Hybrid Parallelization Strategies for Large-Scale Machine Learning in SystemML

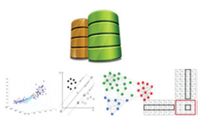
Matthias Boehm, Shirish Tatikonda, Berthold Reinwald, Prithviraj Sen,
Yuanyuan Tian, Douglas R. Burdick, Shivakumar Vaithyanathan

IBM Research – Almaden

Background

Motivation

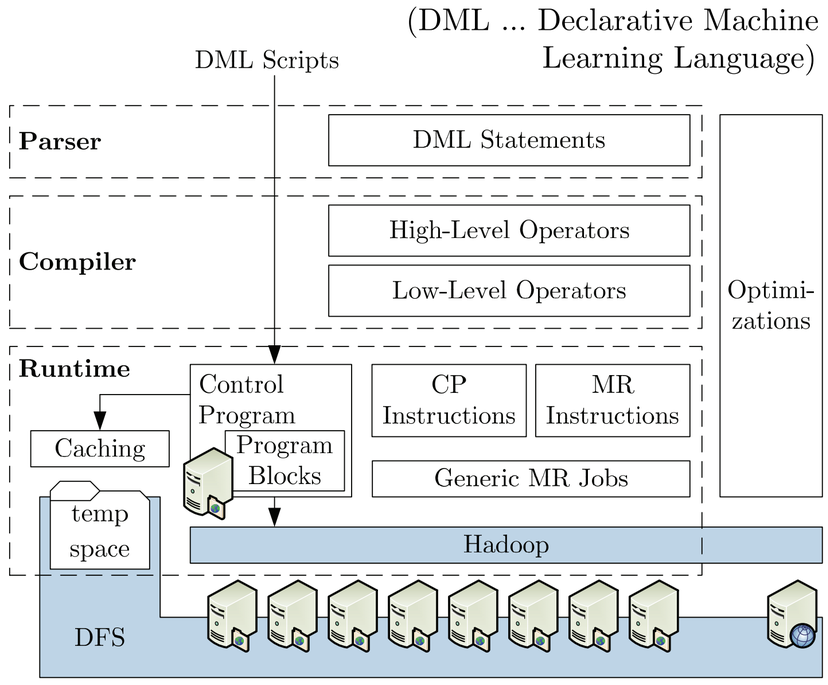
- **Problem**
  - Analyzing big data
  - Advanced analytics / ML



- **SystemML Approach**
  - Declarative ML on top of MapReduce
  - Flexibility, optimization, data independence
  - Primarily data parallelism

- **Challenges** (related via memory constraints)
  - **#1:** MR vs in-memory computations
  - **#2:** No support for task parallelism
  - **Major challenge:** efficiency and scalability for variety of use case

SystemML Architecture



Taxonomy Task-Parallel ML

	Single Model	Multiple Models
Disjoint Data	SQM, DataGen, SGD	Univariate Stats, Indep. Models
Overlapping Data	SQM, CSVM, SGD*	Bivariate Stats, Meta CV
All Data	Dist, kNN, EL	Meta, EL

Example Pairwise Correlation

(**Bivariate statistics:** Pearson's R, Anova F, Chi-square, Degree of freedom, P-value, Cramers V, Spearman)

```
D = read("./input/D");
m = nrow(D);
n = ncol(D);
R = matrix(0, rows=n, cols=n);
parfor( i in 1:(n-1) ) {
   X = D[ ,i];
   m2X = centralMoment(X,2);
   sigmaX = sqrt( m2X*(m/(m-1.0)) );
   parfor( j in (i+1):n ) {
      Y = D[ ,j];
      m2Y = centralMoment(Y,2);
      sigmaY = sqrt( m2Y*(m/(m-1.0)) );
      R[i,j] = cov(X,Y) / (sigmaX*sigmaY);
}}
write(R, "./output/R");
```

Runtime Strategies

Conceptual Design: master/worker (task: group of parfor iterations)

Task Partitioning

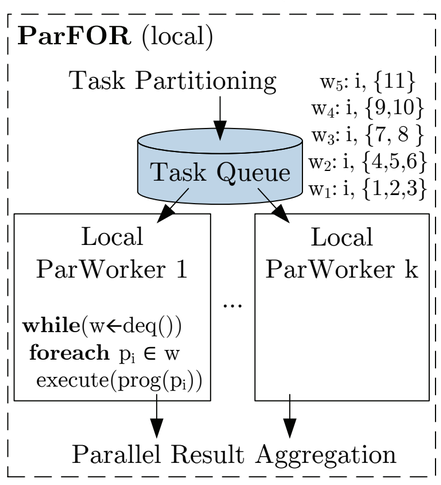
- **Fixed-size schemes: naïve, static, fixed**
- **Self-scheduling: factoring**

Example factoring: N=101, k=4 → (13, 13, 13, 13, 7, 7, 7, 7, 3, 3, 3, 3, 2, 2, 2, 2, 1)

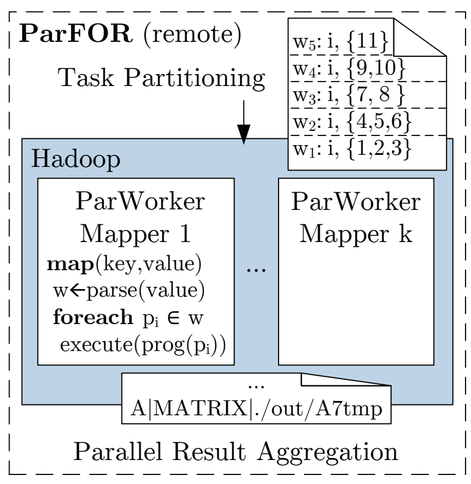
Task Execution

- **Local**
- **Remote**
- **Hybrid**

Local execution (multicore)



Remote execution (cluster)



Result Aggregation

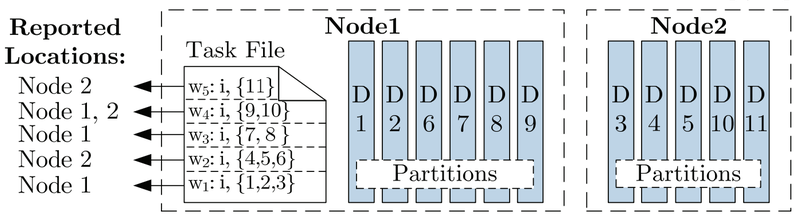
- **w/ and w/o compare**
- **Local in-memory/local file/remote MR**

Runtime Optimizations

- **Data partitioning**
- **Data locality**

Example data locality:

```
X=D[,i]
```

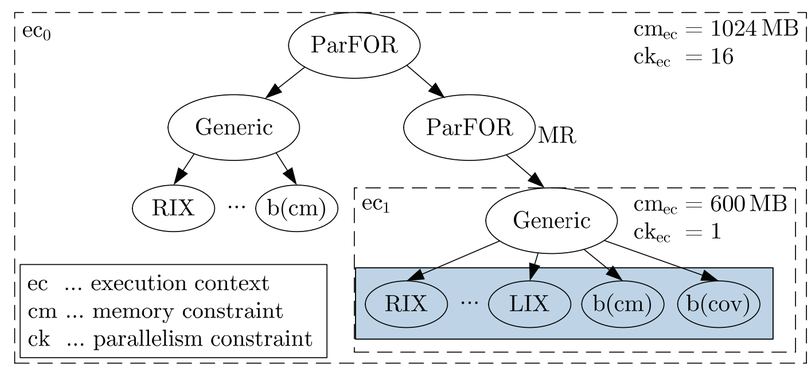


Optimization Framework

Design: runtime optimization

Problem Formulation

- **Plan Tree**
  - **Nodes N_P**
  - **Exec type et**
  - **Parallelism k**
  - **Attributes A**
  - **Height h**
  - **Exec contexts EC_P**



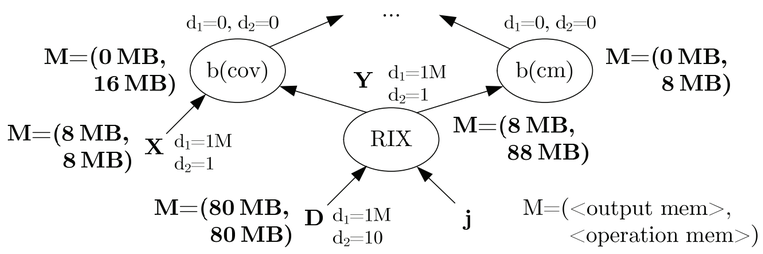
- **Optimization objective**

$$\phi_2: \min \hat{T}(r(P)) \quad s.t. \quad \forall ec \in \mathcal{EC}_P : \hat{M}(r(ec)) \leq cm_{ec} \wedge K(r(ec)) \leq ck_{ec}.$$

Mapped Hop DAG

Cost Model

- **HOP DAG/program size propagation**
- **Worst-case memory estimates**
- **Time estimates**
- **Plan tree statistics aggregation**



Optimizer Overview

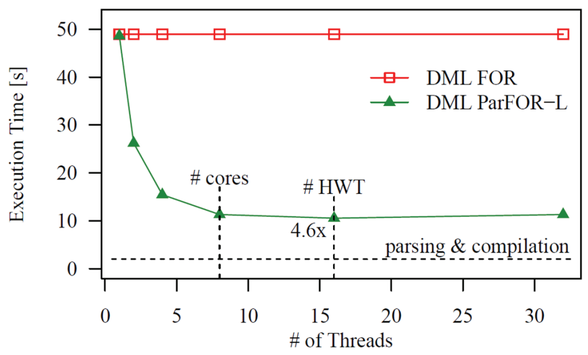
- **Time- and memory-based cost model, w/o shared reads**
- **Heuristic high-impact rewrites**
- **Transformation-based search strategy with global opt scope**

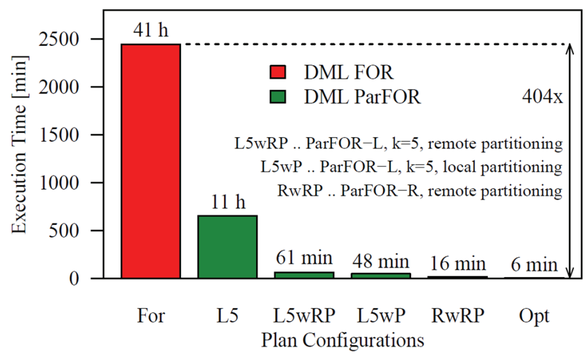
Experiments (as of 07/2013)

Experimental Setting

- **SystemML 07/2013**
- **R 2.15.1 64bit (doMC: 1x8, doSNOW: 5x8)**
- **Spark 0.8.0 (5x16 workers, 5x16GB memory)**
- **Physical Cluster:** 5 nodes, each 2x4 (16HWT), 64GB RAM, 1.5TB storage, 1GbE, SLES 11 64bit
- **Hadoop Cluster:** IBM Hadoop 1.1.1, IBM JDK 1.6.0 64bit, map/reduce capacity: 80/40, HDFS block size 128MB, JVM size master/map/reduce: 1GB, ratio 0.7

Bivariate Statistics





	SystemML	R	R doMC	R doSNOW	Spark DP	Spark TP
XS: 10^3 x 10^2	**3s**	3s	3s	19s	176s	6s / (11s)
S: 10^5 x 10^2	**30s**	21s	6s	45s	319s	13s / (20s)
M: 10^7 x 10^2	**363s**	2,109s	751s*	870s*	7,875s	✖ / (776s)
L: 10^7 x 10^3	**17,321s**	✖	✖	✖	**7.4E7s****	✖ / ✖

Linear Regression – Feature Subsampling

	SystemML	R	R doMC	R doSNOW
S: 10^4 x 10^3	**180s**	239s	1,872s	2,599s
M: 10^6 x 10^3	**238s**	1,631s	3,034	3,739s
L: 10^7 x 10^3	**574s**	✖	✖	✖

Logistic Regression – Parameter Search

	SystemML	R	R doMC	R doSNOW
S: 10^4 x 10^3	**31s**	370s	60s	35s
M: 10^6 x 10^3	**239s**	19,869s	4,621s	1,097s
L: 10^6 x 10^5	**619s**	49,860s	12,355s	2,550s