Cloud computing and the DNA data race

Michael C Schatz, Ben Langmead & Steven L Salzberg

Given the accumulation of DNA sequence data sets at ever-faster rates, what are the key factors you should consider when using distributed and multicore computing systems for analysis?

In the race between DNA sequencing throughput and computer speed, sequencing is winning by a mile. Sequencing throughput has recently been improving at a rate of about fivefold per year[1], whereas computer performance generally follows 'Moore's Law', doubling only every 18 or 24 months[2]. As this gap widens, the question of how to design higher-throughput analysis pipelines becomes crucial. If analysis throughput does not turn the corner, research projects will continually stall until analyses catch up.

How do we close the gap? One option is to invent algorithms that make better use of a fixed amount of computing power. Unfortunately, algorithmic breakthroughs of this kind, like scientific breakthroughs, are difficult to plan or foresee. A more practical option is to develop methods that make better use of multiple computers and processors, whose most recent manifestation is 'cloud computing'.

Parallel computing

When many computer processors work together in parallel, a software program can often finish in significantly less time. Such types of parallel computing have existed for decades in various forms[3–5]. Cloud computing is a model in which users access computational resources from a vendor over the Internet[1], such as from the commercial Amazon Elastic Compute Cloud (http://aws.amazon.com/ec2/) or the academic US Department of Energy Magellan Cloud (http://magellan.alcf.anl.gov/). The user can then apply the computers to any task, such as serving websites—or even running computationally intensive parallel bioinformatics pipelines. Vendors benefit from vast economies of scale[6], allowing them to set fees that are competitive with what users would otherwise have spent building an equivalent facility and potentially saving all the ongoing costs incurred by a facility that consumes space, electricity, cooling and staff support. Finally, because the pool of resources available 'in the cloud' is so large, customers have substantial leeway to elastically grow and shrink their allocations.

Table 1 Bioinformatics cloud resources

Applications	
CloudBLAST[24]	Scalable BLAST in the cloud (http://www.acis.ufl.edu/~ammatsun/mediawiki-1.4.5/index.php/CloudBLAST_Project)
CloudBurst[13]	Highly sensitive short-read mapping (http://cloudburst-bio.sf.net)
Cloud RSD[19]	Reciprocal smallest distance ortholog detection (http://roundup.hms.harvard.edu)
Contrail	*De novo* assembly of large genomes (http://contrail-bio.sf.net)
Crossbow[16]	Alignment and SNP genotyping (http://bowtie-bio.sf.net/crossbow/)
Myrna (B.L., K. Hansen and J. Leek, unpublished data)	Differential expression analysis of mRNA-seq (http://bowtie-bio.sf.net/myrna/)
Quake (D.R. Kelley, M.C.S. and S.L.S., unpublished data)	Quality guided correction of short reads (http://github.com/davek44/error_correction/)
Analysis environments and data sets	
AWS Public Data	Cloud copies of Ensembl, GenBank, 1000 Genomes and other data (http://aws.amazon.com/publicdatasets/)
CLoVR	Genome and metagenome annotation and analysis (http://clover.igs.umaryland.edu)
Cloud BioLinux	Genome assembly and alignment (http://www.cloudbiolinux.com/)
Galaxy[20]	Platform for interactive large-scale genome analysis (http://galaxy.psu.edu)

Cloud computing is not a panacea: it poses problems for developers and users of cloud software, requires large data transfers over precious low-bandwidth Internet uplinks, raises new privacy and security issues and is an inefficient solution for some types of problems. On balance, though, cloud computing is an increasingly valuable tool for processing large data sets, and it is already used by the US federal government (https://apps.gov/), pharmaceutical[7] and Internet companies[8], as well as scientific labs[9] and bioinformatics services (http://dnanexus.com/, http://www.spiralgenetics.com/). Furthermore, several bioinformatics applications and resources

Michael C. Schatz and Steven L. Salzberg are at the Center for Bioinformatics and Computational Biology, University of Maryland, College Park, Maryland, USA; Ben Langmead is at the Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, Baltimore, Maryland, USA. e-mail: mschatz@umiacs.umd.edu

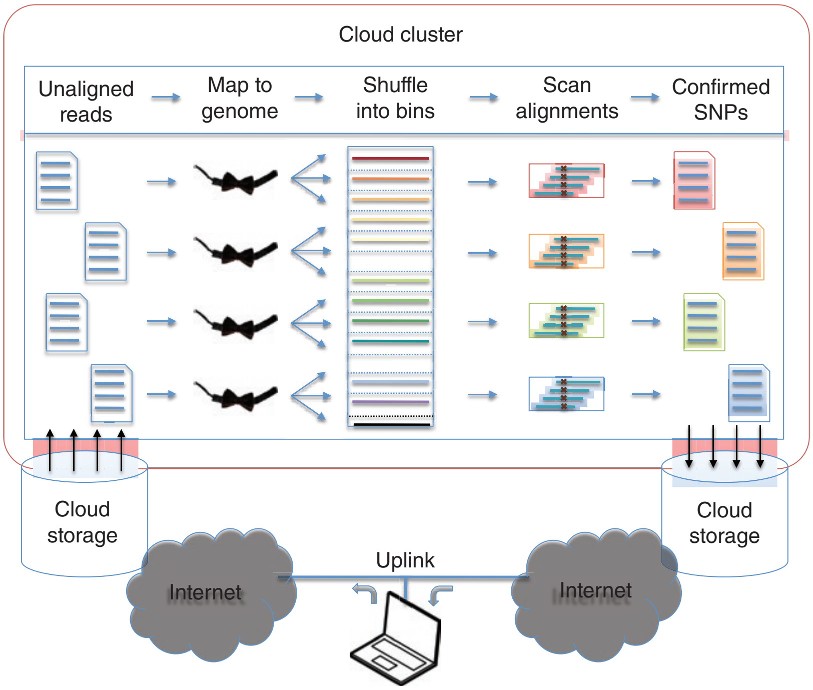

Figure 1 Map-shuffle-scan framework used by Crossbow. Users begin by uploading sequencing reads into the cloud storage. Hadoop, running on a cluster of virtual machines in the cloud, then maps the unaligned reads to the reference genome using many parallel instances of Bowtie. Next, Hadoop automatically shuffles the alignments into sorted bins determined by chromosome region. Finally, many parallel instances of SOAPsnp scan the sorted alignments in each bin. The final output is a stream of SNP calls stored within the cloud that can be downloaded back to the user's local computer.

have been developed specifically to address the challenges of working with the very large volumes of data generated by second-generation sequencing technology (**Table 1**).

MapReduce and genomics

Parallel programs run atop a parallel 'framework', or collection of auxiliary software code, to enable efficient, fault-tolerant parallel computation without making the software developer's job too difficult. The Message Passing Interface framework[3], for example, gives a programmer ample power to craft parallel programs, but it requires relatively complicated software development. Batch processing systems, such as Condor[4], are very effective for running many independent computations in parallel but are not expressive enough for more complicated parallel algorithms. In between, the MapReduce framework[10] is efficient for many (although not all) programs. It makes programming simpler by automatically handling duties, such as job scheduling, fault tolerance and distributed aggregation.

MapReduce was originally developed at Google (Mountain View, CA, USA) to streamline analyses of very large collections of web pages. Google's implementation is proprietary, but Hadoop (http://hadoop.apache.org/) is a popular open-source implementation of the MapReduce framework that is maintained by the Apache Software Foundation. Programs based on Hadoop or MapReduce comprise a series of parallel computational steps (Map and Reduce), interspersed with aggregation steps (Shuffle). Despite its simplicity, MapReduce has been successfully applied to many large-scale analyses within and outside of DNA sequence analysis[11–15].

In a genomics context, MapReduce is particularly well suited for common 'map-shuffle-scan' pipelines (**Fig. 1**) that use the following paradigm:

1. Map: many sequencing reads are mapped to the reference genome in parallel on multiple machines.

2. Shuffle: the sequence alignments are aggregated so that all alignments on the same chromosome or locus are grouped together and sorted by position.

3. Scan: the sorted alignments are scanned to identify biological events, such as polymorphisms or differential expression within each region.

For example, the Crossbow[16] genotyping program leverages the Hadoop implementation of MapReduce to launch many copies of the short-read aligner Bowtie[17] in parallel. After Bowtie has aligned the reads (which may number in the billions for a human resequencing project) to the reference genome, Hadoop automatically sorts and aggregates the alignments by chromosomal region. It then launches many parallel instances of the Bayesian single-nucleotide polymorphism (SNP) caller SOAPsnp[18] to accurately call SNPs from the alignments. In our benchmark test on the Amazon (Seattle) cloud, Crossbow genotyped a human sample comprising 2.7 billion reads in ~4 h, including the time required for uploading the raw data, for a total cost of $85 (ref. 16).

Programs with abundant parallelism tend to scale well to larger clusters; that is, increasing the number of processors proportionally decreases the running time, less any additional overhead or nonparallel components. Several comparative genomics pipelines have been shown to scale well using Hadoop (B.L., K. Hansen & J. Leek, unpublished data; refs. 13,16,19), but not all genomics software is likely to follow suit. Hadoop, and cloud computing in general, tends to reward 'loosely coupled' programs where processors work independently for long periods and rarely coordinate with each other. But some algorithms are inherently 'tightly coupled', requiring substantial coordination and making them less amenable to cloud computing. That being said, PageRank[14] (Google's algorithm for ranking web pages) and Contrail (a large-scale genome assembler; M.C.S., D.D. Sommer, D.R. Kelley & M. Pop, unpublished data) are examples of relatively tightly coupled algorithms that have successfully been adapted to MapReduce in the cloud.

Cloud computing obstacles

To run a cloud program over a large data set, the input must first be deposited in a cloud resource. Depending on data size and network speed, transfers to and from the cloud can pose a substantial barrier. Some institutions and repositories connect to the Internet via high-speed backbones, such as Internet2 and JANET, but each potential user should assess whether their data-generation schedule is compatible with transfer speeds achievable in practice. A reasonable alternative is to physically ship hard drives to the cloud vendor (http://aws.amazon.com/importexport/).

Another obstacle is usability. The rental process is complicated by technical questions of geographic zones, instance types and which software image the user plans to run. Fortunately, efforts such as the Galaxy project[20] and Amazon's Elastic MapReduce service (http://aws.amazon.com/elasticmapreduce/) enhance usability by allowing customers to launch and manage resources and analyses through a point-and-click web interface.

Data security and privacy are also concerns. Whether storing and processing data in the cloud is more or less secure than doing so locally is a complicated question, depending as much on local policy as on cloud policy. That said, regulators and institutional review boards are still adapting to this trend, and local computation is still the safer choice when privacy mandates apply. An important exception is the Health Insurance Portability and Accountability Act (HIPAA); several HIPAA-compliant companies already operate cloud-based services[21].

Finally, cloud computing often requires redesigning applications for parallel frameworks like Hadoop. This takes expertise and time. A mitigating factor is that Hadoop's 'streaming mode' allows existing nonparallel tools to be used as computational steps. For instance, Crossbow uses the noncloud programs Bowtie and SOAPsnp, albeit with some small changes to format intermediate data for the Hadoop framework. New parallel programming frameworks, such as DryadLINQ[22] and Pregel[23], can also help in some cases by providing richer programming abstractions. But for problems where the underlying parallelism is sufficiently complex, researchers may have to develop sophisticated new algorithms.

Recommendations

With biological data sets accumulating at ever-faster rates, it is better to prepare for distributed and multicore computing sooner rather than later. The cloud provides a vast, flexible source of computing power at a competitive cost, potentially allowing researchers to analyze ever-growing sequencing databases while relieving them of the burden of maintaining large computing facilities. However, the cloud requires large, possibly network-clogging data transfers, it can be challenging to use and it isn't suitable for all types of analysis tasks. For any research group considering the use of cloud computing for large-scale DNA sequence analysis, we recommend a few concrete steps.

First, verify that your DNA sequence data will not overwhelm your network connection, taking into account expected upgrades for any sequencing instruments. Second, determine whether cloud computing is compatible with any privacy or security requirements associated with your research. Third, determine whether necessary software tools exist and can run efficiently in a cloud context. Is new software needed, or can existing software be adapted to a parallel framework? Consider the time and expertise required. Fourth, consider cost: what is the total cost of each alternative? And finally, consider the alternative: is it justifiable to build and maintain, or otherwise gain access, to a sufficiently powerful noncloud computing resource?

If these prerequisites are met, then computing in the cloud can be a viable option to keep pace with the enormous data streams produced by the newest DNA sequencing.

ACKNOWLEDGMENTS
The authors were supported in part by US National Science Foundation grant IIS-0844494 and by US National Institutes of Health grant R01-LM006845.

COMPETING FINANCIAL INTERESTS
The authors declare no competing financial interests.

1. Stein, L.D. *Genome Biol.* **11**, 207 (2010).
2. Moore, G.E. *Electronics* **38**, 4–7 (1965).
3. Dongarra, J.J., Otto, S.W., Snir, M. & Walker, D. *Commun. Assoc. Comput. Machinery* **39**, 84–90 (1996).
4. Litzkow, M., Livny, M. & Mutka, M. in *Proceedings of the 8th International Conference of Distributed Computing Systems* 104–111 (IEEE, Washington DC, 1988).
5. Dagum, L. & Menon, R. *IEEE Comput. Sci. Eng.* **5**, 46–55 (1998).
6. Markoff, J. & Hansell, S. Hiding in plain sight, Google seeks more power. *New York Times* <http://www.nytimes.com/2006/06/14/technology/14search.html> (14 June 2006).
7. Foley, J. Eli Lilly on what's next in cloud computing. *Plug Into the Cloud* < http://www.informationweek.com/cloud-computing/blog/archives/2009/01/whats_next_in_t.html> (14 January 2009).
8. Netflix selects Amazon web services to power mission-critical technology infrastructure. *Amazon.com* <http://phx.corporate-ir.net/phoenix.zhtml?c=176060&p=irol-newsArticle&ID=1423977> (7 May 2010).
9. AWS case study: Harvard Medical School. *Amazon Web Services* <http://aws.amazon.com/solutions/case-studies/harvard/>.
10. Jeffrey, D. & Sanjay, G. *Commun. Assoc. Comput. Machinery* **51**, 107–113 (2008).
11. Lin, J. & Dyer, C. *Synthesis Lectures on Human Language Technologies* **3**, 1–177 (2010).
12. Chu, C.-T. *et al. Adv. Neural Inf. Process. Syst.* **19**, 281–288 (2007).
13. Schatz, M.C. *Bioinformatics* **25**, 1363–1369 (2009).
14. Brin, S. & Page, L. *Comput. Netw. ISDN Syst.* 30, 107–117 (1998).
15. Matthews, S.J. & Williams, T.L. *BMC Bioinformatics* **11** Suppl 1, S15 (2010).
16. Langmead, B., Schatz, M.C., Lin, J., Pop, M. & Salzberg, S.L. *Genome Biol.* **10**, R134 (2009).
17. Langmead, B., Trapnell, C., Pop, M. & Salzberg, S.L. *Genome Biol.* **10**, R25 (2009).
18. Li, R. *et al. Genome Res.* **19**, 1124–1132 (2009).
19. Wall, D. *et al. BMC Bioinformatics* **11**, 259 (2010).
20. Giardine, B. *et al. Genome Res.* **15**, 1451–1455 (2005).
21. Anonymous. Creating HIPAA-compliant medical data applications with AWS. *Amazon Web Services* <http://aws.amazon.com/about-aws/whats-new/2009/04/06/whitepaper-hipaa/> (April 2009).
22. Yu, Y. *et al.* DryadLINQ: a system for general-purpose distributed data-parallel computing using a high-level language. *Symposium on Operating System Design and Implementation (OSDI)*, San Diego, California, 8–10 December 2008.
23. Malewicz, G. *et al.* in *PODC 09: Proceedings of the 28th ACM Symposium on Principles of Distributed Computing* 6 (ACM, 2009).
24. Matsunaga, A., Tsugawa, M. & Fortes, J. in *Proceedings of the IEEE Fourth International Conference on eScience,* 222–229 (IEEE, Washington, DC, 2008).