“GFlink: An In-Memory Computing Architecture on Heterogeneous CPU-GPU Clusters for Big Data”

“Contributions/Findings/Conclusions”

- To process advance analytics applications using high computational power of GPU (graphics processing unit) GFlink (Graphics Flink), “an in-memory computing architecture on CPU-GPU clusters” is proposed.
- Defined challenges faced in communication between GPU Kernels and CPU’s JVM and created an effective communication approach, a three-stage pipelining execution method.
- A programming architecture built on Flank’s prototype, i.e., GDST (Dataset based on GPU), masking programming complications in GPUs is advised.
- Extensive research indicated that GPUs large computational potential can be resourcefully exploited using GFlink.

Technology Insights

- Computer programmers can develop GDST (GPU based Dataset) objects called as G-Struct. Using G-Struct, developers can easily align the data bringing easiness for programmers and highly cultivating the performance of the system.
- Decreasing the time interval of data transmission amongst GPUs device memory and main memory is a significant performance boosting technique.
- ‘CUDA kernels’ functioning on GPU framework is only called by host software applications/libraries developed in Python and C.
- GFlink is well-suited in ‘run-time’ as well as in ‘compile-time’ of Flink and is highly compatible with “distributed file system”.

My Key Insights

- GPUs are not only used for general-purpose computing for computer graphics but also GPGPU architecture is a kind of parallel processing between one or more GPUs and CPUs that analyses large amount of data.
- CPUs works on limited cores which processes data serially with few threads at a time. On the other hand, GPUs works on large number of simple cores allowing parallel computing by processing data with thousands of threads at a time
- A CPU-GPU hybrid data analytics system can not only used in executing large number of SQL queries but also used in advance analytics and data mining operations.
- Sharing GPUs in the cluster improves the performance of SQL query execution.

“Improving Utility of GPU in Accelerating Industrial Applications with User-Centered Automatic Code Translation”

Contributions/Findings/Conclusions

- The paper proposed an “automated CPU-GPU source translation framework (GPSME)” for inexpert operators who can exploit computational ability of GPU while driving wide-ranging small enterprise applications.
- Naïve and inexpert users are given a web service platform for calling resource translator with ease.
- A detailed performance assessment of the GPSME architecture for wide-ranging enterprise applications is completed.
- GPSME system accelerates real time advance analytics applications with four times the computation speed used before.

Technology Insights

- Mint, a user-friendly “CPU C to GPU CUDA code translator”, is used.
- With the source code translation, MINT increases the system performance to upto ten times.
- Tools approaching directive methodology are useful in handling the complicated CPU algorithms because annotations can be added to the source code of the CPU.
- Most inexpert GPU users look for actual time saved while processing data for their advance applications rather than high acceleration ratio of GPU/CPU.

My Key Insights

- The notion of parallel computing relies on segregating complex problems into smaller ones before distributing those between several GPU processors.
- GPU users provide original CPU code from their general applications which then is updated by the users for processing using GPSME system. The system generates the GPU codes.
- GPSME architecture is more reliable and practical for real time advance applications then the traditional CPU-GPU architecture.
- The performance of the GPSME system is judged mainly by Acceleration.

Name	Tanmay Bagla
Student ID	19300702
Stream	MSc Data Science
Course ID	CS7NS1
Date	13-09-2019