TACC Technical Report IMP-27

Data Analytics in IMP

Victor Eijkhout*

March 16, 2017

* eijkhout@tacc.utexas.edu, Texas Advanced Computing Center, The University of Texas at Austin

Abstract

Data analytics in IMP

The following IMP reports are available or under construction:

IMP-00 The IMP Elevator Pitch
IMP-01 IMP Distribution Theory
IMP-02 The deep theory of the Integrative Model
IMP-03 The type system of the Integrative Model
IMP-04 Task execution in the Integrative Model
IMP-05 Processors in the Integrative Model
IMP-06 Definition of a 'communication avoiding' compiler in the Integrative Model (under construction)
IMP-07 Associative messsaging in the Integrative Model (under construction)
IMP-08 Resilience in the Integrative Model (under construction)
IMP-09 Tree codes in the Integrative Model
IMP-10 Thoughts on models for parallelism
IMP-11 A gentle introduction to the Integrative Model for Parallelism
IMP-12 K-means clustering in the Integrative Model
IMP-13 Sparse Operations in the Integrative Model for Parallelism
IMP-14 1.5D All-pairs Methods in the Integrative Model for Parallelism (under construction)
IMP-15 Collectives in the Integrative Model for Parallelism
IMP-16 Processor-local code (under construction)
IMP-17 The CG method in the Integrative Model for Parallelism (under construction)
IMP-18 A tutorial introduction to IMP software (under construction)
IMP-19 Report on NSF EAGER 1451204.
IMP-20 A mathematical formalization of data parallel operations
IMP-21 Adaptive mesh refinement (under construction)
IMP-22 Implementing LULESH in IMP (under construction)
IMP-23 Distributed computing theory in IMP (under construction)
IMP-24 IMP as a vehicle for software/hardware co-design, with John McCalpin (under construction)
IMP-25 Dense linear algebra in IMP (under construction)
IMP-26 Load balancing in IMP (under construction)
IMP-27 Data analytics in IMP (under construction)

1 Apache Spark

Spark is a big data tool that can be described in IMP. The basic object is an RDD: Resilient Distributed Dataset, which is analogous to an IMP object.

What is a Seq? What is a Block?

In this report we describe how IMP can cover the expressive functionality of Spark. We will not go into fault tolerance and such.

Map Apply a function to an RDD, giving a new one. We realize this by applying the function and letting $\gamma = \alpha$.

FlatMap Apply a function to return a Seq. We expand the distribution accordingly.

MapPartitions Run a function separately on each block of the partition.

MapPartitionsWithIndex Run a function separately on each block of the partition, and include the index of the partition block.

Filter Select the elements for which a specified function is true. To first order we model this by locally contracting the input distribution to the 'true' elements.

Union Combine two datasets. The resulting distribution is obvious.

!! Intersection !! Very tricky! This needs an Allgather or so. Better: bucket brigade.

!! Distinct !! Keep distinct elements. This is global too, probably through a bucket brigade of comparisons. Which copy do we keep? Lowest location? That may lead to unbalance.

!! GroupByKey !! Questions: how do we relate the number of keys and number of locales? There must be a concept of affinity, but is that otherwise visible?

ReduceByKey Similar.

SortByKey This is basically sorting. No interaction with affinity that we don't already know.

Join Take $\langle K,V\rangle$ and $\langle K,W\rangle$ datasets and return $\langle K,(V,W)\rangle$. Just locally blow up: affinity of (V,W) is affinity of V. This can of course require load balancing.

2 Clustering algorithms

See [1].

3 Minebench

Data mining benchmark [2, 3]; see table 1.

The codes as given are OpenMP only.

K-means See our report [1].

PLSA For Smith-Waterman, see HPSC-**??**.

Application	Category	Description
ScalParC	Classification	Decision tree classification
Naive Bayesian	Classification	Simple statistical classifier
K-means	Clustering	Mean-based data partitioning method
Fuzzy K-means	Clustering	Fuzzy logic-based data partitioning method
HOP	Clustering	Density-based grouping method
BIRCH	Clustering	Hierarchical Clustering method
Eclat	ARM	Vertical database, Lattice transversal techniques used
Apriori	ARM	Horizontal database, level-wise mining based on Apriori property
Utility	ARM	Utility-based association rule mining
SNP	Classification	Hill-climbing search method for DNA dependency extraction
GeneNet	Structure Learning	Gene relationship extraction using microarray-based method
SEMPHY	Structure Learning	Gene sequencing using phylogenetic tree-based method
Rsearch	Classification	RNA sequence search using stochastic Context-Free Grammars
SVM-RFE	Classification	Gene expression classifier using recursive feature elimination
PLSA	Optimization	DNA sequence alignment using Smith-Waterman optimization method

Table 1: Minebench codes

References

[1] Victor Eijkhout. K-means clustering in the integrative model. Technical Report IMP-12, Integrative Programming Lab, Texas Advanced Computing Center, The University of Texas at Austin, 2014.

[2] Northwest Engineering Center for Ultrascale Computing. Minebench homepage. http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html.

[3] R. Narayanan, B. Ozisikyilmaz, J. Zambreno, G. Memik, and A. Choudhary. Minebench: A benchmark suite for data mining workloads. In *2006 IEEE International Symposium on Workload Characterization*, pages 182–188, Oct 2006.