2.1 Video-on-Demand Systems

In this paper, we focus on two major modules of a VOD system: a transcoding module and a video streaming module. In a production environment there may be other modules: ~~like~~ an advertising module, a recommendation module, etc. ~~We~~However, we consider only the transcoding ~~module~~ and streaming ~~module~~modules for two reasons. First, they are necessary, whereas ~~while~~ the other modules are not. Second, distinct characteristics of the two modules already exhibit the complexity in our choice of ~~the~~ underlying platform.

When a new video is sourced (e.g. uploaded by a user), it is~~we~~ transcoded and preserved in storage, once and for all. Each time a user requests ~~for~~ a video, the transcoded version is streamed to the client. Frequently accessed videos may be cached to reduce the response time.

註解 [Editor1]:
Golden English Editing
Mathematics & Computer Science
Computer Science
Sample of work

2.2 Wimpy Nodes with SSDs

Wimpy nodes are ~~P~~platforms with CPUs of low~~less~~ computing power ~~mean wimpy nodes~~.[1][16][9] Compared with powerful processors (e.g. Xeon), wimpy processors (e.g. ARM-based, Atom) feature low power draw and low ~~dollar~~ cost. For example, an Intel Atom N570 has a TDP[1] of 8.5W.[12] The value of an Intel Xeon E5620 is 80W,[5] one order of magnitude more than the Atom processor.

Solid-state-drives (SSDs) outperform traditional harddisk drives (HDDs) with low random read latency (hundreds of microseconds compared to HDD~~'~~s' several to tens of milliseconds[2]), and low power consumption. The defects of SSDs are poor write performance and limited endurance (about 5 years under intensive ~~accesses~~accessing[6]). Previous work demonstrates that wimpy nodes with SSDs are energy-efficient on some kinds of data-intensive jobs ~~like~~such as key-value stores and sorting,[1][16][2] ~~and~~but are not good at handling ~~some~~certain complex jobs ~~like~~such as database applications.[9][10]

2.3 Module Mapping Modes

---

1 Thermal design power

2 http://en.wikipedia.org/wiki/Solid-state_drive

In a heterogeneous data center, there are ~~nodes in~~ different types~~. For example, there may be~~ of nodes such as Xeon servers and Atom-~~-~~based servers. Applications like VOD systems may contain multiple modules. There ~~would be~~are many choices ~~on~~of how to map modules of an application to available nodes in a data center. We classify ~~them~~these into three *module mapping modes*, as is depicted in Figure 1. In the **homo**(geneous) mode, different modules are mapped to different instances of the same type of node. In the **hetero**(geneous) mode, different modules are deployed on nodes of different types. In the **union** mode, different modules are multiplexed on a single node.

Intuitively, the streaming module of a VOD system is I/O-bound and performs well on wimpy nodes with SSDs. The transcoding module is CPU-bound and should be deployed on powerful servers. Furthermore, as the two modules stress I/O components and processors respectively, multiplexing the two modules on a server with SSDs may be a better choice. In this paper, we deploy VOD systems on combinations of various processors and disks, in different mapping modes. The whole process ~~gives~~provides some insights ~~on~~into finding an optimized platform configuration for a given application like VOD systems. Our experience is that actually running the application on each type of node is necessary.

註解 **[CL2]:**

CHECK: Perhaps you could replace this with something like “The experience of this research team” in order to be more specific.

3 Evaluation Metrics

Our experiment spans three dimensions in a configuration. First, we choose two VOD applications, one in Windows and one in Linux. Second, the hardware platforms are combinations of three types of processors and three types of disks. The number of disks is also varied. Third, we use three module mapping modes. We use **power performance ratio** and **cost performance ratio** to quantify the effectiveness. ~~We~~In addition we run each module of the VOD system on each hardware platform. Our experiment also includes running the two modules simultaneously on the same node. During the run we measure the power and performance, and calculate the cost. Given these data, we can then~~may~~ derive the above two ratios of running a module on a specific platform. To compare between the mapping modes, we set up a universal Service Level Agreement (SLA) and calculate the power and cost on achieving that SLA, in each mapping mode. This section explains how we quantify ~~the~~ performance, and describes the power and cost models. Details about the software and hardware we use can be found in Section 4.

3.1 Performance Metrics

A typical SLA may require the VOD service provider handle *X* concurrent streaming requests while ~~transcode~~transcoding *Y* hours of videos per second. We quantify ~~the~~ performance as *the capability of finishing the SLA*.

Performance of the streaming module on a platform is quantified as the number of simultaneous streaming requests it supports. We assume the quality-of-service (QoS) requires no less than 95% of videos being streamed ~~are~~to be *good* playbacks. The playback of a video is deemed good if the actual playback length is within 5% error. The videos being streamed are identical copies of a 15-minute-long video.

註解 [CL3]:

CHECK: It is perhaps unclear what you mean here. Do you mean when the playback length differs by 5% or less from the length of original video? Please clarify this.

3.2 Scaling the Models

The power and cost models ~~go~~are for running a single module on individual nodes. We assume power and cost scale linearly with the number of nodes. Previous work demonstrates that due to heavy communications, this is not the case for database workloads~~, due to heavy communications~~.[10] ~~Anyway~~Nevertheless, scaling proportionally is probably true for VOD systems because neither of the modules requires heavy inter-process communications. Hardly any inter-node communications are involved either~~. This makes~~, making it reasonable to assume power and cost goes up approximately linearly with the number of nodes involved.

註解 [CL4]:

CHECK: Consider replacing this with “appropriate”.