Hardware Component as a Service (HCaaS): Physical Hardware on Demand

Tanvir Rahman
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
tanvir.rahman1@g.bracu.ac.bd

Tanjid Ahmed
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
tanjid.ahmed@g.bracu.ac.bd

Sabbir Hossain
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
md.sabbir.hossain1@g.bracu.ac.bd

Humaion Kabir Mehedi
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
humaion.kabir.mehedi@g.bracu.ac.bd

Mustakin Alam
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
md.mustakin.alam@g.bracu.ac.bd

Annajiat Alim Rasel
Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—The idea of distributed computing represents a structure in which the components required to achieve a particular system are distributed among multiple computers or nodes. Furthermore, the structure of distributed computing systems requires a master node to perform work distribution and load balancing. While, in cloud computing the whole infrastructure is exposed over the internet through an abstraction layer. The abstraction layer however, introduces an incompatibility between connecting a hardware component with a cloud infrastructure and directly accessing it on a client machine in the system space without a permanent investment.**

For now we are describing an approach through which hardware components can be made available to consumer level by expanding the idea of distributed computing without breaking cloud computing constraints. The fourth layer of cloud computing provides Infrastructure as a Service (IaaS) in which the physical components are at a distance from the consumer and an interaction layer is exposed to the client. This makes it harder to connect additional hardware components to the cloud infrastructure. Also, the consuming device cannot be benefited from the components available in the cloud. With the help of distributed computing methodology, HCaaS will enable the cloud providers to sell hardware components on demand and let the consumers connect any custom hardware to their local machine and make it available in the cloud through kernel level integration. Therefore, in this paper we are focusing on the abstraction layer living in the kernel to make any hardware available in the system space while in reality that component is not connected with the machine through copper conductivity.

Brief: With cloud computing, sharing resources across nodes or computers became a norm. But not every resource can be shared over the internet analogously. The idea of the internet evolved when HTML document file sharing over the network became more popular and people find it easier to share files over the internet using the same HTTP protocol through which HTML documents are served. Later in our service other protocols like file, ssh, SMTP have been developed to further extend the idea of sharing documents but in other formations. HTTP itself is enough to transfer not only Hyper Texts but also other mime types over the network using the good old TCP protocol. But the fundamentals of TCP suggest an abstract layer of packet management to actually send and receive streams of data over the network. Therefore, an easy to implement network API has been made available to consumer level operating systems that enable them to communicate with servers without having to worry about the packet management.

Since, storage sharing requires nothing but carrying the file I/O operations analogously over the network and streams of data can easily be shared over using HTTP protocol, any software requiring a file that is not available in the local system can request the chunks of the file through HTTP requests and rest will be handled by the underlying operating system. This implementation has made file sharing using cloud computing concept much easier. But unlike UNIX not everything is a file. Communication between a software application and a hardware component is handled by a layer called the kernel. Kernel is responsible for fulfilling the request made by software to a specific hardware device by creating a communication bridge between them. Splitting algorithms to make a master slave distributed system is a practice that has been proven to be working for most of the cases. But the splitting algorithm involves complex analogy and investment of a large amount of time. There is no generic rule to split any algorithm across worker nodes to make it work in parallel. Thus, system designers came to the conclusion that splitting data is much easier than splitting algorithms.

Therefore, if a computer game requires a complex equation of physics to be solved by a GPU device it can not send the instructions over the network and get the result as a response. The only feasible solution to this problem is running the game completely where the GPU is physically connected to the system and streaming the video data over the network to the client. In the case of sharing a GPU device it is virtually impossible as of now to share only the device but not the system as a whole. To make a hardware device compatible with a software which is physically distant from the component requires special treatment at the software level which breaks the constraints of cloud computing.

However, we are offering an approach through which a hardware component can be shared over the network without sharing the whole system and without any modification in the consumer application level.