*Paper Critique – Week 13*

**Name: PHAM CONG VINH Student ID: 2019711010**

**Ceph: A Scalable, High-Performance Distributed File System**

This paper present about a file system that support the extreme workloads of high performance system such as supercomputer or big data center.

In this context, in which workload may include such extreme cases as tens or hundreds of thousands of hosts concurrently reading from or writing to the same file or creating files in the same directory, hard disks are replaced with intelligent object storage devices (OSDs) which combine a CPU, network interface, and local cache with an underlying disk or RAID. The failure, replacement, increment of OSD is inevitable and traditional storage media as HDD can't cope with these situation.

The author proposed a solution for this problem is distributed file system. The Ceph file system has three main components: the client, a cluster of OSDs (which collectively stores all data and metadata) and a metadata server cluster (which manages the namespace). The philosophy design of Ceph is separating the metadata management and data will eliminate the bottleneck of centralized metadata management which will increase the scalability (performance and durability) of storage my leveraging the capability of OSD.

The paper will be more interested if author evaluate how good the performance of distributed file system Ceph on small-scale environment (couples OSD with HDD only).

**Barrier-Enabled IO Stack for Flash Storage**

The problem which author stated in this paper is that the limitation in throughput of Flash based storage medium caused by the unnecessary constraint inherited from the old legacy methodology enforcing the write order – "Transfer-and-Flush" in which dispatching the following request only after the data block associated with the preceding request is completely transferred to the storage device and is made durable.

The author proposed a new file system that overcome this limitation bottleneck by eliminating the Transfer-and-Flush overhead with Barrier-Enabled IO stack. The barrier-enabled IO stack consists of the cache barrier-aware storage device, the order-preserving block device layer, and the barrier enabled file system. By this proposed file system, host application can issue the following request before the preceding request is serviced and yet the IO stack can enforce the storage order between them. The author modified the Linux's EXT4 file system in order implement their file system then benchmark by SQLite workload and evaluate their solution.

The strength of the paper is that author conducted the experiment on both mobile storage to the high performance Flash storage and argue about the result. This obviously convinced audience that their solution is a useful foundation in designing a new IO stack for the Flash storage.

The paper will be more interested if there were some benchmarks with workload that required high bandwidth such as reading high resolution video file.