Congratulations! Your paper has been accepted for publication in the Emerging Computational Methods for the Life Sciences Workshop Proceedings of the ACM International Symposium on High Performance Distributed Computing (HPDC 2011).

All camera ready version of papers should be formatted in the same style as HPDC papers, with a minimum of 4 pages and a maximum of 12 pages. Please take the enclosed reviewer comments into account carefully when preparing the camera-ready version. The deadline of final version submssion is March 20, 2011. We will send a separate email including detailed instructions.

The acceptance of your paper is conditional on the agreement that at least one of the authors will attend the conference and presents the paper. It is mandatory that at least one of the authors of your paper register for the conference before or at the time the camera-ready paper is submitted. Please visit the conference website (<http://www.hpdc.org/2011/index.php>) for registration details. Please note that if this requirement is not satisfied, your paper will not be included in the conference proceedings. Furthermore, please note that in case of a no-show at the conference, the associated paper will not be included in the proceedings.

Congratulations once again on having your paper accepted. We look forward to seeing you in San Jose, California June 8-11, 2011.

Regards,

The organizing committee

---------------------------- REVIEW 1 --------------------------

PAPER: 7

TITLE: Characterizing Deep Sequencing Analytics Using BFAST: Towards a   Scalable Distributed Architecture for Next-Generation Sequencing   Data

OVERALL RATING: 0 (borderline paper)

Kim and colleagues explore fine and coarse-grained parallelism of one next-gen sequencer in their DARE framework.  Rather than take a bioinformatics/cloud tack like the Salzberg group at Maryland (and others), they propose their custom framework as a gateway for next-gen sequencing analysis.

Philosophically, this problem is no different from the BLAST problem that was published in a few papers last year in this workshop.  It is also unclear to me the distinction between DARE, SAGA-BigJob, and ADAMS previously reported by these authors in this workshop, mostly because they are flavors of SAGA-BigJobs and the limitations of a "distributed programming environment" discussed in the ADAMS paper is also applicable here.  Mapreduce and other abstractions are used in this community -- and I am not saying always used well -- because they are in some sense easier to program.  My opinion is that the advances in workflows must accompany improvements in infrastructure, but I recognize where they are coming from.  I agree that the nature and amount of parallelism is different between DARE and ADAMS and this is an emerging problem of interest to the life sciences community.  My suggestions for the authors prior to publication are to adequately address the scaling contribution m

ore than the application engineering aspect herein.  Detailed comments follow that outline this logic.  A consequence of knowing this area well is often a longer (and in a sense harsher review) but I find this work interesting just needs more information on the HPC side to fully support their claims in a self contained manuscript.

I guess BFAST was chosen because it is multithreaded and fits into the DARE framework; however,it is one of the least used short read aligners mostly because it is based on a suffix array that is voracious either in memory or disk.  These issues are mentioned in the manuscript indirectly but no thought is given how to scale out a job that would require 200GB of local space.  BWA and the Bowtie, both based on the Burrow-Wheeler transform, are more practical tools to look at for run time needs mostly because they are the more popular (216 and 382 Google scholar citations, respectively, vs. 26). Bowtie, cited here,also has been ported to a cloud environment as mentioned by these authors. The authors should strongly consider comparison to other frameworks (esp. bowtie) for a fair comparison of the practical benefit of their approach.

Relatedly, the paper is very hard to evaluate with respect to its impact on life sciences. To be fair, the authors care more from the HPC/infrastructure side but ultimately the users of this software/gateway will come from outside of the HPC community.  The number one concern for scalability is the size of the index files and the state of the parallel I/O system, but these are not given in Tables 5-7.  It would have been very helpful to connect the examples in Table 3 to the results of the paper.  I will agree with the authors that the size of the query is important and is a key variable in Figures 5-7 but it discounts the practical concerns of parallelizing a tool like BFAST, which constructs a suffix array.  Please clarify what data is which for these tables so biologists can better evaluate the usefulness of the system.

There is also no traditional scaling of the workload here.  This wouldn't be an issue but the motivation of the paper is:

"an existing framework -- which enables the seamless utilization of heterogenous resources" and

When I think of seamless utilization of heterogeneous resources I think of swift, work queue/makeflow, and other lightweight "ad hoc" cloud systems.  The contribution here is that compiling BFAST in the DARE framework allows using different resources, but not at the same time?  It is unclear without reading the previous work why this is an improvement on something like swift or other workflow-enabling lightweight frameworks.

also the authors claim this as a motivation:

""an existing efficient framework upon which to build a scalable infrastructure to support a wide range of abstractions and execution environments.""

When I think efficient, I usually think of the HPC concept of speedup / # of processors or a more loosely defined programability argument.  None of the traditional measures are present in this paper.  For example, in Table 6 the framework appears to scale to at least 80 cores but I have at least three concerns with these data:  1.)  how long did the job take on a serial system, and therefore what is the speedup on the grid/cloud?  2.)  Were the authors able to process a complete run of either the human or Burkholderia? This is unclear from the abstract examples. 4.) Does the grid/cloud version provide speedup over a plain threaded version of BFAST on similar hardware?  5.)  No ability to measure how scalable the software is as there are only two data points on an HPC grid (LONI) that I assume had a shared file system?  The same number of cores were used in Table 7.

I would have liked to see ideally a complete workflow on the grid or cloud or scaling (and ideally both) not a statement related to overhead that, although true, does not support scaling as well as empirical results.

I don't buy the cloud argument in section 3.3.2 because it avoids the fundamental issue in any data intensive task:  data transfer between master and workers (or from a site to Amazon in the comercial setting).  The authors stress a flexible \*\* and scalable \*\* framework and there is no scaling for this on more than 8 cores and no real clear evaluation of how a parallel file system (or lack thereof) fits into this clearly data intensive problem.  There is little support for the statement in the abstract that this work "scales-up and scales out over production grid and cloud environments" as there are only two data points for scaling up data (that are very small relative to real data in Figure 2 depending or not if there was quality information inside) and cores (in the case of the cloud, looking at a few instances).  The real interesting question is the one they reserve for future work: data access mechanisms to the different types of storage, which is where the scaling in our

experience always suffers beyond a few hundred cores.

Overall the paper is written well but could benefit from some streamlining and more results.  Also note that in biological names the second is not capitalized.

---------------------------- REVIEW 2 --------------------------

PAPER: 7

TITLE: Characterizing Deep Sequencing Analytics Using BFAST: Towards a   Scalable Distributed Architecture for Next-Generation Sequencing   Data

OVERALL RATING: 2 (accept)

This paper studied the performance of mapping NGS reads to a reference genome by using BFAST on a scalable distributed computing architecture. The paper carried out several useful experiments and concluded the algorithm may scale well. Overall, the paper provide an interesting case study, although the paper can be improved in a couple of ways. First, the summary of existing efforts on cloud computing of NGS read mapping is not comprehensive. For example, the work by Salzberg and colleagues "Searching for SNPs with cloud computing" Genome Biology 2009 was not mentioned. And CloudBurst is worth comparing with BFAST. Figure 2 should be done on the whole human genome too to compare the scalability.

---------------------------- REVIEW 3 --------------------------

PAPER: 7

TITLE: Characterizing Deep Sequencing Analytics Using BFAST: Towards a   Scalable Distributed Architecture for Next-Generation Sequencing   Data

OVERALL RATING: 3 (strong accept)

How to handle the vast amounts of NGS data efficiently is a hugely important topic, one in which we are in the infancy of

research. This is thus a very timely contribution, and the paper seems to have useful insights into scale-up strategies for

grids and clouds.

---------------------------- REVIEW 4 --------------------------

PAPER: 7

TITLE: Characterizing Deep Sequencing Analytics Using BFAST: Towards a   Scalable Distributed Architecture for Next-Generation Sequencing   Data

OVERALL RATING: 1 (weak accept)

The authors make the point that large genomic datasets require sophisticated computing environments in order to attain acceptable performance, and support this assertion with examples using BFAST. The case is made logically, examining performance characteristics in progressively more complex and distributed environments from a single machine to a cloud. Aside from small textual errors (the authors should define the term CAL before using it, and some explanation of machine names such as Eric and Painter would be useful outside of tables and captions before we encounter thiem in text) I see no faults with the paper, but it amounts to little more than benchmarking with some explanations of the results.