Ogres: A Systematic Approach to Big Data Benchmarks

Geoffrey C.Fox[1], Shantenu Jha[2], Judy Qiu[1], Andre Luckow[2]
(1) School of Informatics and Computing, Indiana University, Bloomington, IN 47408, USA,
(2) RADICAL, Rutgers University, Piscataway, NJ 08854, USA

What is an Ogre?

The Berkeley Dwarfs [1] were an important step to define an exemplar set of parallel (high performance computing) applications. The recent NRC report [2] gave Seven Computational Giants Of Massive Data Analysis which make a good start to define critical types of data analytics problems. We proposed [3] Ogres -- an extension of these ideas based on an analysis by NIST of 51 big data applications [4]. Big Data Ogres provide a systematic approach to understanding applications, and as such they have facets which represent key characteristics defined both from our experience and from a bottom-up study of features of several individual applications. The facets capture common characteristics which are inevitably multi-dimensional and often overlapping. We note that in HPC, the Berkeley Dwarfs were very successful as patterns but did not get adopted as a standard benchmark set. Rather the NAS Parallel Benchmarks [5], Linpack [6], and (mini-)applications played this role. This suggests that benchmarks do not follow directly from patterns, but the latter can help by allowing one to understand breadth of applications covered by a benchmark set.

Ogres have Facets

We suggested that Ogres would have properties that we classified in four distinct dimensions or views. Each view consists of facets; when multiple facets are linked together, they describe classes of big data problem represented as an Ogre. One view of an Ogre is the overall **problem architecture** which is naturally closely related to although different from the machine architecture needed to support data intensive application. Then there is in more detail the **execution (computational) features** view that describing issues such as I/O versus compute rates, iterative nature of computation and the classic V's of Big Data defining problem size, rate of Change etc. The **data source & style** view includes facets specifying how the data is collected, stored and accessed. The final **processing** view has facets which describe classes of processing steps including algorithms and kernels. Ogres are specified by the particular value of a set of facets linked from the different views. The views contain the following facets.

Facets in Problem Architecture View: Pleasingly Parallel; Classic MapReduce; Map-Collective; Map Point-to-Point (graphs); Shared memory (as opposed to distributed parallel algorithm); Global Analytics; Single Program Multiple Data SPMD; Bulk Synchronous Processing BSP; Fusion; Dataflow?; Agents; Orchestration (workflow)
Facets in Execution View: Performance Metrics; Flops per I/O or Memory Byte; Communication Interconnect; Communication Synchronization; Dynamic?; Regularity; Iterative?; Volume; Velocity; Variety; Veracity; Data Abstraction(key-value, bag of words, spatial, vectors, sequence, graph); Metric Space or not?; $O(N^2)$ or $O(N)$?; Libraries needed?
Facets in Data Source&Style View: SQL/NoSQL/NewSQL?; Enterprise data model (warehouses); Files/Objects?; HDFS/Lustre/GPFS?; Archive/Batched/Streaming; Shared/Dedicated/ Transient/Permanent; Metadata/Provenance; Internet of Things; HPC Simulations; Geographic Information Systems;
Facets in Processing View: Micro-benchmarks; Local Analytics; Recommender Engine; Search/Query/Index; Classification; Learning; Linear/Quadratic Programming; Combinatorial

Optimization; Streaming; Alignment; Machine Learning; Nonlinear Optimization; Least Squares; Expectation Maximization; Linear Algebra Kernels; Graph Algorithms; Visualization

In our language instances of Ogres can form benchmarks. One can consider composite or atomic (simple, basic) benchmarks and for example, a clustering benchmark is an instance of an Ogre with a Map-Collective facet in the Problem Architecture view and the machine learning facet in the Processing view. The Execution view describes properties that could be different for different clustering algorithms and would often be measured in benchmarking process. Note a simple benchmark like this could ignore the data source&style view and just be studied for in memory data. Alternatively we can consider a composite benchmark linking clustering to different data storage mechanisms. A given benchmark can be associated with multiple facets in a single view; for example clustering has other problem architecture facets including SPMD, BSP, and Global Analytics.

Particular Benchmarks as instances of Ogres

Our approach suggests choosing benchmarks from Ogre instances that cover a rich range of facets. At this stage, we give some examples rather than trying to be comprehensive. Note that kernel benchmarks are instances of Ogre Processing facets and this is where the NAS parallel benchmarks or TeraSort [7] would fit. On the other hand, micro benchmarks such as MPI ping-pong and SPEC [8] are measures of Ogre execution facets.

Baru and Rabl's tutorial [9] has a thorough discussion of benchmarks such as the TPC series [10], HiBench [11], Yahoo Cloud Serving Benchmark [12], BigDataBench [13], BigBench [14] and Berkeley Big Data Benchmark [15] that quantify the Ogre data source&style facets.

The processing view has the well-known Graph500 [16] benchmarks (and associated machine ranking) but of course libraries like R [17], Mahout [18] and MLlib [19] also include many candidates for analytics benchmarks. We are part of a recent NSF project from the Dibbs (Data Infrastructure Building Blocks) program where one can use Ogres to classify the Building Blocks that are focus of this program. Below we list a few examples of problems we are studying with full set available at [20, 21]. Note each problem below can provide benchmarks for many different execution view facets.

Graph Problems: *Community detection, Subgraph/motif finding, Finding diameter, Clustering coefficient, Page rank, Maximal cliques, Connected component, Betweenness centrality, Shortest path* which are instances of the Graph Algorithm facet of the processing view and also of either the Map Point-to-Point and/or Shared memory facets in the Problem architecture view.

Spatial Analytics: *Spatial relationship based queries* from the Search/Query/Index and MapReduce facets; *Spatial Clustering* from Global Machine Learning, Map-Collective and Global Analytics facets; *Distance based queries* from Pleasingly Parallel and Search/Query/Index facets. These 3 benchmarks all have the spatial data abstraction facet.

Machine Learning in general and for image processing: several *Clustering* algorithms illustrating $O(N)$, $O(N^2)$, and Metric (non-metric) space execution view facets; *Levenberg-Marquardt Optimization* and *SMACOF Multi-Dimensional Scaling* with Linear Algebra Kernels and Expectation maximization facets from processing view; *TFIDF Search and Random Forest* with Pleasingly Parallel facets. All of course twinkle with the machine learning facet of the processing view.

Ogre-Driven Benchmarking

The suggested process is to examine current benchmarking and list facets they cover. Then augment with new benchmarks to cover those facets not addressed in initial choice. One must of course also address the many well studied general points of benchmarking such as agreeing on datasets with various sizes (Volume facet in execution view); requiring correct answers for each implementation and choice between pencil and paper and source code specification of benchmark.

References

1. Asanovic, K., R. Bodik, B.C. Catanzaro, J.J. Gebis, P. Husbands, K. Keutzer, D.A. Patterson, W.L. Plishker, J. Shalf, S.W. Williams, and K.A. Yelick. *The Landscape of Parallel Computing Research: A View from Berkeley*. 2006 December 18 [accessed 2009 December]; Available from: http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html.
2. Committee on the Analysis of Massive Data; Committee on Applied and Theoretical Statistics; Board on Mathematical Sciences and Their Applications; Division on Engineering and Physical Sciences; National Research Council, *Frontiers in Massive Data Analysis*. 2013: National Academies Press. http://www.nap.edu/catalog.php?record_id=18374
3. Shantenu Jha, Judy Qiu, Andre Luckow, Pradeep Mantha, and Geoffrey C. Fox, *A Tale of Two Data-Intensive Approaches: Applications, Architectures and Infrastructure*, in *3rd International IEEE Congress on Big Data Application and Experience Track*. June 27- July 2, 2014. Anchorage, Alaska. http://arxiv.org/abs/1403.1528.
4. NIST, *NIST Big Data Public Working Group (NBD-PWG) Use Cases and Requirements.* 2013. http://bigdatawg.nist.gov/usecases.php
5. NASA Advanced Supercomputing Division. *NAS Parallel Benchmarks*. 1991 [accessed 2014 March 28]; Available from: https://www.nas.nasa.gov/publications/npb.html.
6. A. Petitet, R. C. Whaley, J. Dongarra, and A. Cleary. *HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers*. 2008 September 10 [accessed 2014 July 19,]; Available from: http://www.netlib.org/benchmark/hpl/.
7. Apache Hadoop. *TeraSort map/reduce sort*. [accessed 2015 January 12]; Available from: http://hadoop.apache.org/docs/current/api/org/apache/hadoop/examples/terasort/package-summary.html.
8. *SPEC's Benchmarks*. [accessed 2015 January 12]; Available from: https://www.spec.org/benchmarks.html.
9. Chaitan Baru and Tilmann Rabl. *Tutorial 4 " Big Data Benchmarking" at 2014 IEEE International Conference on Big Data*. 2014 [accessed 2015 January 2]; Available from: http://cci.drexel.edu/bigdata/bigdata2014/tutorial.htm.
10. Transaction Processing Performance Council. *TPC Benchmarks*. [accessed 2015 January 12]; Available from: http://www.tpc.org/information/benchmarks.asp.
11. Lan Yi. *Experience with HiBench: From Micro-Benchmarks toward End-to-End Pipelines at WBDB 2013 Workshop* 2013 [accessed 2015 January 12];
12. Brian Cooper. *Yahoo Cloud Serving Benchmark with Key-value store (Accumulo, Cassandra, Hbase, MongoDB etc.) benchmarks*. [accessed 2015 January 12]; Available from: https://github.com/brianfrankcooper/YCSB/
13. Jianfeng Zhan. *BigDataBench: A Big Data Benchmark Suite from ICT, Chinese Academy of Sciences*. [accessed 2015 January 12]; Available from: http://prof.ict.ac.cn/BigDataBench/.
14. Tilmann Rabl. *Big Data Analytics Benchmark (BigBench)*. [accessed 2015 January 12]; Available from: http://www.msrg.org/projects/project_view.php?id=12.
15. AMPLab. *Berkeley Big Data Benchmark: Redshift, Hive, Shark, Impala, Stinger/Tez benchmarks*. [accessed 2015 January 12]; Available from: https://amplab.cs.berkeley.edu/benchmark/.
16. *Graph 500 Benchmarks*. [accessed 2015 January 12]; Available from: http://www.graph500.org/specifications.
17. *R open source statistical library*. [accessed 2012 December 8]; Available from: http://www.r-project.org/.
18. *Apache Mahout Scalable machine learning and data mining* [accessed 2012 August 22]; Available from: http://mahout.apache.org/.
19. *Machine Learning Library (MLlib)*. [accessed 2014 April 1]; Available from: http://spark.apache.org/docs/0.9.0/mllib-guide.html.
20. Indiana, Rutgers, Virginia Tech, Kansas, Stony Brook, Arizona State, and Utah. *CIF21 DIBBs: Middleware and High Performance Analytics Libraries for Scalable Data Science Summary*. 2014 December 18 [accessed 2015 January 12]; Available from: http://grids.ucs.indiana.edu/ptliupages/presentations/Dibbs%20-%20Overall%20-%20Dec18-2014.pptx
21. *HPC-ABDS Kaleidoscope of over 270 Apache Big Data Stack and HPC Tecnologies*. [accessed 2014 April 8]; Available from: http://hpc-abds.org/kaleidoscope/.