BioPig

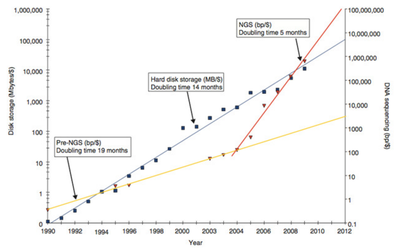
A Hadoop-based Analytic Toolkit for Large Scale Sequence Data

Kai Wang and Henrik Nordberg, Joint Genome Institute, Berkeley Lab

Abstract

The recent revolution in sequencing technologies has led to the exponential growth of sequence data. As a result, most of the current bioinformatics tools become obsolete as they fail to scale with data. To tackle this “data deluge”, here we introduce the BioPig sequence analysis toolkit as one of the solutions that scale to data and computation. BioPig is built upon the Apache’s Hadoop map-reduce system and the Pig data flow language. Its programmability greatly reduces development time for parallel bioinformatics applications. Testing linear algorithms implemented in BioPig with up to 500 Gb sequences demonstrates that it scales linearly with size of data. We also demonstrate that BioPig can be ported without modification on two independent pieces of Hadoop infrastructure, the Magellan system at NERSC, and the Amazon Elastic Compute Cloud. In summary, BioPig represents a novel program framework with the potential to greatly accelerate data-intensive bioinformatics analysis.

Motivation



Cost of sequencing is decreasing faster than the cost of storage. We need better algorithms and methods to handle TeraBases of data.

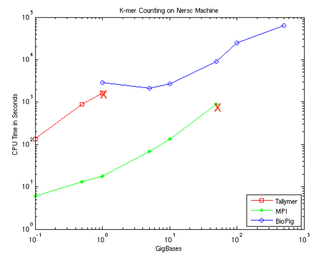
Easy to program

```
register /.../lib/biopig-core-0.3.0-job.jar
A = load '$reads' using gov.jgi.meta.pig.storage.FastaStorage as
(id: chararray, d: int, seq: chararray);
B = foreach A generate id, FLATTEN(
gov.jgi.meta.pig.eval.KmerGenerator(seq,              30)) as
(kmer:bytearray);
C = group B by kmer;
D = foreach C generate group, COUNT(B);
E = group D by $1;
F = foreach E generate group, COUNT(D);
store F into '$output';
```

Listing 1: Just 8 lines for K-mer histogram calculation.

Figure 1. BioPig uses Apache Pig and Hadoop. BioPig runs on one machine, or a cluster of machines. It ports easily between different architectures, e.g., NERSC Magellan and Amazon EC2.

Designed to handle very large data



Comparison of memory use and run time of Serial, MPI, and BioPig versions of K-mer histogram code.
BioPig is capable of handling 500GB of data.

Flexibility

```
#!/usr/bin/python
from org.apache.pig.scripting import *

PP = Pig.compile("""
register /…/biopig-core-0.3.0-job.jar;
-- load the target sequences
readindex = load '$indexFile' using PigStorage as
(seq: bytearray, kmer: bytearray);
-- now index the contigs
contigs = load '$contigFile' using PigStorage as
(geneid: chararray, seq: chararray);
...
        store notextended into 'output/data-$i';
""")

for i in range(50):
   stats = PP.bind().runSingle()
       contigFile = 'output/step-'+str(i)
```

Listing 2: Can use Python and JavaScript to take advantage of advanced control structures.

Conclusion

BioPig is designed for processing large datasets in bioinformatics applications. We found that BioPig is easy to use -- for the gene discovery pipeline, it requires 61 lines of BioPig script code. It is scalable -- it is able to process datasets as large as 500 Gb. It is portable – it can be ported to Amazon EC2 easily. It may be somewhat slower than handcrafted MPI. We note that performance is not the only criterion by which to evaluate the various methods. Indeed, for some methods it is their feasibility that is of highest importance. Some methods simply break down when applied to large data. When dealing with huge data sets, there is no way around having enough resources. BioPig shifts the need from expensive resources such as large memory (>= 1TB RAM) machines and/or parallel programming expertise, to commodity hardware and/or the expertise to run in the Cloud. Publication pending.

Acknownledgements: Karan Bhatia, Shane Cannon,Tatyana Smirnova, Aijazuddin Syed, Zhong Wang

The work conducted by the U.S. Department of Energy Joint Genome Institute is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.