On the Latency-Accuracy Tradeoff in Approximate MapReduce Jobs

Juan F. Pérez
Universidad del Rosario, Colombia

Robert Birke and Lydia Chen
IBM Research Zurich, Switzerland

IEEE INFOCOM – May, 2017
Outline

1. Introduction
2. System Operation
3. The Model
4. Experimental assessment
5. Wrap-up
What?
MapReduce Jobs

- Large data sets
- Exploit large number of parallel resources
- Open source implementations (Hadoop)
- Well studied for batch jobs
- **Online** job execution
- Deadlines: constraints on response time
MapReduce Jobs

- Large data sets
- Exploit large number of parallel resources

- **Online** job execution
- Deadlines: constraints on response time
Online MapReduce Jobs

- Deadlines: constraints on response time
- Process all data → exact answer
- **Approximate** Computing
- Process some data → approximate answer
- Save computing time and resources
- Comply with deadlines
Why?
Latency-Accuracy Tradeoff

- Drop more data $\rightarrow$ Reduce latency $\rightarrow$ Reduce accuracy
- Drop less data $\rightarrow$ Increase latency $\rightarrow$ Increase accuracy
- Solve the **Latency-Accuracy Tradeoff**
Goal?
Solve Latency-Accuracy Tradeoff

- Given a *Latency* objective $\rightarrow$ Determine *Achievable Accuracy*
- Given an *Accuracy* objective $\rightarrow$ Determine Achievable Latency
- Additional considerations:
  - Jobs arrive in a random fashion
  - Queueing times waiting for jobs in front to be processed
  - How to drop data for approximate computations?
Solve Latency-Accuracy Tradeoff

- Given a \textit{Latency} objective $\rightarrow$ Determine \textit{Achievable Accuracy}
- Given an \textit{Accuracy} objective $\rightarrow$ Determine Achievable Latency
- Additional considerations:
  - Jobs arrive in a random fashion
  - Queueing times waiting for jobs in front to be processed
  - How to drop data for approximate computations?
How?
Our Approach to the Latency-Accuracy Tradeoff

- Stochastic models to predict Job Latency
- Capture Online MapReduce operation
- Capture Approximate Computing
- Data Drop options
Our Approach to the Latency-Accuracy Tradeoff

- Stochastic models to predict **Job Latency**
- Capture Online MapReduce operation
- Capture Approximate Computing
- Data Drop options
Outline

1. Introduction
2. System Operation
3. The Model
4. Experimental assessment
5. Wrap-up
Basic Operation
Basic Operation

C computing slots

MR Jobs Queue
Basic Operation

C computing slots

MR Jobs Queue
Basic Operation

C computing slots

MR Jobs Queue

Map task queue

M M

M M
Basic Operation

- MR Jobs Queue
- Map task queue
- C computing slots
Basic Operation

C computing slots

MR Jobs Queue

Map task queue
Basic Operation

C computing slots

MR Jobs Queue

Map task queue

J. F. Pérez, R. Birke, L. Chen
Latency vs. Accuracy Approx. MapReduce
IEEE INFOCOM 2017
Basic Operation

C computing slots

MR Jobs Queue

Map task queue
Basic Operation

C computing slots

MR Jobs Queue

Map task queue
Basic Operation

- Computing slots
- MR Jobs Queue
- Map task queue
Basic Operation

C computing slots

MR Jobs Queue

Reduce task queue

R

R

R

R
Basic Operation

C computing slots

MR Jobs Queue

Reduce task queue

R

R

R
Basic Operation

C computing slots

MR Jobs Queue

Reduce task queue

R R
Basic Operation

C computing slots

MR Jobs Queue

Reduce task queue

J. F. Pérez, R. Birke, L. Chen
Latency vs. Accuracy Approx. MapReduce
IEEE INFOCOM 2017
Basic Operation

- MR Jobs Queue
- Map task queue

C computing slots
Approximate Computing
Data Drop
Early Dropping
Data Drop: Task drop - Early
Data Drop: Task drop - Early
Data Drop: Task drop - Early
Straggler Dropping
Data Drop: Task drop - Straggler

![Diagram showing task drop - straggler](image)
Data Drop: Task drop - Straggler
Data Drop: Task drop - Straggler
Input Sampling
Data Drop: Input Sampling

- M
- R

Time

M

M

R

M

R

M
Data Drop: Input Sampling
Data Drop: Input Sampling
Error
\[ t_{N_{m-1,1-\alpha/2}} \sqrt{N_m \left( \left( \frac{1}{\theta_m} - 1 \right) s_u^2 + \left( \frac{1}{\eta_m} - 1 \right) s_i^2 \right)} , \]
Outline

1. Introduction
2. System Operation
3. The Model
4. Experimental assessment
5. Wrap-up
Basic Operation - Main parameters

- $C$ computing slots
- $N_m$: number of map tasks
- $N_r$: number of reduce tasks
- $1/\mu_m$: mean map task execution time
- $1/\mu_r$: mean reduce task execution time
Basic Operation - Job Service Time

- Job Service Time: **Phase-type** distribution
- Phase: number of tasks to complete
The Model

Basic Operation - Job Service Time

\[ N_m + N_r \rightarrow N_m - 1 + N_r \rightarrow N_m - 2 + N_r \rightarrow \ldots \]

\[ C \mu \]

J. F. Pérez, R. Birke, L. Chen
Latency vs. Accuracy Approx. MapReduce
IEEE INFOCOM 2017 47 / 78
Basic Operation - Job Service Time

The Model

C + Nr → (C-1)Nr → (C-2)Nm

J. F. Pérez, R. Birke, L. Chen
Latency vs. Accuracy Approx. MapReduce
IEEE INFOCOM 2017 48 / 78
Basic Operation - Job Service Time

$$1 + N_r \xrightarrow{\mu} N_r \xrightarrow{C\mu} N_r - 1 \xrightarrow{C\mu} \ldots$$
Basic Operation - Job Service Time

\[ C \mu \rightarrow (C-1) \mu \rightarrow (C-2) \mu \]

J. F. Pérez, R. Birke, L. Chen
Latency vs. Accuracy Approx. MapReduce
IEEE INFOCOM 2017 50 / 78
Basic Operation - Job Service Time

\[ \text{...} \rightarrow 2 \mu \rightarrow 1 \mu \rightarrow \text{...} \]
Modeling Approximate Computation
Introducing Early Task Dropping

- Execute fraction $\theta_m$ map tasks only
- Drop fraction $1 - \theta_m$ map tasks
- Effective number of map tasks: $\overline{N}_m = \lceil N_m \theta_m \rceil$
- Replace $N_m$ by $\overline{N}_m$
Introducing Input Sampling

- Process fraction $\eta_m$ of each map task data only
- Ignore fraction $1 - \eta_m$ of each map task data
- Effective mean map task execution time: $\frac{1}{\mu_m} = \frac{\eta_m}{\mu_m}$
- Replace $\mu_m$ by $\bar{\mu}_m$
Early Dropping + Input Sampling

Transition Rates:

\[ f(n) = \begin{cases} 
  C\mu_m, & \text{if } \bar{N}_r + C < n \leq \bar{N}_r + \bar{N}_m, \\
  (n - \bar{N}_r)\mu_m, & \text{if } \bar{N}_r < n \leq \bar{N}_r + C, \\
  C\mu_r, & \text{if } C < n \leq \bar{N}_r, \\
  n\mu_r, & \text{if } 0 < n \leq C.
\]
Straggler Dropping

- Start with \( N_m + N_r \) tasks
- Decrease one by one until \( N_m - (\bar{N}_m - 1) + N_r \) are left
- Jump down to \( N_r \)
- Effective number of map tasks executed: \( \bar{N}_m \)
Straggler Dropping + Input Sampling

Transition Rates:

\[ f(n) = \begin{cases} 
\min\{n - N_r, C\} \bar{\mu}_m, & N_r + N_m - \bar{N}_m + 1 \leq n \\
\min\{n, C\} \bar{\mu}_r, & N_r - \bar{N}_r + 1 \leq n \leq N_r.
\end{cases} \]
Analysis
Analysis

- FCFS scheduling
- Job service time: phase-type (PH) - $(\alpha, G)$
- Job arrival process: Markovian arrival process (MAP)
- MAP/PH/1 queue
- Sengupta’s paper: age-based analysis
- $(X(t), A(t), S(t))$: age of job in service, arrival process phase, service phase
Analysis

Solve a matrix-integral equation

\[ T = I_d \otimes G + \int_0^\infty \exp(Tx) \left( \exp(D_0x)D_1 \otimes \tilde{G} \right) dx, \]

- Obtain waiting time distribution \((\beta_w, B_w)\)
- Known service-time distribution \((\alpha, G)\)
- Obtain response time distribution \((\beta_r, B_r)\):

\[ \beta_r = [\phi \beta_w \ (1 - \phi)\alpha], \quad B_r = \begin{bmatrix} B_w & \ (-B_w1)\alpha \\ 0 & G \end{bmatrix}, \]
Overlapping Execution
FCFS Scheduling - Non-overlapping

Job queue

J3  J2

Pool of M slots

idle slots  Job 1

R1  R1
R1  R1
R1  R1
Overlapping scheduling

- Allow jobs to start Map phase as soon as slots become available.
- At most 2 jobs in service: one in Map phase, one in Reduce phase.
Overlapping Scheduling

- No longer a MAP/PH/1 queue
- Service times depend on system state
- High load: high chance back-to-back jobs
- Low load: similar to non-overlapping
Overlapping Scheduling

- Generalize Sengupta’s analysis
- \((X(t), A(t), M(t), R(t), Y(t))\)
- \(M(t)\): number of remaining map tasks of oldest job
- \(R(t)\): number of remaining reduce tasks of oldest job
- \(Y(t)\): number of remaining map tasks of youngest job
Overlapping Scheduling

Three phases:

- \( M \) one job, map phase
- \( R \) one job, reduce phase
- \( R/M \) two jobs, oldest in reduce phase, youngest in map phase
Overlapping Scheduling

(Steady-state) Service time distribution: $\text{PH}(\beta_s, B_s)$

$B_s = \begin{bmatrix} Q^{R/M,R/M} & Q^{R/M,M} & Q^{R/M,R^*} \\ 0 & Q^{M,M} & Q^{M,R^*} \\ 0 & 0 & B^{R^*,R^*} \end{bmatrix}, \quad (1)$

$\beta_s = \begin{bmatrix} \beta^{R/M}_s \\ \beta^M_s \\ \beta^{R^*}_s \end{bmatrix}. \quad (2)$
Outline

1. Introduction
2. System Operation
3. The Model
4. Experimental assessment
5. Wrap-up
Experimental assessment

FCFS - Early Dropping

![Graph showing latency vs. accuracy for different utilizations (20%, 50%, 80%) with varying \( \theta_m \) values.](image-url)
FCFS - Early vs Straggler Dropping

(a) Early dropping

(b) Straggler dropping
Experimental assessment

FCFS - Early vs Straggler Dropping vs Input Sampling

(a) Early dropping
(b) Straggler dropping
(c) Input sampling
Observation 1

Best Mechanisms are:
1. Straggler dropping
2. Input sampling
3. Early dropping

Observation 2

An $x\%$ data/task dropping removes
1. $x\%$ longest tasks (straggler dropping)
2. $x\%$ all tasks, including longest (input sampling)
3. $x\%$ tasks (early dropping)
Varying number of tasks

(a) Input sampling

(b) Straggler dropping
FCFS vs. Overlapping (Early dropping)

(a) FCFS

(b) Overlapping
FCFS vs. Overlapping (Straggler dropping)

(a) FCFS

(b) Overlapping
Experimental assessment

FCFS vs. Overlapping (No approximation)

\[ \lambda \]

\[ \text{RT 99} \]

- Exclusive
- Overlap

J. F. Pérez, R. Birke, L. Chen

Latency vs. Accuracy Approx. MapReduce

IEEE INFOCOM 2017 76 / 78
Outline

1. Introduction
2. System Operation
3. The Model
4. Experimental assessment
5. Wrap-up
Model captures

- Straggler Dropping > Input Sampling > Early Dropping
- Overlapping > FCFS Scheduling
- Latency - Accuracy Tradeoff

Given a:

- **Latency Budget**: what accuracy can be expected/offered?
Wrap-up

Model captures

- Straggler Dropping > Input Sampling > Early Dropping
- Overlapping > FCFS Scheduling
- Latency - Accuracy Tradeoff

Given a:

- **Latency Budget**: what accuracy can be expected/offered?
- **Accuracy Budget**: what latency can be expected/offered?
Wrap-up

Model captures

- Straggler Dropping > Input Sampling > Early Dropping
- Overlapping > FCFS Scheduling
- Latency - Accuracy Tradeoff

Given a:

- **Latency Budget**: what accuracy can be expected/offered?
- **Accuracy Budget**: what latency can be expected/offered?
Wrap-up

Model captures

- Straggler Dropping > Input Sampling > Early Dropping
- Overlapping > FCFS Scheduling
- Latency - Accuracy Tradeoff

Given a:

- **Latency Budget**: what accuracy can be expected/offered?
- **Accuracy Budget**: what latency can be expected/offered?

Thank you!