A Novel Nested Qos Model for Efficient Resource Usage in Storage Servers

Hui Wang* Peter Varman
hw5@rice.edu pjv@rice.edu
Rice University, USA

The increasing popularity of storage and server consolidation introduces new challenges for resource management, capacity provisioning, and guaranteeing application performance.Typical Service Level Objectives (SLOs) provide performance guarantees in terms of throughput (IOPS) or response time limits (ms). The bursty nature of storage workloads (where instantaneous arrival rates significantly exceed the average) implies a large gap between peak and average resource requirements in meeting response time bounds, leading to low overall server utilization and high cost. This situation is driving the development of elastic QoS models that allow clients greater flexibility in adopting SLOs tailored to their workload characteristics and performance requirements, while allowing the service provider opportunities to optimize provisioning and scheduling decisions.

In this paper we propose a novel Nested QoS service model to provide flexible QoS performance guarantees to clients. The model formalizes the empirical observation that a disproportionately large amount of resources are used to handle the small tail of badly behaving requests. The amount of server resources required can be reduced significantly by using workload decomposition to identify these requests dynamically and scheduling them with less stringent response time requirements. The Nested QoS model provides a formal (but intuitive and auditable) way to specify the notion of graduated QoS where a single client's SLOs is specified in the form of a response time distribution rather than a single worst-case guarantee.

Figure 1 shows the framework of our Nested QoS service model. The performance SLOs is determined by multiple nested classes. Each class is characterized by a traffic envelope and a response time limitation. For example, in the 3-class Nested QoS mode, all the requests in the workload that satisfy the Class 1 envelope have a response time guarantee of $\delta_1$ms; the requests that satisfy the less restrictive Class 2 envelope arrival constraint have a latency bound of $\delta_2$ms, while those conforming to the Class 3 envelope arrival bound have a latency limit of $\delta_3$ms.

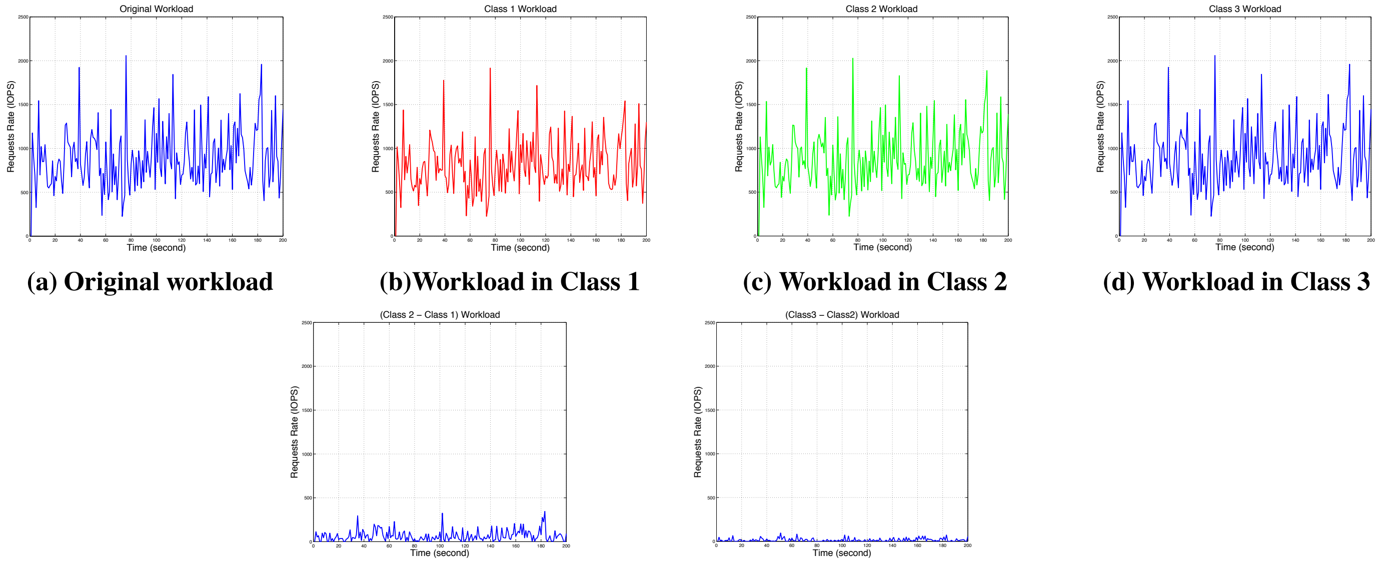
The Nested QoS model consists of two components: *request classification* and *request scheduling*. The former places requests into different classes, based on the traffic envelope. The latter schedules requests across all classes based on class tag and other information. The detail about how to set the traffic envelope, how to estimate capacity required and how to schedule the requests will be discussed in following full paper.

We have implemented the Nested QoS model in a process-driven system simulator and evaluated it using several block-level storage workloads from UMass Storage Repository. Table 1 compares the capacity required by the workloads for the nested and single-level QoS models (which is a single worst-case guarantee of δ_1). The capacity required for Nested QoS is significantly less (several times smaller) than that for a single-level QoS, while the service seen by the clients is only minimally degraded. Table 2 shows classification results based on the traffic envelops for the three classes of each workload. As shown, in each case a large percentage (92%+) of the workload meets the 5ms response time bound, and a tiny 0.1% or less requires more than 50ms. Figure 2 (b) - (d) show the the *Exchange* workload decomposed into the three classes, while Figure 2(e) (respectively (f)) shows the portions of the workload in class 2 (respectively 3) but not in class 1 (respectively 2). By varying the envelop parameters, different tradeoffs in capacity, QoS, and (implicitly) cost can be obtained.

Our model provides several advantages: (i) significantly reducing resource provisioning over usual SLOs specifications while still providing comparable QoS for clients, (ii) making it possible to estimate the server capacity (as verified in experiments) (iii) providing more flexible SLOs to clients with diverse performance/cost tradeoffs, and (iv) providing a conceptual structure of SLOs in workload decomposition.

Our work continues to explore different implementations, more accurate capacity estimation, relating workload characteristics with nested model parameters, semantic restrictions on decomposition, scheduling multiple decomposed workloads on a shared server, and a Linux block-level implementation.

*** Student Author and Presenter**



(a) Original workload **(b)Workload in Class 1** **(c) Workload in Class 2** **(d) Workload in Class 3**

(e) Workload in (Class2-Class1) **(f) Workload in (Class3-Class2)**

Figure 2. Decomposition of workload into different classes.

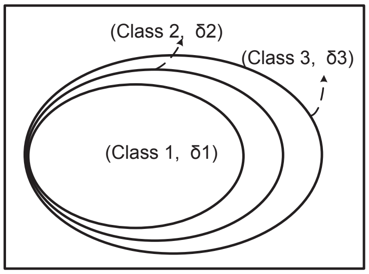


Figure 1. Nested Qos Framework for storage workload

WORKLOADS	Nested-QoS	Single-Level QoS
WebSearch1	751	2400
WebSearch2	713	2100
FinTrans	660	3000
OLTP	456	1400
Exchange	6710	16400

Table 1. Capacity (IOPS) required for Nested-QoS and Single-Level QoS for each workload

WORKLOADS	δ_1	δ_2	δ_3
WebSearch1	93.9%	99.5%	100%
WebSearch2	94.2%	99.8%	100%
FinTrans	92.6%	97.4%	99.92%
OLTP	93.7%	99.8%	100%
Exchange	93.8%	99.3%	99.99%

Table 2. Percentage of workload in each class with response time limits 5, 50 and 500 ms