Resource Usage Characterization for Social Networks Analytics on Spark

Irene Manotas[†], Rui Zhang[*], Min Li[↑], Renu Tewari[*], Dean Hildebrand[*]
[†]University of Delaware, [*]IBM Research – Almaden, [↑]IBM TJ-Watson Research Center

Social Network Analytics on Big Data Platforms

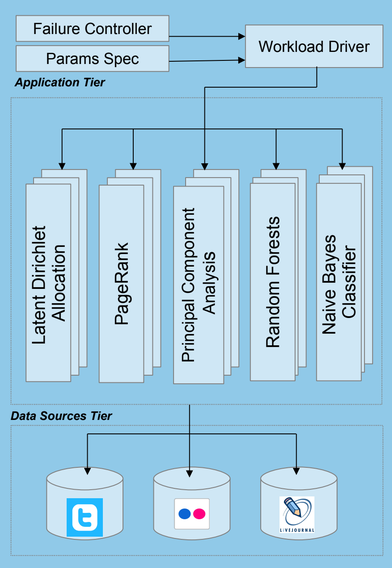
Online Social Networks (OSNs) impose new challenges for Big Data Analytics: terabytes of data being generated, billions of users being hosted, different data types for analysis, etc.

Graph algorithms (e.g. Page Rank), text analytics (e.g. Latent Dirichlet Allocation) and other machine learning methods are critical to making sense of OSN data.

Implementing the aforementioned OSN analytics algorithms on big data platforms like Spark enables more efficient analysis, but incurs complex resource usage patterns to the underlying systems.

We are not aware of prior studies about the resource usage characterization of these algorithms on big data platforms like Spark, when it comes to analyzing OSN data.

Workload Provider Design



- Enables data patterns/resource utilization analysis for a set of workloads with different input parameters
- Provides a mechanism to investigate platform's performance under failures (e.g., node failures)

Cluster Configuration

- 5 Nodes cluster (Hadoop 2.6.0 and Spark 1.4.0)
- 1 Resource Manager (YARN)
- 1 Name Node
- 4 DN/NM

Node's Description:

- 46 GB Memory
- 16 CPU
- 202 GB Disk (HDFS)
- 1 Gigabit Ethernet

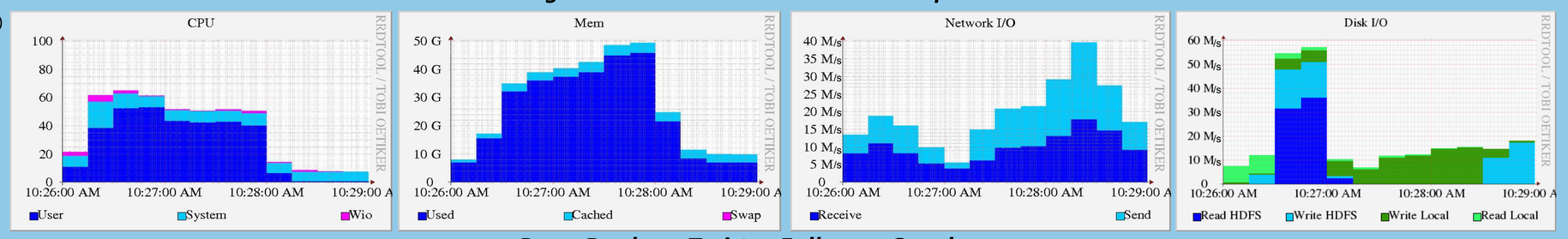
Experiments and Data Patterns

Config. NE/EM/C	Data/Workload	Execution Time	Input H/M/D	Shuffle R/W	Stages/Tasks
5/33GB/16	LiveJournal/PR	2.8 min	1.0GB/4.5GB/0	27GB/1.2GB	21/168
5/33GB/16	Twitter09/PR	168 min	25GB/43GB/530KB	146GB/44GB	43/4000
5/33GB/16	Tweets/LDA	18 min	1.1GB/22GB/0	12GB/15GB	78/678

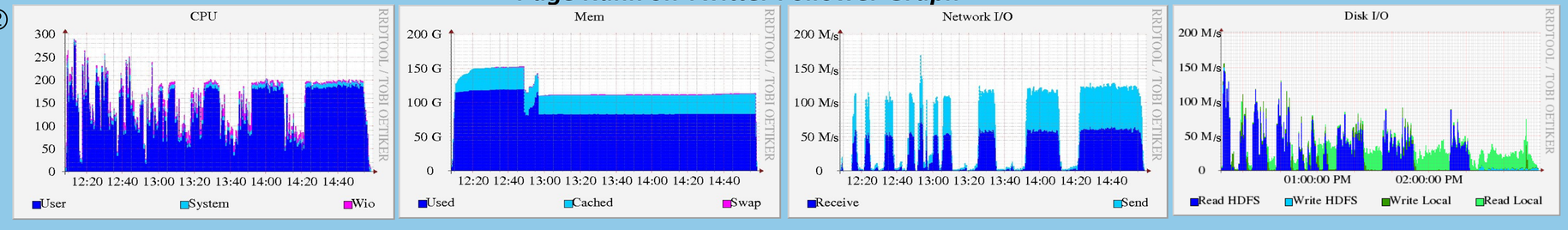
NE/EM/C=Number Executors/Executor Memory/CPU cores
PR=Page Rank
LDA=Latent Dirichelt Allocation
H/M/D=Hadoop/Memory/Disk

- Different Iterative algorithms could exhibit contrasting memory Patterns and Number of Stages.

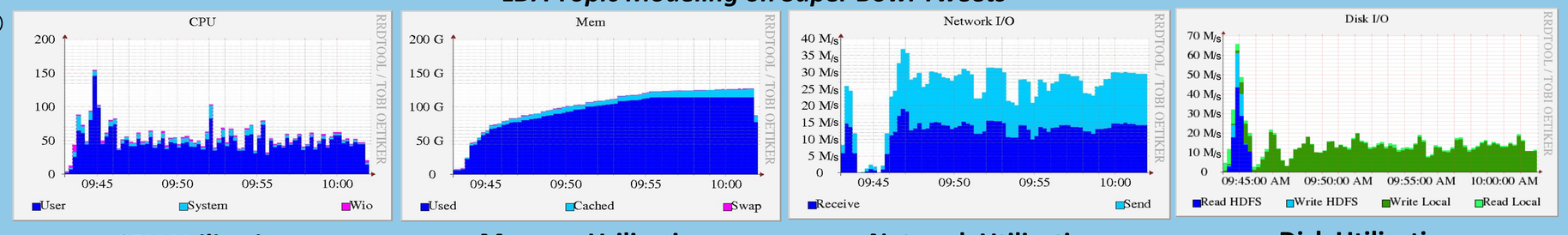
Page Rank on Social Live Follower Graph



Page Rank on Twitter Follower Graph



LDA Topic Modeling on Super Bowl Tweets



CPU Utilization | **Memory Utilization** | **Network Utilization** | **Disk Utilization**

- I/O patterns in Page Rank change as the input data size increases:

  – for the smaller dataset (LiveJournal, 1GB) writes are dominant (local writes into temporary space).
  – for the bigger dataset (Twitter, 25GB), reads are dominant (HDFS reads and local reads from temporary space).

- When the input data size is large, Page Rank produces heavy shuffle operations that require a considerable amount of temporary file space. Failure to provision sufficient temp space could lead to lost executors or even entire job failures.

- LDA exhibits a constant read/write pattern in local disk, which is representative of the shuffle operations originated by the term-topic distribution updates in each iteration.

- Page Rank network I/Os are bursty, whereas LDA network I/Os present a more uniform pattern.

Conclusions

- OSN analytics (e.g. graph analysis like Page Rank) on Spark can exhibit dramatically different resource usage patterns, depending on the data input size and type.
- For OSN analytics on Spark, performance problems can occur in unexpected resources such as the local disk space allocated for Spark temporary data.

Future Work

- Analyze more workloads i.e., Principal Component Analysis, Random Forest and Naive Bayes Classifier.
- Conduct detailed I/O characterization on distributed storage.
- Correlate the algorithm's logic with the resource utilization for the different workloads.
- Enable Fault-tolerance Analysis for workloads' execution on Spark.