Response to review of

"Understanding MapReduce-based Next-Generation Sequencing Alignment on Distributed Cyberinfrstructure"

ECMLS2012

Author(s): Mantha, Pradeep Kumar; Kim, Nayong; Luckow, Andre; Kim, Joohyun; Jha, Shantenu

The original submission of the paper received three reviews. We have revised our paper based on their comments, and we think the suggestions have indeed improved the manuscript. Our response (in Times) follows along with excerpts from the original reviews (in Courier).

**Reviewer 1 :**

PMR sounds interesting and is a strong motivation of this framework, but unfortunately PMR is under review at another conference and in my opinion should not count as novelty here.

Our PMR paper was accepted and we added it to the reference.

Scale up.  The authors look at 32 mapper processes on what appear to be three configurations: 4 nodes with 4 workers (2 cores per worker), 4 nodes with 8 workers (1 core per worker), 8 nodes with 8 workers.  It was unclear from the legend of Figure 2 how the extra 32 workers help in the third configuration.  Also, the last column in Table 2 does not match the legend as all nodes have 8 cores (and therefore 8 cores are only available with one worker per node?).  Figures 2 and 3 do not have entries in Table 2 (specifically the 16 node job).

As the reviewer misinterpreted, Table 2 is confusing and improved it by describing all experiments properly. This, hopefully, helps the readers to understand right configurations for the experiments.

I don't think the Seqal comparisons are fair in that they use a shared filesystem (and scaling issues therein). It seems a little forced for comparison as there is the bottleneck of I/O for this tool.  On one hand you can view this as an advantage of their tool but I would expect similar "caching" for PMR (but maybe I am wrong) if they have the same number of chunks.

Our comparison is at best the illustration of the comparison of real usages without further expert optimization between Hadoop-based Seqal and PMR. We emphasized again that the comparison does not intend to show the best optimization result with Hadoop-based approaches. However, with the FutureGrid environment, it is true that our PMR provide a way to overcome such I/O bottleneck by adjust data storage and management, whereas Hadoop configuration for Seqal is inflexible to avoid them due to the file system environment of FutureGrid.

Extensibility.  … I can see where the authors are coming from with respect to the advantages of pilot jobs/ad hoc clouds but this is not described/tested in the paper; they only run on a few nodes they obtain from FutureGrid.  Further, I'm not convinced the map and reduce functional paradigm is better than, for example, something based on a straight-up master worker framework like some of the recent bioinformatics work with tools like Makeflow (to name only one).  The authors extend the framework to multiple tools but they still require a shared filesystem, which in practice severely limits "scale across" (see below and a bunch of recent papers)

We added the reference(s) for works of Makeflow. (Joohyun: I wonder we need to respond to his concern on the requirement of the shared Filesystem?)

Comparison of tools.  The Seqal results in Fig4 are clearly biased in that the underlying file system was different.  Figure 5 was more problematic as it was unclear what crossbow they were comparing to.  Ideally, they should have used the latest one that does the same work as their pipeline.  In one part of the manuscript, they refer to it being also run using a shared FS and that the reduce step was not implemented.  With this in mind, what version was here and what does it mean that Crossbow is competitive with PMR for small collections of reads.  I would have much preferred discussion of the results (that differ from the previously submitted PMR framework) rather than an advertisement for PMR in the discussion.

The aims of the comparison to Seqal and Crossbow are i) (Joohyun: we need here to list our goals with the comparisons and want to discuss what his understandings are a bit biased on his view…)

The authors should compare with the emerging GATK framework (although in theory assuming JVMs on SAGA the map-reduce-like framework could run on theirs)

Considering the time available to us, we think this suggestion would be considered in other future works. And, GATK are a bit different from the tools we compared, Seqal and Crossbow, and another historic tool, Cloudburst in several ways including the overall development focuses, and thus we think a more comprehensive work is needed, which is beyond of the scope of this work.

Discussion.  The paper does not discuss how the reference genome is distributed and how/when it is indexed.  This is a concern for smaller data sets and large genomes (although the granularity here of ~1 hour jobs should make this negligible).

We added how the reference genome is distributed and indexed.

The authors should not discuss "scale across" as they only used one platform (FutureGrid) and many of their comparisons relied on a shared file system given limitations of the disks.  This would be fixed by using more cores (32 max is relatively low, especially as "common" servers are approaching 16+ cores).  Also, how data is transferred can be important especially when the clusters are present at different institutions (say with Condor).

(Joohyun Kim, we need some sort of disucssions for scale across and his criticism to use the same type of two clusters instead of the use of multiple platform clusters)

There is also little data on "scale out" (32 cores only on up to 4 nodes?)  For example, in this workshop a year ago there was a paper looking at a purely distributed file system framework based on "makeflow" and "work queue" that ran on up to 300 cores with 195X speedup with a similar map (megablast instead of a BW-based aligner) but a more complicated workflow.  Looking at the makeflow website this has been applied to BWA also and Weaver seems to be proposed as an abstraction similar to PMR (which as I understand is just a map and reduce running on SAGA?.  Their scaling seems heavily based on a hybrid distributed/multicore platform that I guess is a plus, but this puts restrictions on the type of systems usable for good performance.

For showing the scalability with a bigger problem, we added a bigger data set calculation with 512 cores.

In short, promising start -- especially when PMR is published -- but the authors should focus on scale up (more than 32 cores), showing scale across, and performing a better comparison with the state of the art such that current mapreduce tools are clearly run in a comparison (as opposed to edits) and that they use HDFS as intended.  Increasing the granularity such that the files can fit could help, maybe.

(Joohyun Kim, Maybe we need some discussions here?)

**Reviewer 2 :**

The paper is well written, and the authors demonstrate the platform is useful for genomics. However, the paper has 2 substantial flaws that limit its impact: (1) The performance measurements made against a vanilla Hadoop installation (Figures 4 & 5) should be completely redone using proper Hadoop system. (2) The authors should implement the reduce phase of Crossbow to truly measure the capability of distributed PMR. Without the reduce phase, the system is effectively a distributed batch scheduling system and does not demonstrate that distributed shuffle of large datasets is feasible. Until these two flaws are corrected, their major conclusions (PMR is a viable solution for scale-across NGS, and that PMR has advantages over Hadoop) are not proven.

We revised our manuscript to explain why the proper Hadoop system is practically impossible with FutureGrid we used for this work. And, also, we revised to describe why the experiment with the SNP finding step (the third Hadoop streaming step in Crossbow pipeline) is not compared to ours. First of all, using SOAPsnp, as Crossbow is implemented, for PMR is hindered due to the fact that SOAPsnp accepts only the output format from SOAPaligner. Crossbow developers modified their own software Bowtie for producing the formatted output working with SOAP developers, we think. Secondly, we implemented the SNP finding with SAMTools, a main tool used in the 1000 Genome project for small scale genome structure variation discovery. Due to the different tool, we believe the comparison for SNP finding step is not in a right category this paper focuses. In the future work, the comparison with biological consequences will be made, instead.

It would also be useful for the paper to describe a few more details of Pilot and SAGA -- most significantly, are they open source and freely available so that they can be deployed as easily as Hadoop can be? What other requirements do they have? Finally, Table 1 should be updated - Crossbow is very extensible, and the Crossbow code forms the basis for the RNAseq analysis pipeline Myrna published in 2010: <http://genomebiology.com/2010/11/8/R83>

The paper dedicated to PMR, a Pilot implementation of MapReduce was accepted recently in MapReduce workshop in HPDC2012 and the reference is added for more detailed information.

**Reviewer 3 :**

The authors describe a novel implementation of MapReduce using " Pilot task and data management". Unfortunately most of the interesting details are hidding in reference [16[ which is submitted to another venue. This makes the review kind of ... hard.

Overall the authors describe much needed improvements of the Hadoop MapReduce implementation. The authors have convinced this reviewer that their implementation  is more efficient.

We revised to give a better introduction for PMR, according to the reviewer’s suggestion. As we discussed with our responses to the comments of other reviewers, our configuration for Hadoop is more like out-of-box approach, which presumably reflects the normal non-Hadoop-expert user experiences. Anyway, we revised our manuscript to stress the comparison is not made with the best optimization for Hadoop performance mainly due to the restraint with the system-wise file system of FutureGrid.