Service Oriented Big Data Processing Performance Analysis

for
MPSoC 2016

Da Qi Ren, Junfeng Zhao

Huawei US R&D Center
Santa Clara, CA 95050

HUAWEI

Performance Model for Service in Data Center and Cloud

End to end big data benchmarking has become an extreme attention of ICT industry, the related techniques are being investigated by numerous hardware and software vendors.

Compute and storage devices, as one of the core components of a data center system, need specially designed approaches to measure, evaluate and analyze their performance.

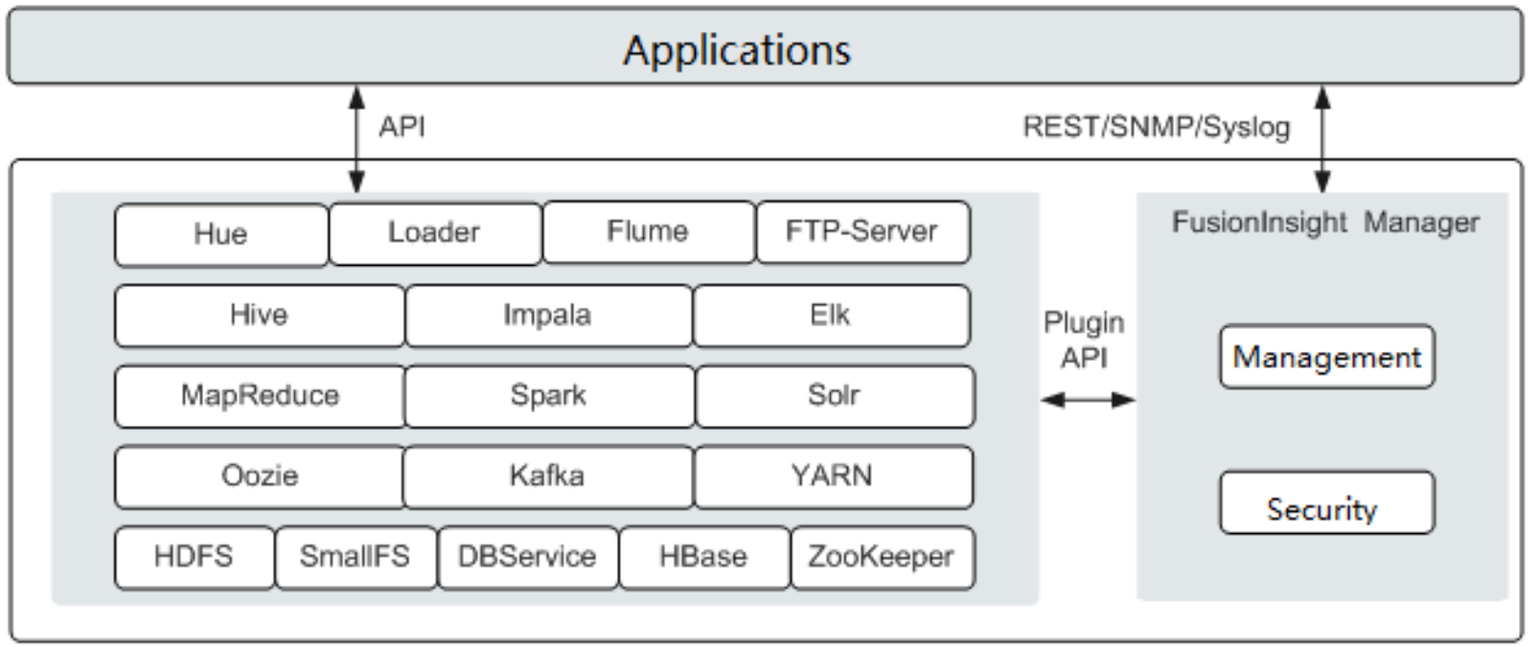
This talk introduces our methods to create the performance model based on workload characterization, algorithm level behavior tracing and capture, and software platform management.

The functionality and capability of our methodology for quantitative analysis of big data storage have been validated through benchmarks and measurements performed on real data center system.

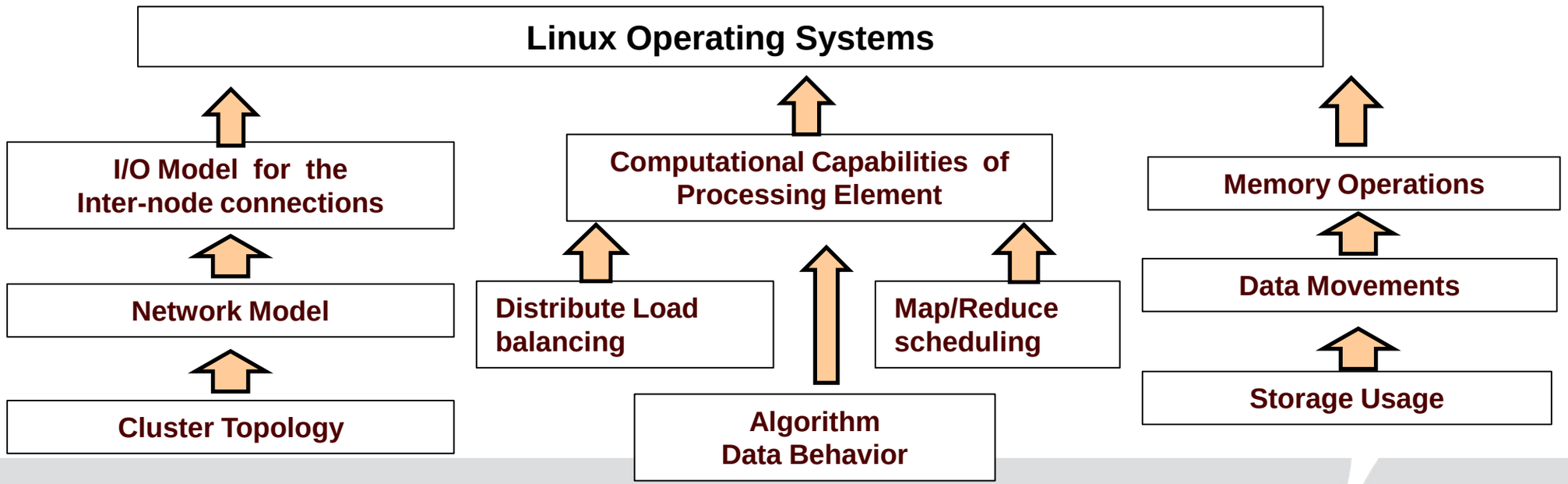
A FusionInsight System for Big Data

Performance Modeling and Analysis

Software Stack



Performance Analysis



Program Behavior Model from Measurement

1. Correlates data from all the experiments based on the common sampling rate and common time line;
2. Analyze parameter for the application's (Data Sorting) behavior
3. Analyze parameter for the application's (Transactions) behavior
4. Analyze parameter for the application's (K-Means Clustering) behavior
5. Identified program characters and create leading markers;
6. Identified program segments and perform detailed analysis on each segment;
7. Developed a Model that can use data captured from the systems stimuli and explore bottlenecks and dependencies.

TPC-BB Execution steps

End-to-End Benchmark

- ❑ **Loading**
- ❑ **Power Test (single user run)**
- ❑ **Throughput Test I (multi user run)**
- ❑ **Data Maintenance**

Execution

Query	Input Datatype	Processing Model	Query	Input Datatype	Processing Model
#1	Structured	Java MR	#16	Structured	Java MR (OpenNLP)
#2	Semi-Structured	Java MR	#17	Structured	HiveQL
#3	Semi-Structured	Python Streaming MR	#18	Unstructured	Java MR (OpenNLP)
#4	Semi-Structured	Python Streaming MR	#19	Structured	Java MR (OpenNLP)
#5	Semi-Structured	HiveQL	#20	Structured	Java MR (Mahout)
#6	Structured	HiveQL	#21	Structured	HiveQL
#7	Structured	HiveQL	#22	Structured	HiveQL
#8	Semi-Structured	HiveQL	#23	Structured	HiveQL
#9	Structured	HiveQL	#24	Structured	HiveQL
#10	Unstructured	Java MR (OpenNLP)	#25	Structured	Java MR (Mahout)
#11	Unstructured	HiveQL	#26	Structured	Java MR (Mahout)
#12	Semi-Structured	HiveQL	#27	Unstructured	Java MR (OpenNLP)
#13	Structured	HiveQL	#28	Unstructured	Java MR (Mahout)
#14	Structured	HiveQL	#29	Structured	Python Streaming MR
#15	Structured	Java MR (Mahout)	#30	Semi-Structured	Python Streaming MR

Power Measured Processes

- Power measurement for Loading
- Single User Test (*i* run)
- Throughput Test I (multi user run)
- Data Maintenance
- Throughput Test II (multi user run)
- Result

$$E_{load} = P_{load} T_{load}$$

$$E^{i}_{power} = P^{i}_{power} T^{i}_{power}$$

$$E^{k}_{th} = P^{k}_{th} T^{k}_{th} (k = 1)$$

$$E_{mt} = P_{mt} T_{mt}$$

$$E^{k}_{th} = P^{k}_{th} T^{k}_{th} (k = 2)$$

$$E_{total} = E^{1}_{th} + E^{2}_{th} + E_{mt} + E_{load}$$

Energy Metric for TPC-BB

When TPC option is chosen for reporting, the TPC-BB metric reports the power per performance and is expressed as

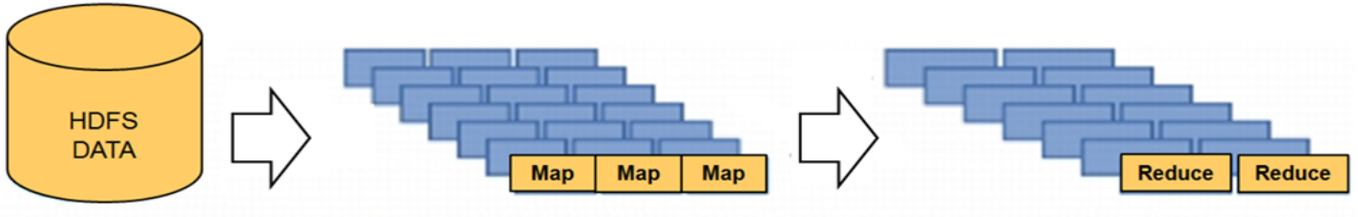
QphBB@SF.

(see TPC-Energy specification for additional requirements).

Each secondary metric shall be referenced in conjunction with the scale factor at which it was achieved.

For example, Load Time references shall take the form of Load Time @ SF, or “Load Time = 10 hours @ 300GB”.

Performance Issues in Each Layer of Big Data Computing Platform



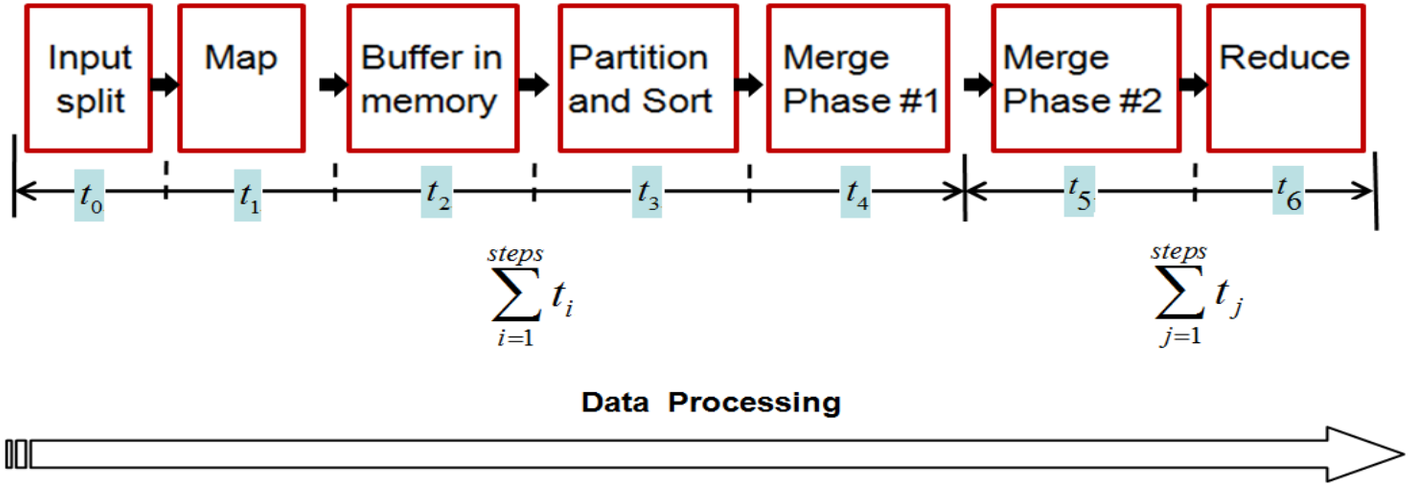
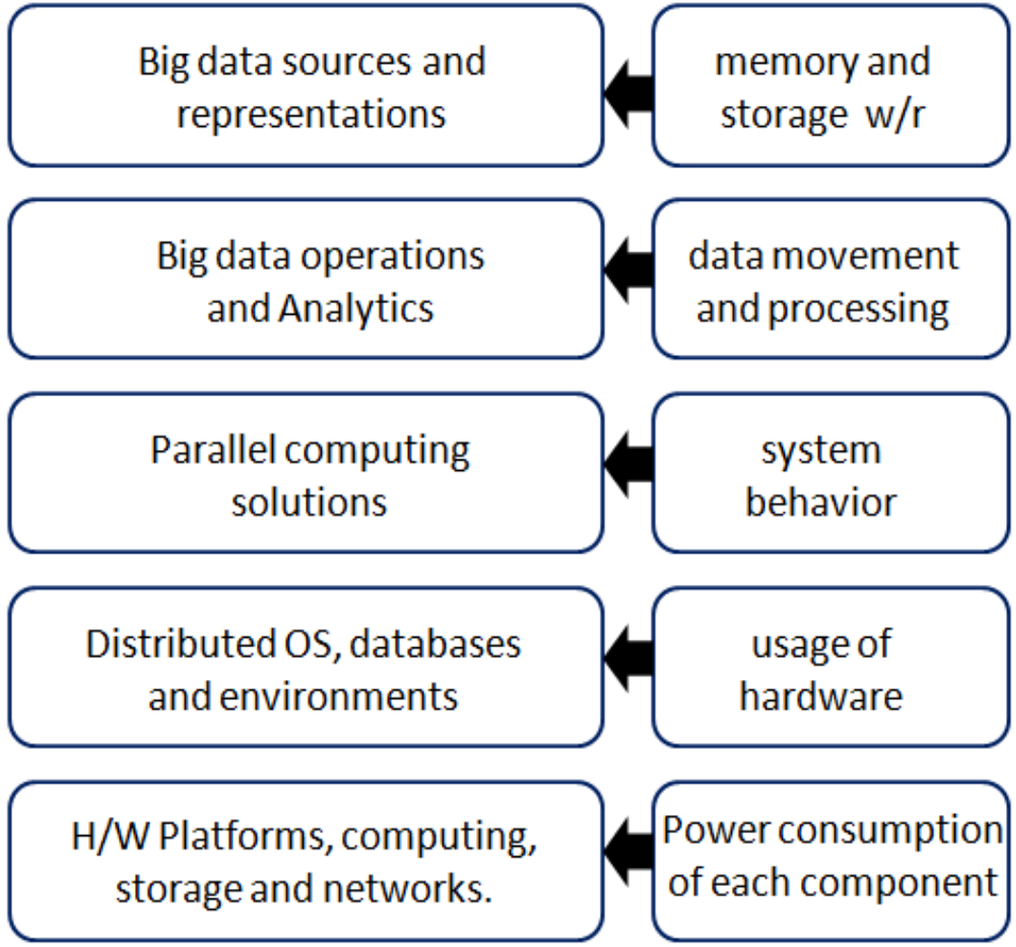
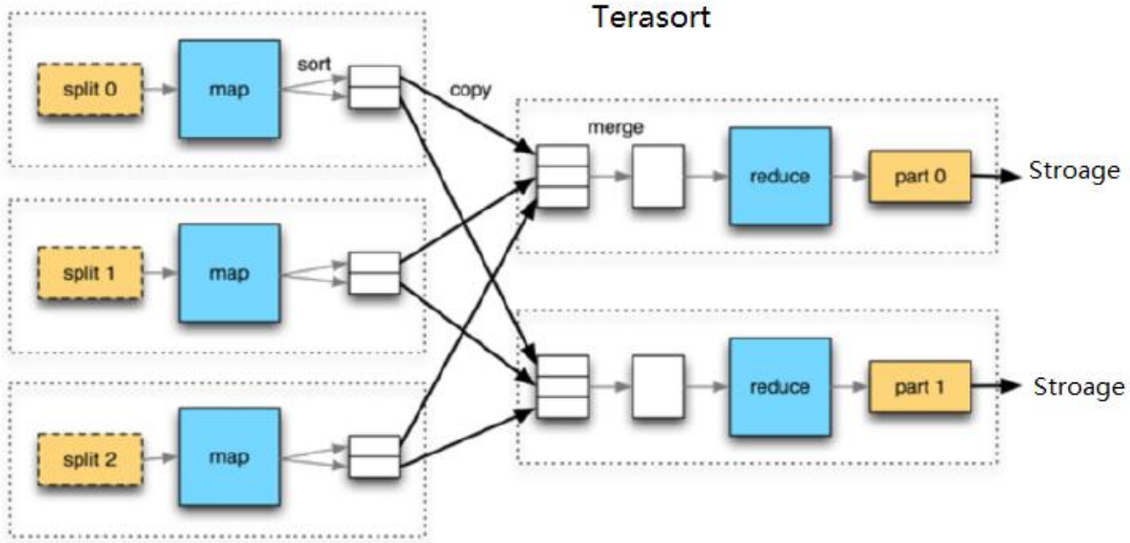
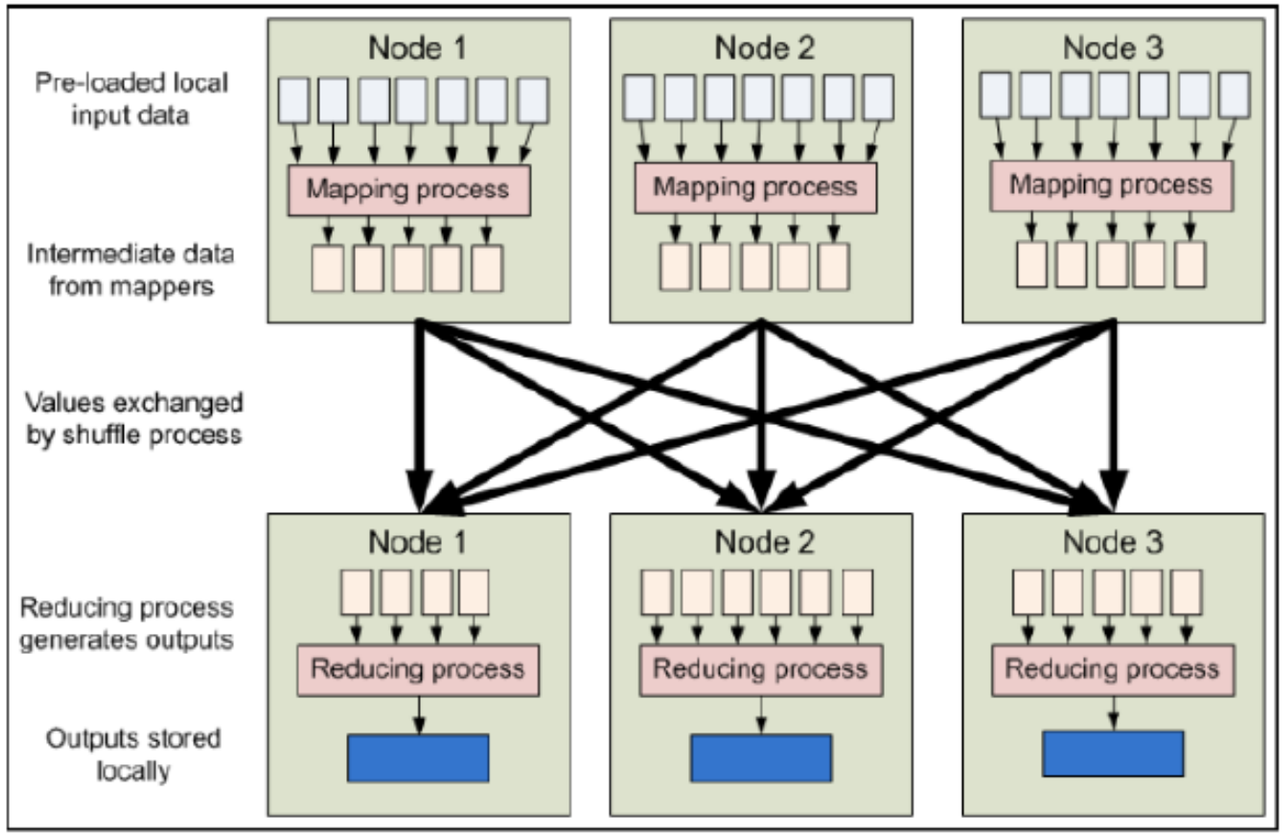


FIG. 3

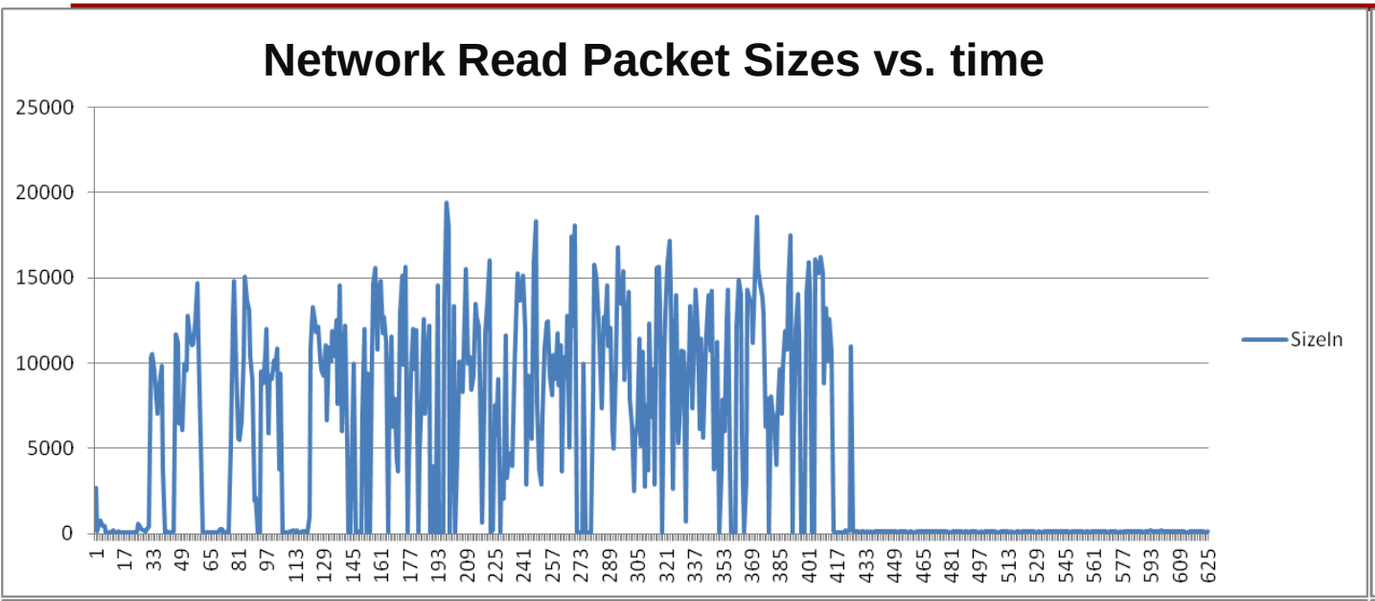


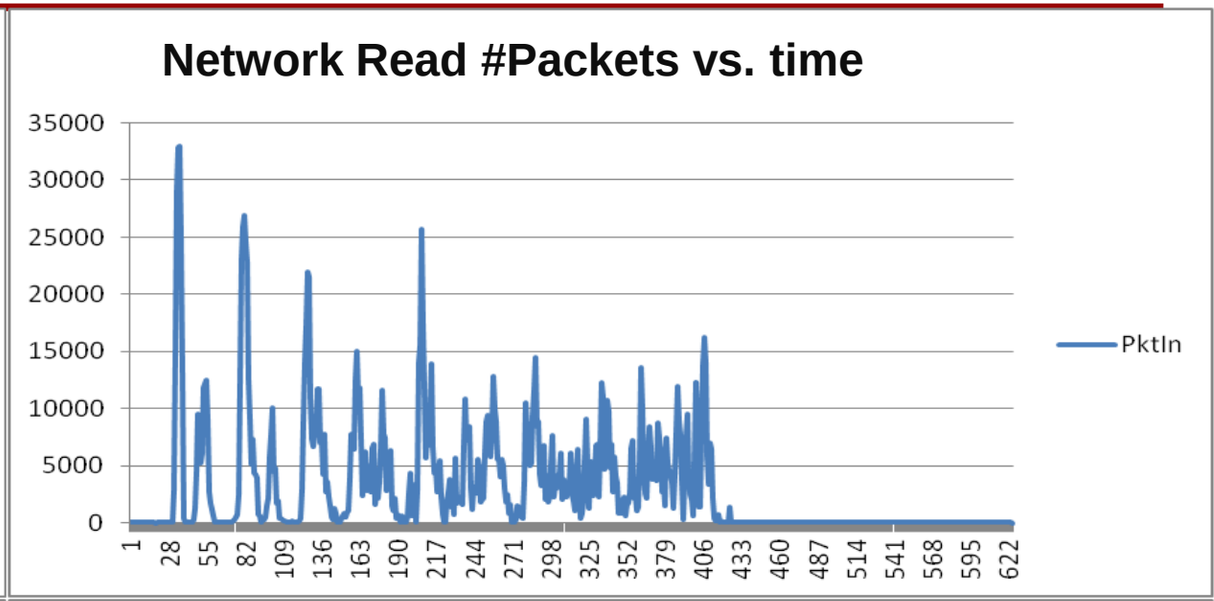
Algorithm Behaviors in K-Means Clustering

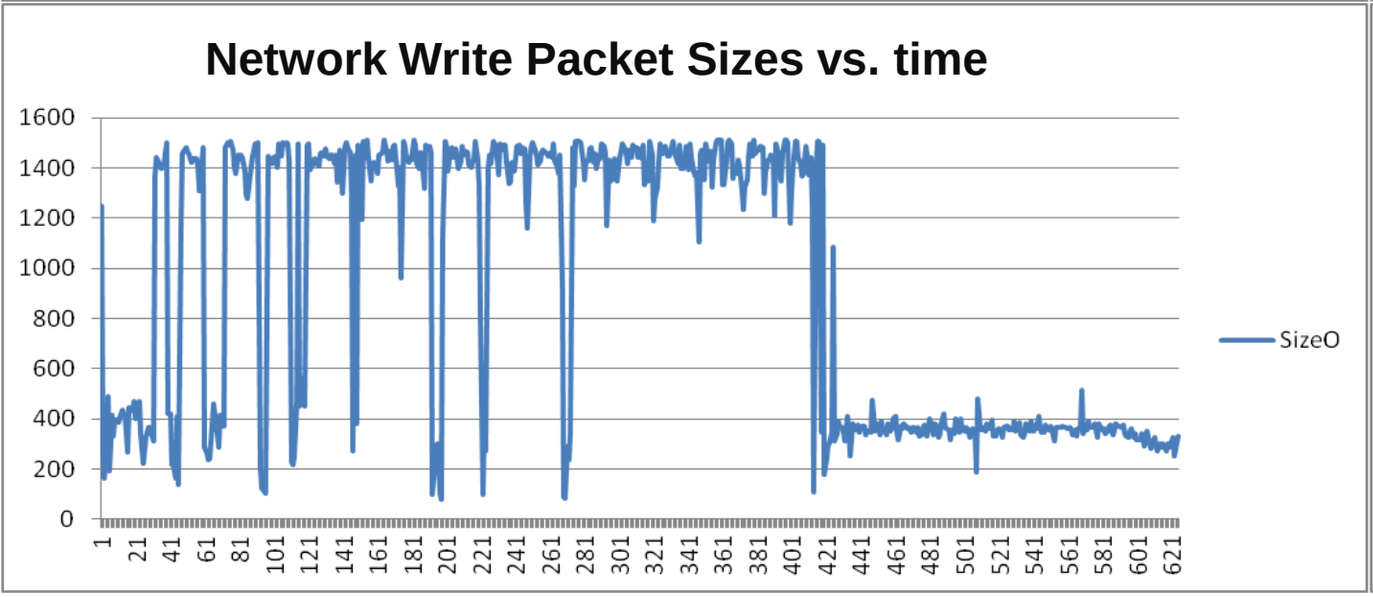


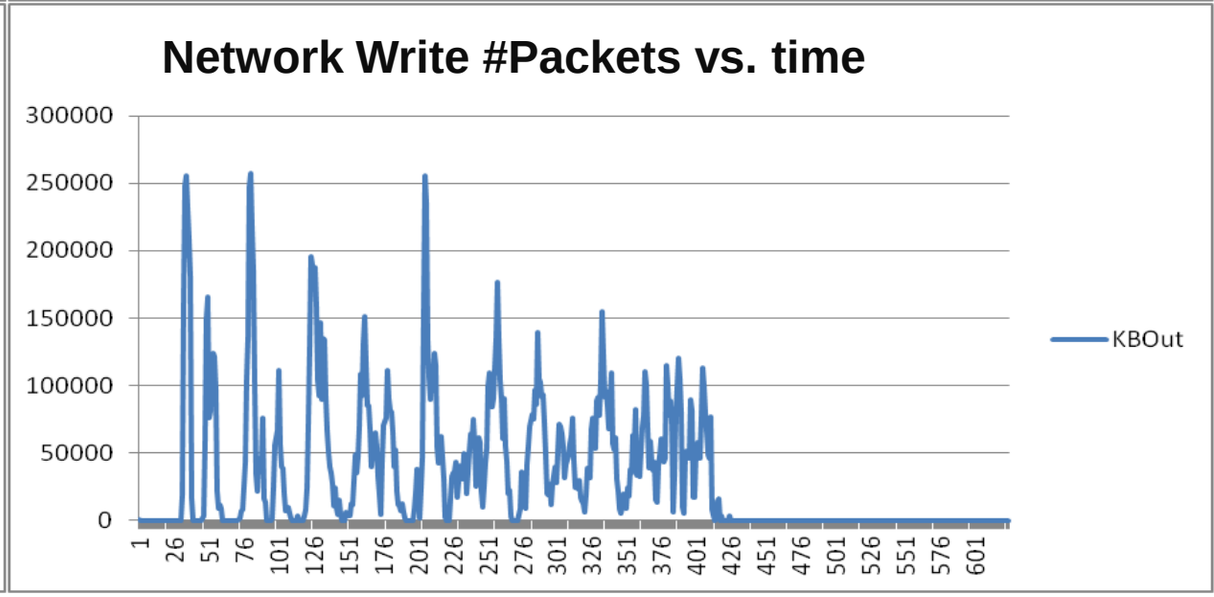


Sorting Network I/O distribution characters

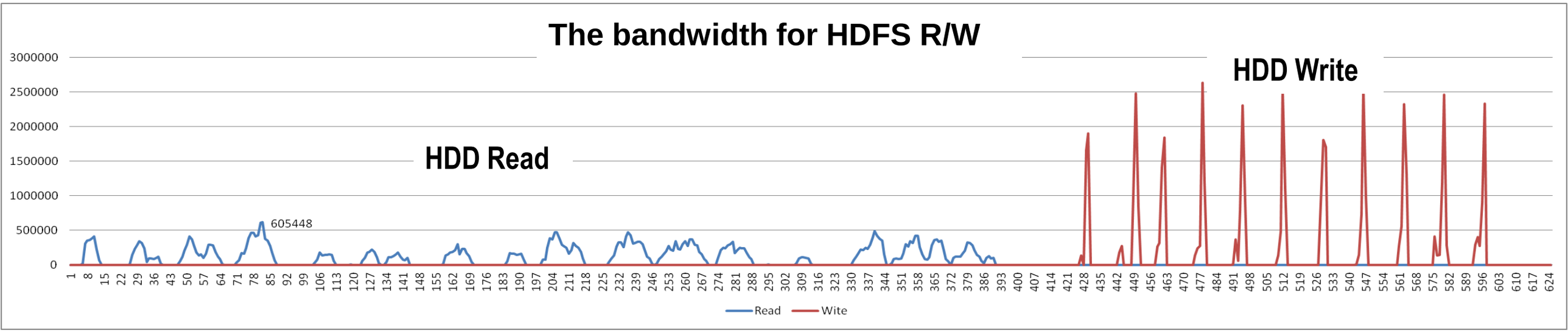


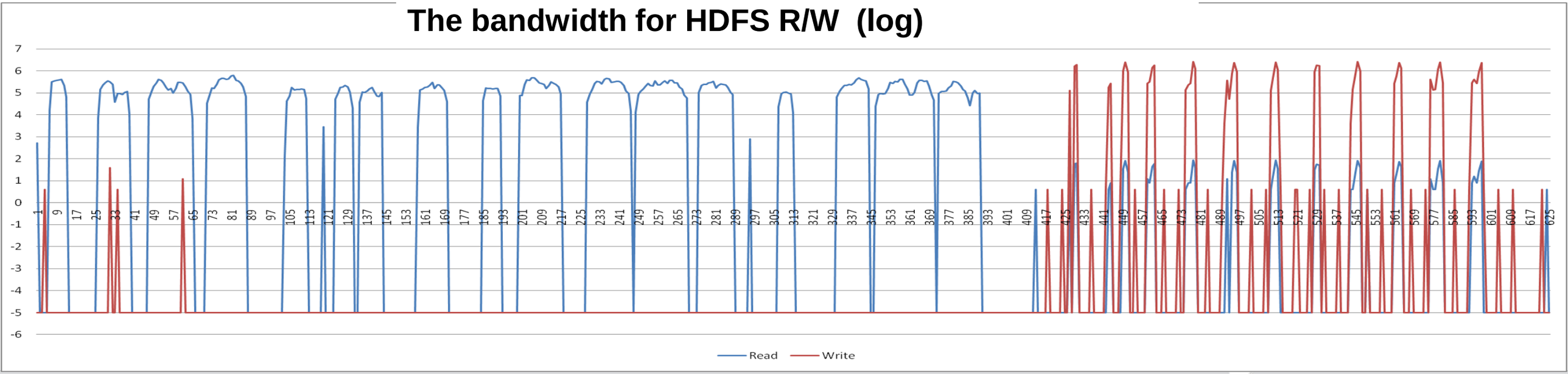




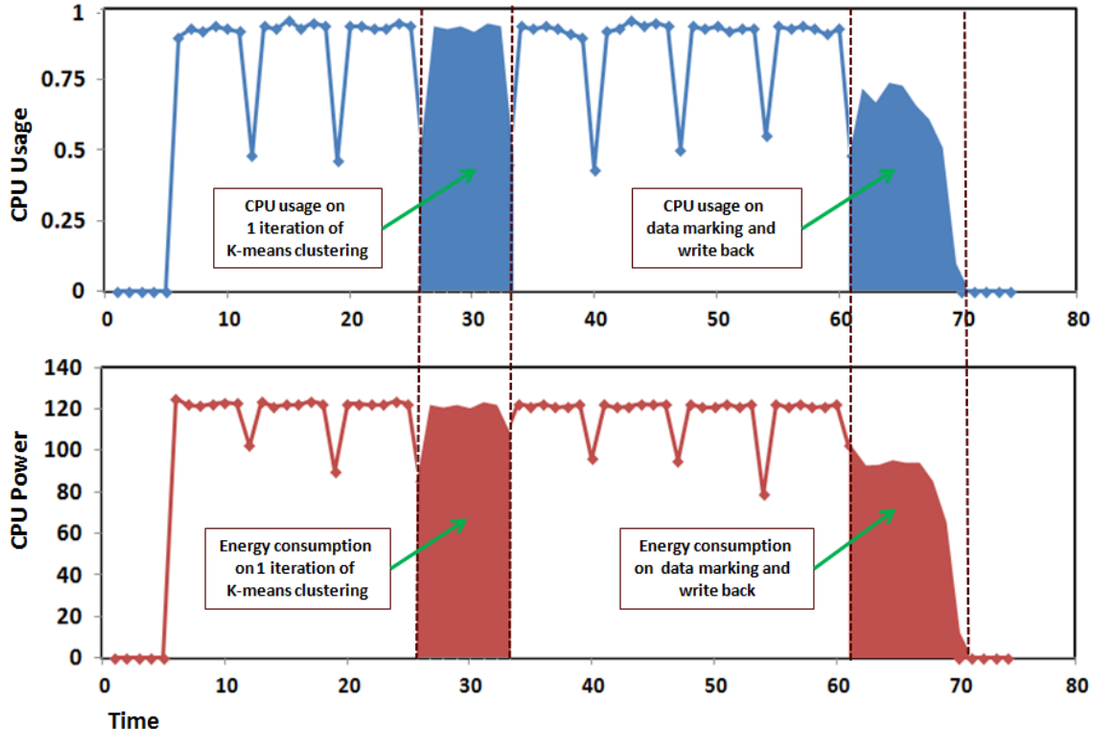


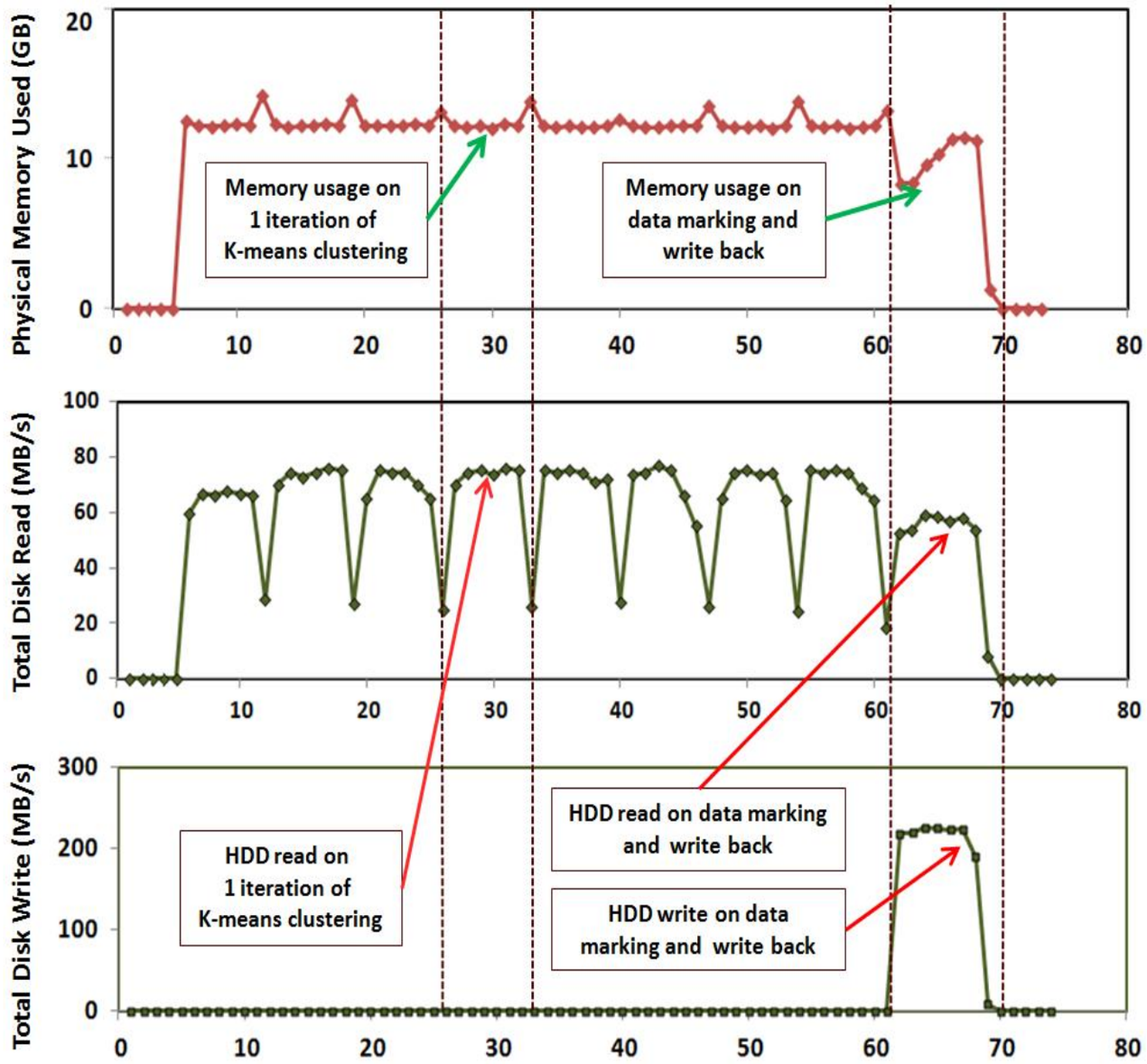
Sorting HDFS HDD I/O Bandwidth against time



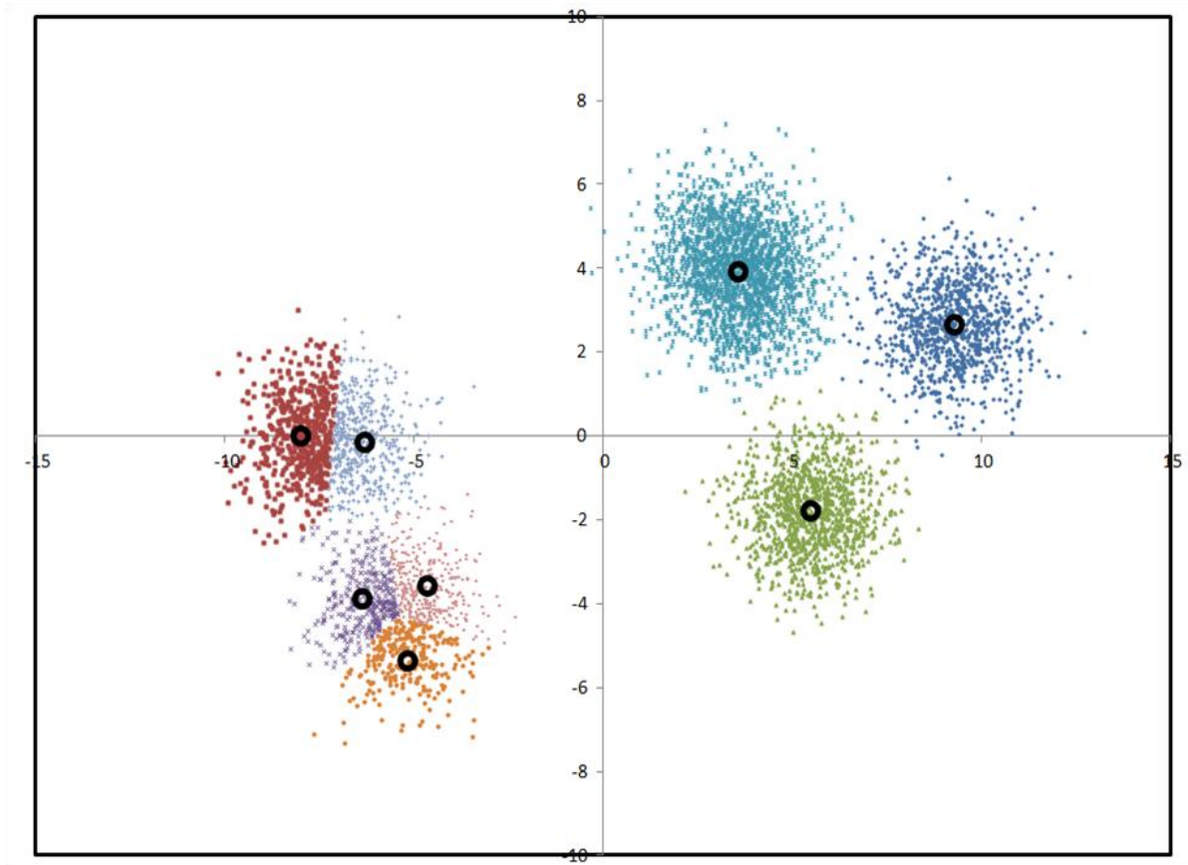


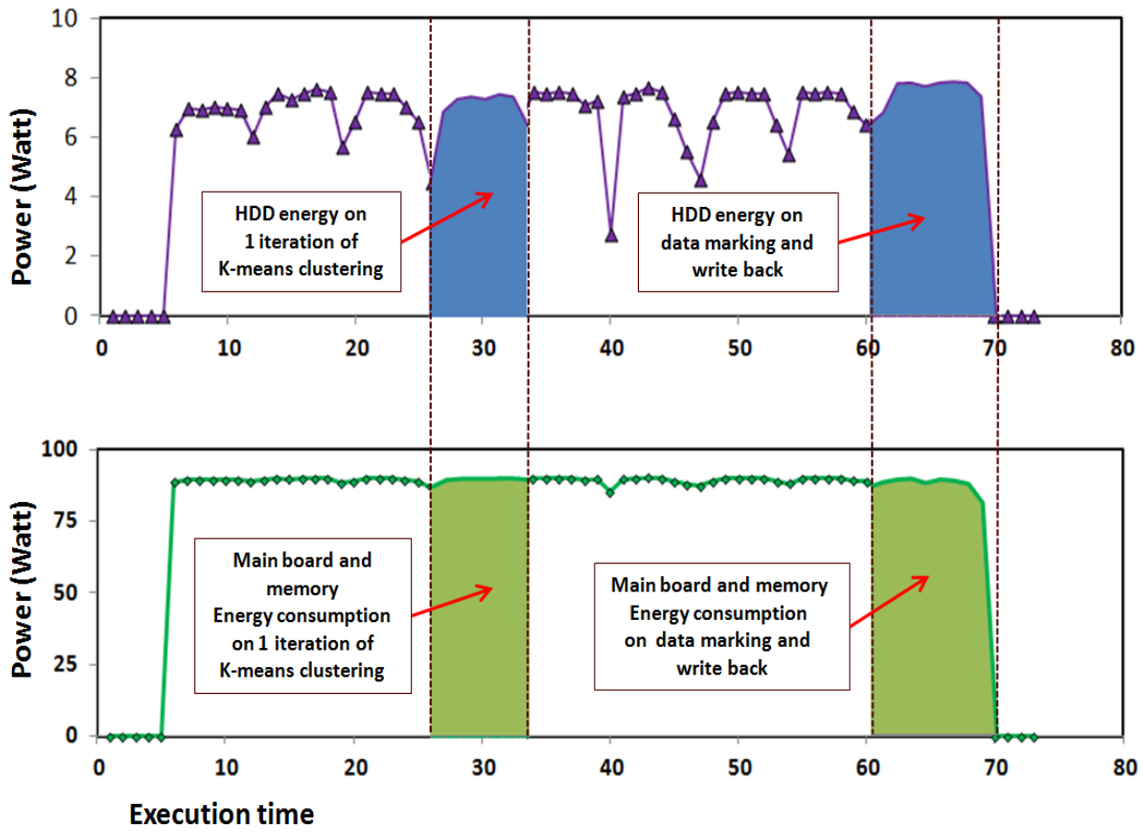
Algorithm Behaviors in K-Means Clustering



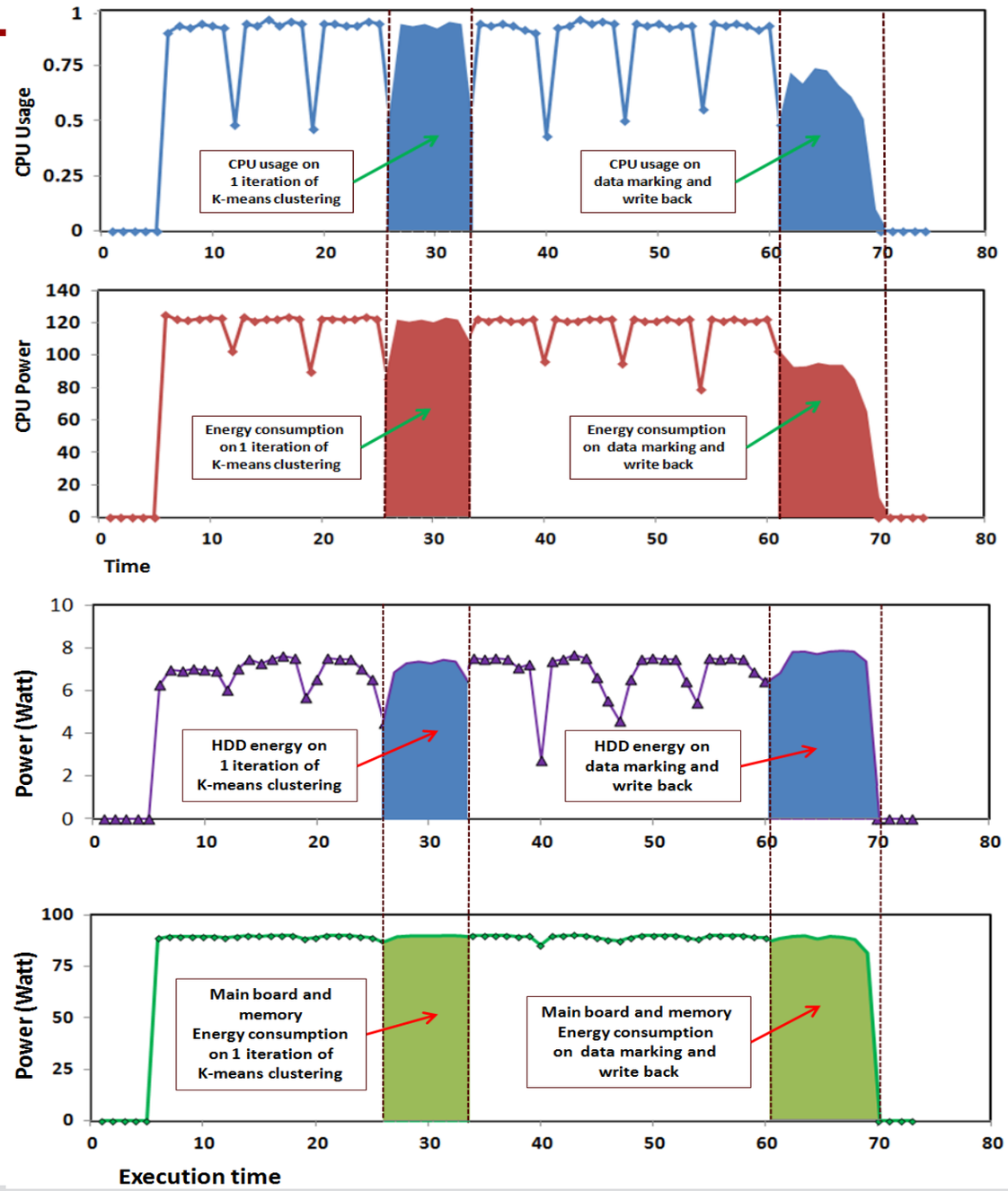


Algorithm Behaviors in K-Means Clustering





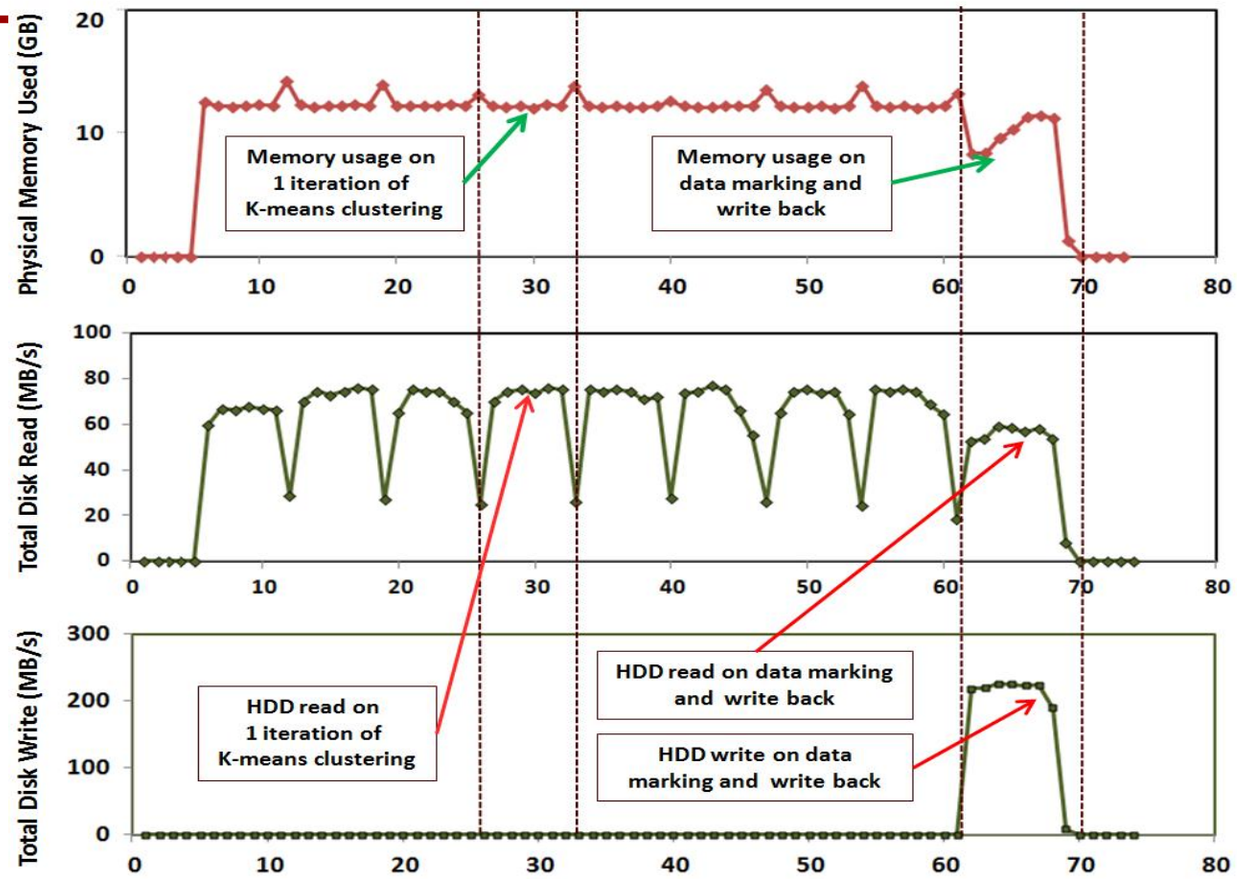
Characters of K-means Clustering



CPU usage and the corresponding power measurement results, when executing K-means clustering operations following the system configuration.

HDD and Main board (include memory) power measurement results, when executing K-means clustering operations following the system configuration.

Characters of K-means Clustering

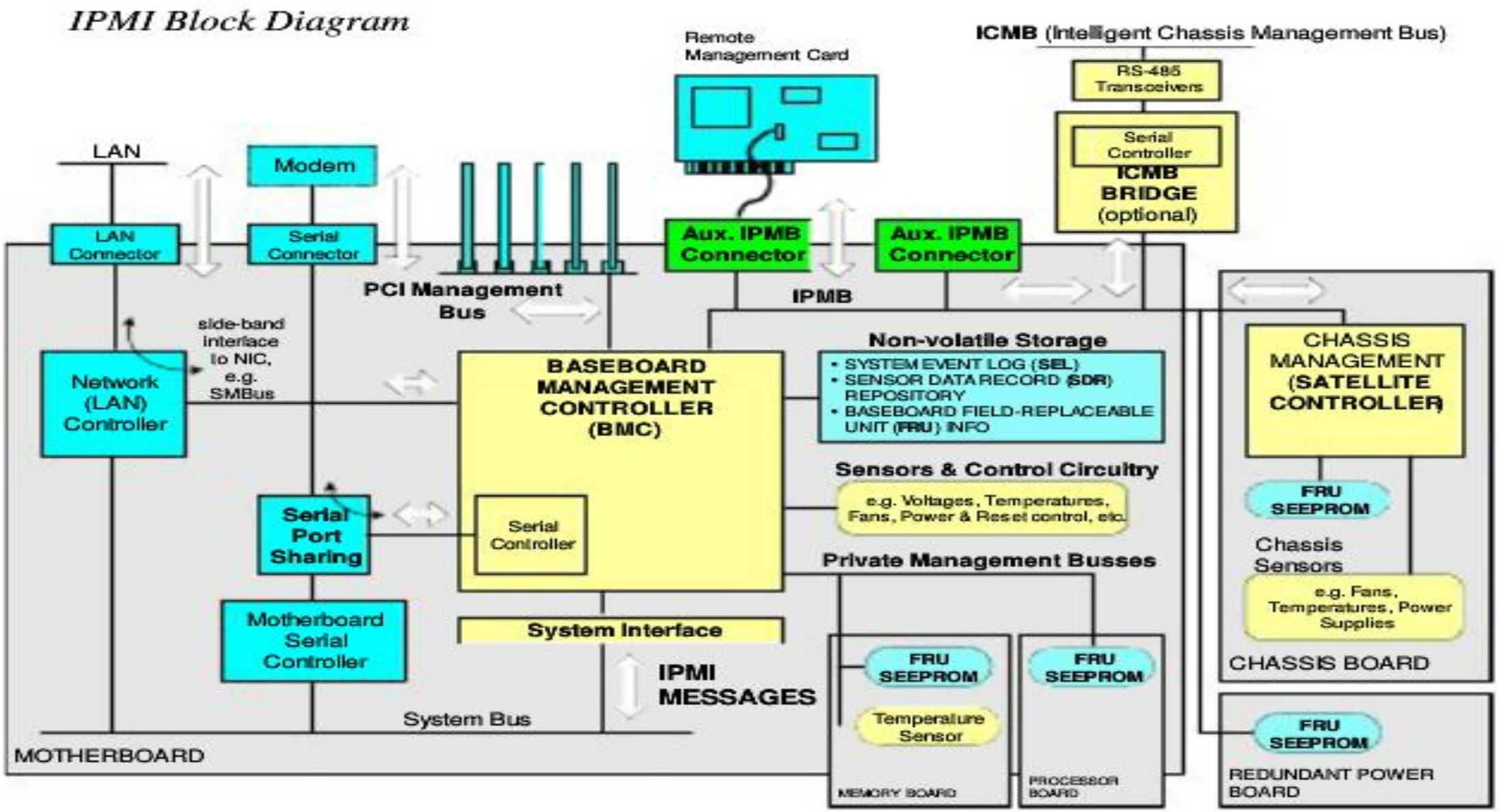


K-Means Job Parameter	Results (Clustering Phase)	Results (Data Writing Phase)
CPU Power (average)	122.21W	98.67W
Main board Power (incl. memory)	72.14W	72.09W
HDD Energy	7.03W	7.17W
Results		
Data Size / Watt	126.04M/Watt	158.13M/Watt

The Physical memory usage, total disk read and write measurement results, when executing K-means clustering operations following the system configuration.

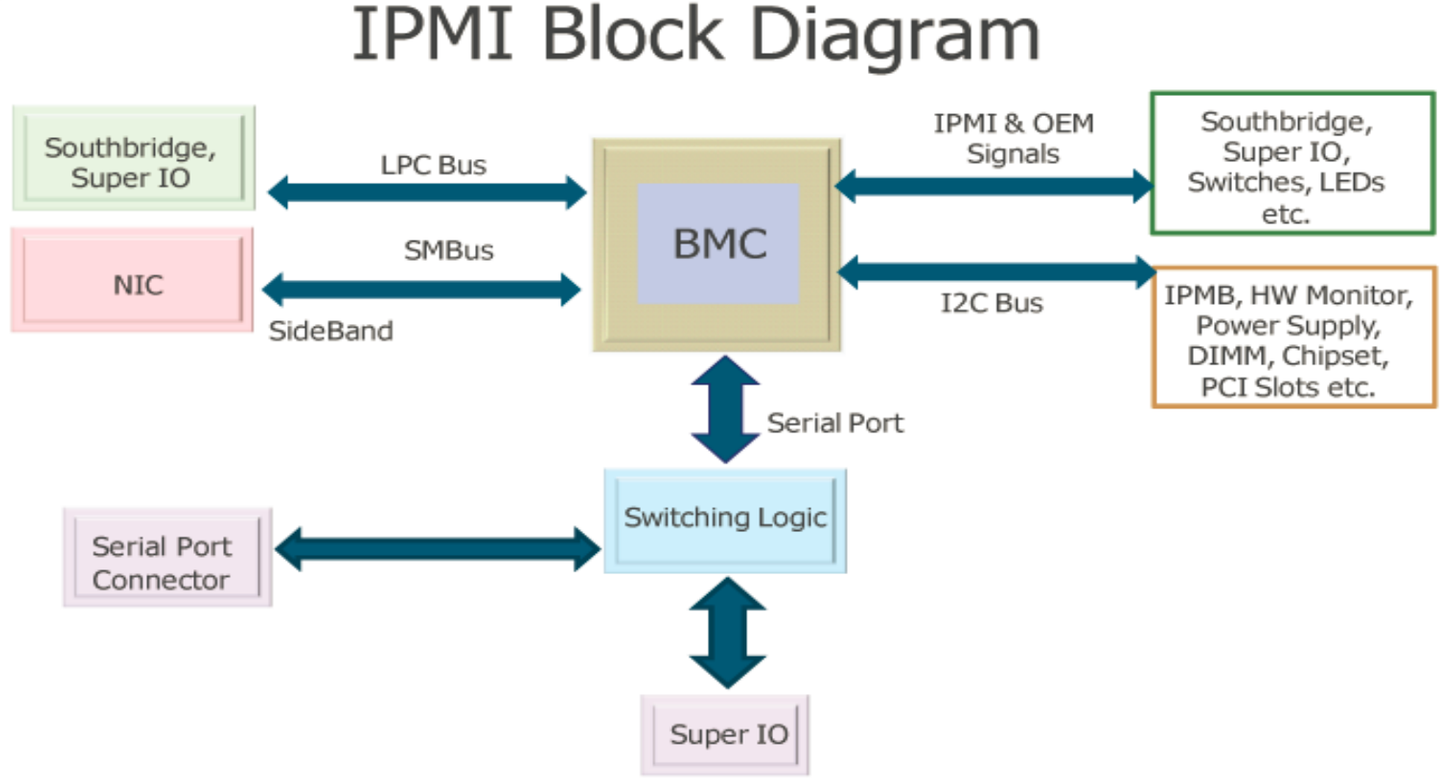
The average of power measurement results for k-means clustering at clustering operation phase and data writing phase. The throughput and power performance of the target platform is concluded.

IPMI (Intelligent Platform Management Interface)



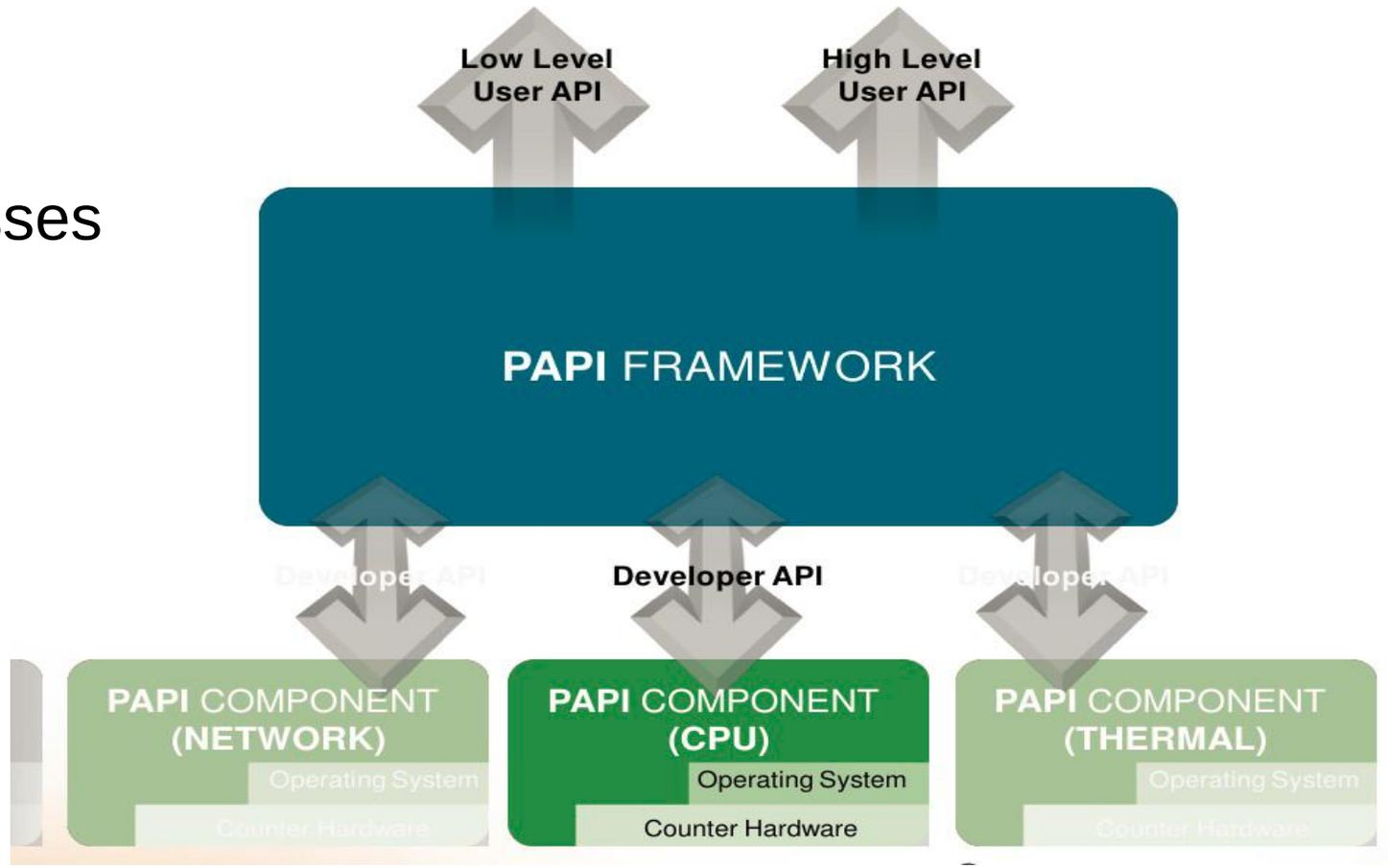
IPMI and BMC

1. Measure energy consumption and other components
2. Baseboard Management Controller (BMC)
   - IPMI
   - On chip averaging
   - Higher sample rate



Performance Application Programming Interface (PAPI)

1. Read special registers (MSR)
2. Performance counter hardware
3. Measure
   - Energy and
   - Flops, cycles
   - Memory access, cache misses
   - Ivy bridge 11 counters



Discussion

- **Kept the Hadoop settings constant.**
- **The Hadoop scheduler which runs at the cluster level can be optimized to a varying compute node configurations**
- **Test the sensitivity of performance for different HW settings**
  - Only the relative results matter.
- **Test performance - Consider power density or compute density.**
  - The recommendation taking density into account might change.

Conclusions

- A general approach is introduced for quantitative estimation and analysis on Hadoop clusters for big data computing.
- The workload characteristics have been analyzed. Based on that, performance parameters are captured by measuring the power of each component in a PE on real system.
- The Hadoop PE's performance feature can then be analyzed and concluded.
- Performance of a program with same computational characters of any size that executing on the PE can be predicted based on the performance feature.
- Finally the approach is validated by measuring real computation on the target hardware.
- Future work The analysis method can be refined to enhance its precision by including more components, and based on it the performance parameters can be tuned for obtaining the best performance for given problems.

Thank you

www.huawei.com