluster. The rest of the computers are called "compute"[3] nodes. Typically, only the head node is connected to the internet and the outside world. The compute nodes are accessible only through the head node. The user submits a job to the head node, which places it in a "queue". Depending on the priority of the job, the job is "scheduled" to be run on a compute node that is free. Sometimes, each job may be split up into many smaller jobs and then distributed to the compute nodes. The compute jobs report back to the head node when the jobs are done. The results are collated by the head node into a form that the user can use.

Typically the head node will have more disc space, and more RAM, than the compute nodes. The compute nodes may run different operating systems (some Windows, some Mac, some Unix), but it is generally convenient to have all compute nodes to be as homogeneous as possible. The network connections between the nodes have to be as fast (high throughput). All nodes except the head node will be "headless" that is, without a monitor or other input/output devices like a keyboard, mouse, or printer. Each compute node is a stripped-down version of a workstation or server class computer (they will not have wirelesss cards, video or sound cards, or even a USB port). They are likely to have "hot-swappable" hard drives (which can be replaced while the computer is running). The cluster will be on power backup, should have the ability to power itself down in the case of an outage, and should have a staggered power on schedule (so as not to cause a power spike when booting). Finally, the computer will be in a cool, cool room. More about this later.

[1] Lecture notes for Patient Oriented Research Talks (PORTS), May 6, 2005.
[2] also "mother" and "master"
[3] also "daughter" and "slave"

A few numbers The unit of computing power is the FLOPS, which stands for Floating point operations per second.[4] The fastest computer is IBM's Blue Gene/L whose capacity is 135 teraflops.[5] That's 10^{12} flops, which is a trillion flops. Google's search engine system has been estimated[6] to be between 126 and 316 teraflops (using Google's filing with the Securities and Exchange Commission, and a paper[7] in the journal IEEE Micro published by the Institute of Electrical and Electronics Engineers, IEEE). The distributed computing project Folding@Home[8] claimed to have a processing power of 187 teraflops on April 13, 2005.[9] By comparison, my single processor 1.8 GHz Mac G5 computer has a rating of 7.2 gigaflops.

A FEW EXAMPLES

Below we give a few examples of how some computer-intensive applications can be speeded by using distributed computing clusters.

Google The best example of successful use of clusters is probably the Google search engine. However, they are not alone (and certainly not the first) to use computer clusters in the internet economy. Most large websites and Internet Service Providers (ISPs) use computer clusters to serve web pages or to route email. At Google distribution is used at several levels. Google has several clusters located in places around the globe. When a search request arrives, it is routed to a cluster closest to the origin of the query. Next, the query is matched against an index constructed from Google's web crawlers. The results returned from the index search are passed to the document servers which give the matched documents. The process of searching the index, as well as that of returning the documents can be distributed with a consequent gain in speed.

BLAST The most-used post-genomic computational tool is BLAST (Basic Local Alignment Search Tool)[10] It is used to find if a selected neucleotide or amino acid sequence (the "query") matches the sequences in a "target" database of a genome, or genomes, database of sequences such as GenBank, or database of proteins. Given a query sequence, BLAST will break it up into smaller words. For each of these words, it will calculate a similarity score with each sequence in the database. Target sequences scoring above a specified threshold are selected. The similarity between the words in the query sequence words and the target sequences in the databases selected are then combined to arrive at a set of sequences in the database that have "matched".

Folding@Home A number of human diseases such as cystic fibrosis, and Mad Cow disease can be traced to how proteins misfold. How proteins take their three-dimensional shape starting from a one-dimensional chain of amino acids is an open problem. Some scientists have taken to directly modeling the physical interactions between the molecules that make up a protein at a fine temporal scale up to the nanosecond. This necessitates a huge computational load. A trick[11] that uses the fact that there is little "memory" or long term dependence between rare events makes a parallelization possible. This was used by the Folding@Home group lead by Vijay Pande to simulate the dynamics of proteins using distributed computing systems. They approached computer users around the world to donate their idle CPU cycles by downloading a "screensaver" which is then used to perform the distributed computation.

[4] A floating-point number is a digital representation for an arbitray (real) number on a computer. A floating point operation would be something like addition or subtraction of two numbers.

[5] As of 24 March, 2005; see http://www.top500.org for a list of the top 500 computers.

[6] http://www.tnl.net/blog/entry/How_many_Google_machines

[7] " Web Search for a Planet: The Google Cluster Architecture" by Luiz André Barroso, Jeffrey Dean, Urs H olzle, http://www.computer.org/micro/mi2003/m2022.pdf

[8] http://folding.stanford.edu

[9] For current stats visit http://vspx27.stanford.edu/cgi-bin/main.py?qtype=osstats.

[10] Altschul SF et.al. (1990) Basic local alignment search tool. *J.Mol.Biol.* 215:403-410. See also http://www.ncbi.nlm.nih.gov/blast.

[11] Voter AF (1998) Parallel replica method for dynamics of infrequent events. *Physical Review B* 57:4983-4987.

BIOMEDICAL COMPUTING

The preceeding examples, interesting as they are, are probably not directly relevant to the research of most biomedical researchers. The following examples, drawn from a statistical angle may come a little closer.

Database searches Electronic medical records are becoming more and more common. The VA's administrative and clinical databases are being increasingly used in medical research.[12] If one is assembling a cohort of all individuals who were tested for creatnine in 1998, it would be enough to search separately by each VA site, and then collate the results.

Simulation Whenever a new statistical procedure is invented, it is now standard practice to verify its properties using simulated data. For example, if we wanted to see if the t-test gives us the right Type I error, we would simulate data from two Gaussian distributions with the same mean, say 10,000 times. For each simulation, we will apply the t-test to the data, and record the result. After all the simulations have finished, we can compare the distribution of the t-statistic to that of the t-distribution. The process of simulating the Gaussian distributions, and calculating the t-statistic in each case can be easily distributed.

Computer-intensive statistical methods As the complexity of biomedical data grows, the associated statistical analyses have grown increasingly complex. There are broadly three type of challenges that come from (a) the size of the data, (b) the distributional patterns of the data, (c) the data structures and associated models. Both major "religions" in statistics – "Frequentist" and "Bayesian" – have responded to the challenge that the price of increasing computational complexity.

Frequentist statistical methods such as random forests, bootstrap, and cross-validation are becoming increasingly popular because of their ability to deal with the complexities outlined above. Their main strength is that they do not make too many distributional assumptions about the data. The strength of Bayesian statistical methods such as Markov chain Monte Carlo (MCMC) methods and multiple imputation lies is their ability to incorporate existing scientific knowledge in modeling, and missing data.

Both classes of statistical methods employ repetitive calculations that can be easily distributed. Bootstrapping and cross-validation involve analyzing multiple resampled copies of the original dataset that can be done independently of each other. MCMC is sequential by nature, but is typically, a few independent runs are required to ensure convergence. As a result, MCMC can also be easily distributed.

PRACTICAL CONSIDERATIONS

Although distribute computing seems quite attractive, there are some practical issues that need to be considered before such a undertaking becomes feasible.

Amdahl's Law Named after Gene Amdahl, Amdahl's Law states that if f is the fraction of a calculation that is sequential (such as reading and writing the data), then the maximum speedup that can be obtained with n processors is

$$\frac{1}{f + (1-f)/n} \to \frac{1}{f}, \quad \text{as} \quad n \uparrow \infty.$$

For example, if 10% of a job is sequential, then the maximum speedup possible with an infinite number of processors is 10. Note that the price-performance ratio gets unfavorable pretty quickly. For a typical data analysis job, reading in the data on the head node, making the decision to distribute the job, and collating the results from the compute nodes cannot be parallelized. Also, there may be network limitations to distributing the job because of insufficient bandwidth (traffic jam of information to and from the head node).

[12] See some papers from our own group such as O'Hare A et.al (2005)

Fault tolerance The cluster has to be tolerant of faults. If one computer goes down, then it should not bring down the entire system. The two components most likely to fail are disc drives and power supplies. If a cluster has n nodes, and each has a p probability of failure in a month (which is ver small), then the probability of at least one node failing is approximately

$$1 - \exp(-np).$$

So, if p is 0.1% and we have 100 nodes, then the probability of at least one node failing is approimately 9.5%, which is non-negligible. For a company like Google with 15000 nodes and a failure rate of 0.01% per month, the monthly failure probability of the overall cluster would be a whopping 78%!

Power and cooling The power and cooling requirements of clusters can be formidable. By the law of conservation of energy, all the electricy coming into the computers has go out somehow. It does so as heat. Assume you have a dual processor Itanium CPU computer which is rated at 130W each. This means that your computer including the disc drive and RAM will run at about 300W. With a hundred such computers we are looking at 30KW for the cluster. That is about 40 horsepower, which works to about 100,000 BTU per hour.[13]

OUTLOOK

The vast majority of biomedical information storage, retreival, and analysis can be performed on standard workstation-class desktops and laptops. However, certain applications involving large datasets (such as administrative datasets, GenBank, etc.) or involving complex modeling (such as protein folding, microarray data, structural equation models) may benefit from high-performance computing. Such applications are likely to be increasingly common as medical recordkeeping becomes automated, data structures get more complex, and high-througput assays become ubiquitous. Research groups anticipating tackling research questions that use such resources may benefit from planning to install and use computer clusters.[14]

[13] A household airconditioner for a 150 square feet room is rated at about 5,000 BTU.

[14] A number of research groups at UCSf have computer clusters. The Division of Biostatistics, UCSF has recently installed a 6-node dual-CPU cluster using Apple's Xserve hardware. Research groups interested in learning more about the cluster or in using it may contact the author. For more information seen http://www.biostat.ucsf.edu/sen/cluster.