2015 IEEE 35th International Conference on Distributed Computing Systems

# Multi-Tenant Latency Optimization in Erasure-Coded Storage with Differentiated Services

Yu Xiang
*Department of ECE*
*George Washington University*
*DC, USA*
*xy336699@gwmail.gwu.edu*

Tian Lan
*Department of ECE*
*George Washington University*
*DC, USA*
*tlan@gwu.edu*

Vaneet Aggarwal
*School of Industrial Engineering*
*Purdue University*
*West Lafayette, IN, USA*
*vaneet@purdue.edu*

Yih-Farn R Chen
*AT&T Labs-Research*
*Bedminster, NJ, USA*
*chen@research.att.com*

The effect of coding on content retrieval latency in data-center storage system is drawing more and more significant attention these days, and customizing elastic service latency for the tenants is undoubtedly appealing to cloud storage, but it also comes with great technical challenges: due to the lack of analytic latency models for erasure-coded storage, most of the literature is limited to the analysis of average service latency, e.g., [1], [2], having assumptions like homogeneous files, exponential service time distribution [3], fixed erasure codes [4], which is unsuitable for a multi-tenant cloud environment where each tenant has a different latency requirement for accessing files in an erasure-coded, online cloud storage. Optimizing differentiated service delay in an erasure-coded storage system is an open problem. This work considers an erasure-coded storage with multiple tenants and differentiated delay demands, studies two types of service policies, non-preemptive priority queue and weighted queue, quantifying service latency of these policies, propose a novel optimization framework that provides differentiated service latency to meet heterogeneous application requirements in cloud storage.

We assume that files are divided into 2 service classes, one for delay-sensitive files and the other for delay-insusceptible files. The file requests arrives as a Poisson process. For a given placement for each file, and erasure-coded parameters $(n, k)$, we will use probabilistic scheduling to select $k$ file chunks when a file is requested. This scheduling strategy was proposed in [2] and was shown to be equivalent to choosing each of the storage nodes with certain probability, $\pi_{i,j}$.

Next, we describe the two queuing models that are used in this work. Our first policy is modeled as a non-preemptive priority queue. We assign a high priority for delay-sensitive files and a low priority for delay-insusceptible files. There are two sets of queues (high/low priority) at each storage node. We assume that service time distribution for all the storage nodes is the same. A chunk is served from high priority queue as long as there is a chunk in the queue, and a chunk is serviced from the low priority queue only if there is no chunk in the high priority queue, the request which is already in service will not be affected by the later arrival of high priority requests. Weighted queuing apportions service rate among different service classes in proportion to given weights. Tenants with higher weights receive more service rates, while tenants with lower weights can still receive their fair share if the weights are properly balanced. Unlike priority queuing, each server now is able to serve two requests from different classes at the same time, offering different service bandwidth for different classes.

We consider two types of delay in the latency upper bound, *Queuing delay* $\mathbf{Q}_j$ and *Connection delay* $\mathbf{N}_j$. $\mathbf{Q}_j$ is the waiting time a chunk request receives in node $j$ and is determined by service rates and arrival rates of chunk request at each storage node according to our queuing models. We assume that the connection delay is independent of $\mathbf{Q}_j$. Then the latency of a file-$i$ request is determined by the maximum of queuing plus connection delay of the $k_i$ nodes in $\mathcal{A}_i$. Therefore, we have:

$$\bar{T}_i = \mathbb{E}[\max_{j \in \mathcal{A}_i}(\mathbf{N}_j + \mathbf{Q}_j)] \tag{1}$$

The authors of [2] gave an outer bound on $\bar{T}_i$ using its mean and variance as follows.

$$\bar{T}_i \leq \min_{z \in \mathbb{R}} \left\{ z + \sum_{j \in \mathcal{S}_i} \frac{\pi_{i,j}}{2} \left(\mathbb{E}[\mathbf{D}_j] - z\right) + \sum_{j \in \mathcal{S}_i} \frac{\pi_{i,j}}{2} \left[\sqrt{(\mathbb{E}[\mathbf{D}_j] - z)^2 + \mathrm{Var}[\mathbf{D}_j]}\right] \right\}, \tag{2}$$

where $\mathbf{D}_j = \mathbf{N}_j + \mathbf{Q}_j$ is the aggregate delay on node $j$ with mean $\mathbb{E}[\mathbf{D}_j]$ and variance $\mathrm{Var}[\mathbf{D}_j]$.

We notice that the latency bound depends on mean and variance of the aggregate delay, which depends on our queuing models. For priority queuing, we consider two priority queues (high/low priority) for each storage node. We analyze non-preemptive priority queues on each node and obtain the mean and variance of $\mathbf{D}_j$ using variations of *Pollaczek-Khinchine* formula to obtain an upper bound on service latency for each file in the two service classes.

For weighted queuing, each storage node employs a separate queue for each service class. Queuing delay for

1063-6927/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDCS.2015.111

requests of each class on a node depends on the queuing weights since service bandwidth on each storage node is shared among all queues in proportion to their assigned weights. Due to Poisson property of request arrivals, each weighted queue can be modeled as a independent M/G/1 queue whose mean and variance can be found in closed-form.

Using these analysis, we propose a novel optimization framework for minimizing differentiated service latency for all tenants in an erasure-coded storage system.We need to optimize over i) chunk placement ii) access probabilities iii) weights for different files in the case of weighted queues. We formulate latency optimization problem using the queuing models: For priority queues, we propose a two-stage optimization problem as follows. First, we jointly optimize the chunk placement and access probabilities for all files in high priority class to minimize service latency. Then, latency for low priority files are minimized based on existing traffic generated by high priority files. Let $\hat{\lambda}_k = \sum_{i \text{ is file of priority class k}} \lambda_i$ be the total arrival rate for high priority requests, and thus $\lambda_i/\hat{\lambda}_k$ is the fraction of file $i$ requests among the class $k$ priority files.

$$\min \sum_{i \in \mathcal{R}_k} \frac{\lambda_i}{\hat{\lambda}_k} \tilde{T}_{ik}$$

We can see this optimization problem is a mixed integer optimization due to we have fixed $n$ servers to select for chunk placement for each request, so it is hard to solve in general, thus we propose to break this problem into 2 sub-problems: placement and scheduling. We consider the sub-problem for optimizing scheduling probabilities and recognize that for fixed chunk placements, which is convex in $\pi_{ij}$. We also show that the placement sub-problem can be equivalent to a bipartite matching problem and can be efficiently solved by Hungarian algorithm. For weighted queuing we consider a joint optimization of all files in different service classes by minimizing a weighted aggregate latency. Let $\hat{\lambda}_k = \sum_{i \in \mathcal{R}_k} \lambda_i$, $\hat{\lambda} = \sum_k \hat{\lambda}_k$ and and $\tilde{T}_{ik}$ be given by the upper bound.

$$\min \mathcal{C}_1 \tilde{T}_1 + \mathcal{C}_2 \tilde{T}_2$$

$$\tilde{T}_k = \frac{\hat{\lambda}_k}{\hat{\lambda}} \sum_{i \text{ is file of class } i} \tilde{T}_{ik}$$

Similarly we have a fixed $(n, k)$ erasure code applied, problem is an mixed integer optimization and is broken into three sub-problems: (i) a weight sub-problem for optimizing service bandwidth among different queues by choosing weights, (ii) a scheduling sub-problem and (iii) a placement sub-problem.

We first recognize the scheduling sub-problem is convex. As for the placement problem we again cast it into a matching, similar to the one proposed for priority queuing. Also, we show that the weight sub-problem is convex with respect to weights.

To validate our theoretical analysis and joint latency optimization for different tenants, we provide a prototype of the proposed algorithms in *Tahoe*, which is an open-source, distributed file system based on the *zfec* erasure coding library for fault tolerance. A Tahoe storage system consisting of 12 storage nodes are deployed as virtual machines in an OpenStack-based data center environment. One additional storage client was deployed to issue storage requests. From the experiment results, we first find that the service time distribution is proportional to the bandwidth of the server, which validates an assumption used in the analysis of the weighted queue latency. Further, the experiment results validate fast convergence of our differentiated latency optimization algorithms. We see that our algorithms efficiently reduce latency both with the priority and the weighted queues, and the results from the experiments are reasonably close to the given latency bounds for both the models. Finally, we note that priority queuing could lead to unfairness since the low priority tenants only share residual service rates left over by high priority tenants, while weighted queuing is able to balance service rates by optimizing weights assigned to each service class.

To summrize, relying on a novel probabilistic scheduling policy, this work develops an analytic upper bound on average service delay of multi-tenant, erasure-coded storage with arbitrary number of files and any service time distribution using weighted queuing or priority queuing to provide differentiated services. An optimized distributed storage system is then formalized using these queues. Even though only local optimality can be guaranteed due to the non-convex nature of the problems, the proposed algorithm significantly reduces the latency. Both our theoretical analysis and algorithm design are validated via a prototype in an open-source, distributed cloud storage deployment that simulates three geographically distributed data centers through bandwidth reservations.

## REFERENCES


[1] L. Huang, S. Pawar, H. Zhang and K. Ramchandran, "Codes Can Reduce Queueing Delay in Data Centers," *Journals CORR*, vol. 1202.1359, 2012.

[2] Y. Xiang, T. Lan, V. Aggarwal, and Y. R. Chen, "Joint Latency and Cost Optimization for Erasure-coded Data Center Storage," *Proc. IFIP Performance*, Oct. 2014 (available at arXiv:1404.4975 ).

[3] G. Joshi, Y. Liu, and E. Soljanin, "On the Delay-Storage Trade-off in Content Download from Coded Distributed Storage Systems," *arXiv:1305.3945v1*, May 2013.

[4] S. Chen, Y. Sun, U.C. Kozat, L. Huang, P. Sinha, G. Liang, X. Liu and N.B. Shroff, " When Queuing Meets Coding: Optimal-Latency Data Retrieving Scheme in Storage Clouds," *IEEE Infocom*, April 2014.