膨大なデータの処理を簡単化のための高レベル並列プログラムミング

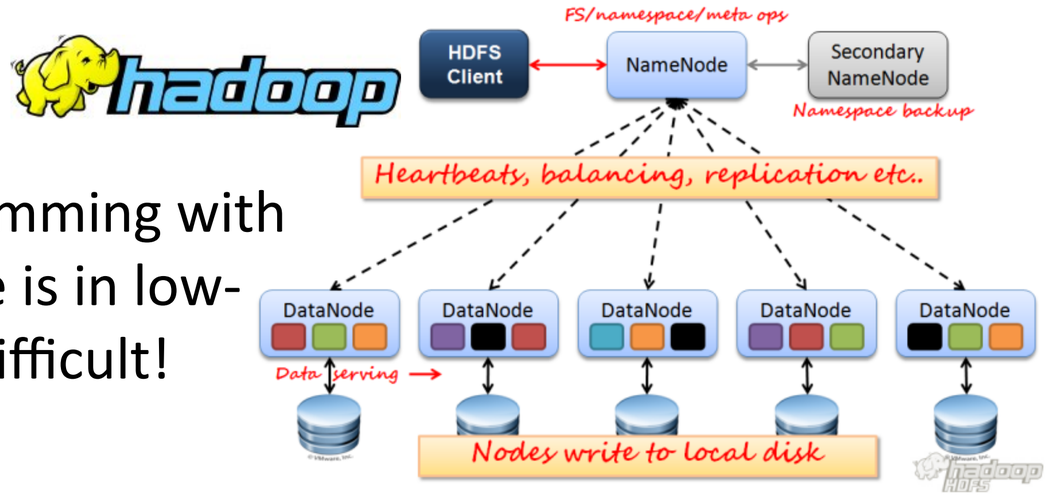
High-level Parallel Programming for Simplifying Processing of Big Data

劉 雨　胡 振江
国立情報学研究所

動機と目標

インターネット時代のビッグ・データは増加速い、処理難しい

For internet-scale data, traditional infrastructures and programming paradigms are no more applicable. New frameworks like MapReduce are developed and studied for handle PB-level data.



But, programming with MapReduce is in low-level thus difficult!

MapReduceに基づく、高レベル並列プログラミング

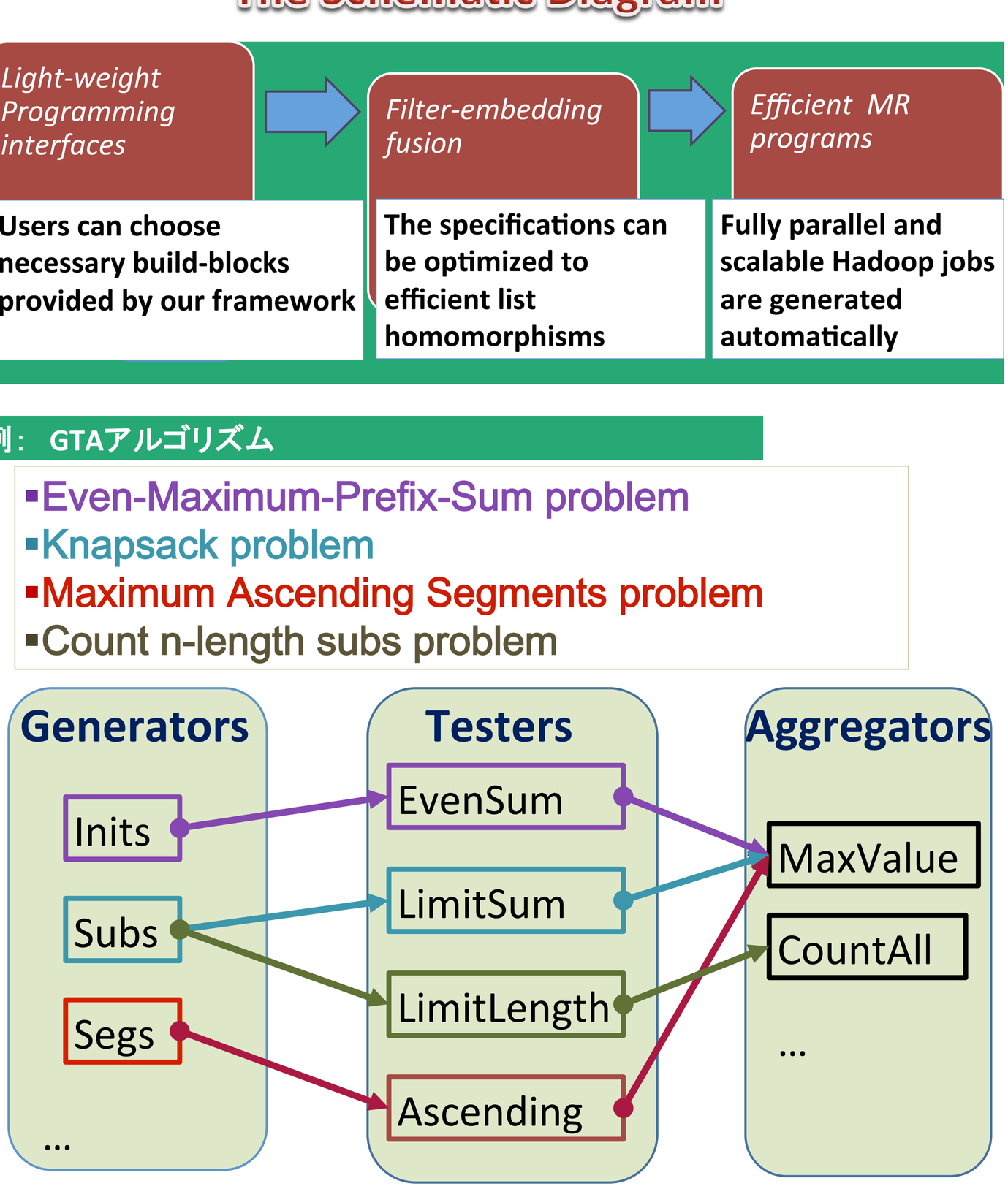
- Users write high-level (simpler) programs;
- Our framework transform them to MapReduce programs which can be deployed on large cluster.

The Schematic Diagram

Light-weight Programming interfaces	*Filter-embedding fusion*	*Efficient MR programs*
Users can choose necessary build-blocks provided by our framework	**The specifications can be optimized to efficient list homomorphisms**	**Fully parallel and scalable Hadoop jobs are generated automatically**

例： GTAアルゴリズム

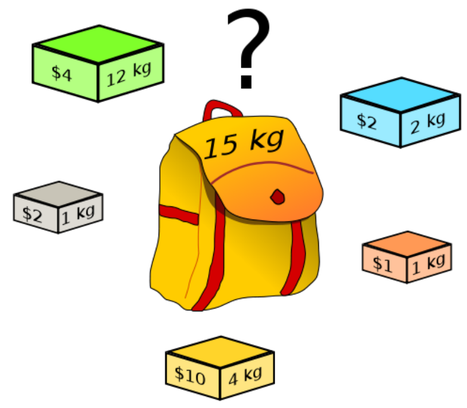
- Even-Maximum-Prefix-Sum problem
- Knapsack problem
- Maximum Ascending Segments problem
- Count n-length subs problem



Programming with GTA is to define the specification by using GTA build-blocks

実際の例

Knapsack 問題の計算



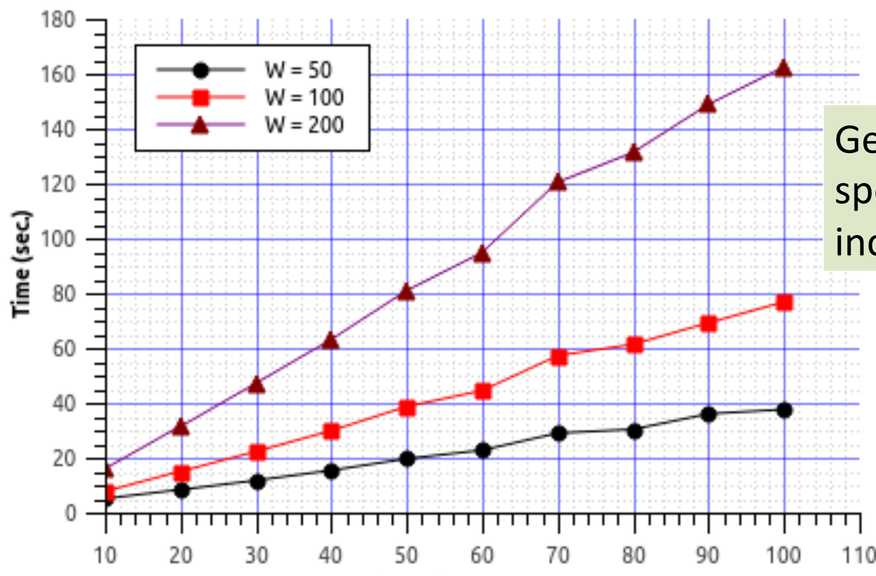
$$\text{maximize } \sum_{i=1}^{n} v_i x_i$$
$$\text{subject to } \sum_{i=1}^{n} w_i x_i \leq W \ , \ x_i \in \{0,1\}$$

A few lines of code, but scalable and efficient, using our framework:

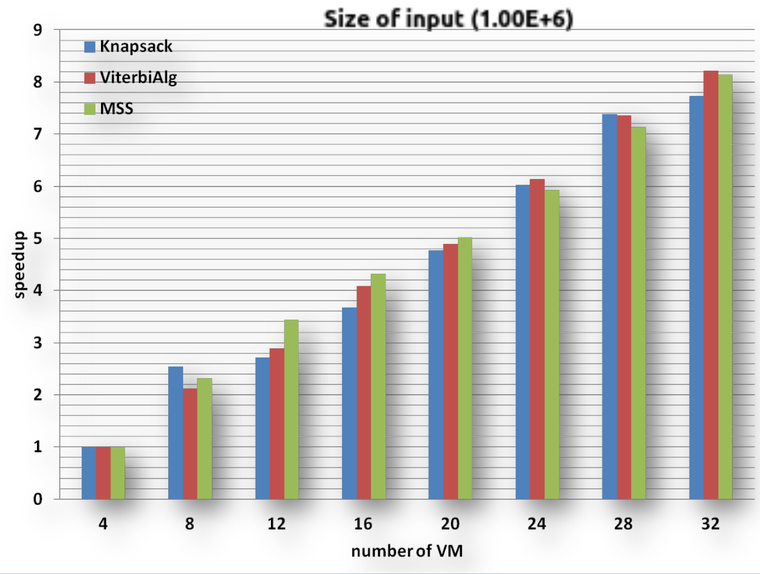
```
//G+T+A
def knapsack(contex: SparkContext, x: spark.RDD[DS.KnapsackItem]) = {

  val allSellects = new AllSelects[KnapsackItem] //generator
  val withLimit_100 = new WeightLimit(100)       // tester
  val gta = generate(allSellects)    .
            filter (withLimit_100) .
                    aggregate (maxTotalValue)    //aggregator
  val rst = gta.postProcess(x.map(gta.f(_).get).reduce(gta.combine(_, _)))
```

Using Hadoop-APIs maybe need 100s lines

高い性能とスケーラビリティを持つ



Get near linear speedup be increasing CPUs



Double the nodes of cluster can gain almost twice speedup

結論

- **Performance:** Good speedup and scalability
- **Programmability:** Clear and Light-weight interface
- **Practicability:** Various problems have been resolved

Reference

- Kento Emoto, et al, ***Generate, Test, and Aggregate --A Calculation-based Framework for Systematic Parallel Programming with MapReduce***. *ESOP 2012.*
- Yu Liu, et al, **A Generate-Test-Aggregate Parallel Programming Library**, PMAM 2013.

連絡先：　胡　振江（Zhenjiang HU）／ 東京都　千代田区　一ツ橋　国立情報学研究所　教授
TEL：03-4212-2530　FAX：03-4212-2533　Email：hu@nii.ac.jp