The Tail at Scale: How to Predict It?

Minh Nguyen, Zhongwei Li, Feng Duan, **Hao Che**, Yu Lei, Hong Jiang

Department of Computer Science and Engineering

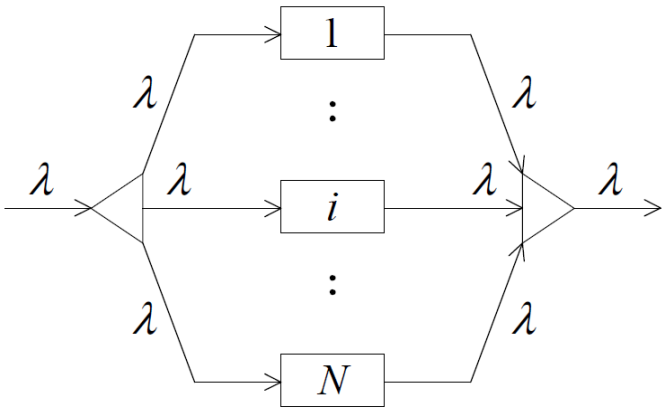
The University of Texas at Arlington

Motivation

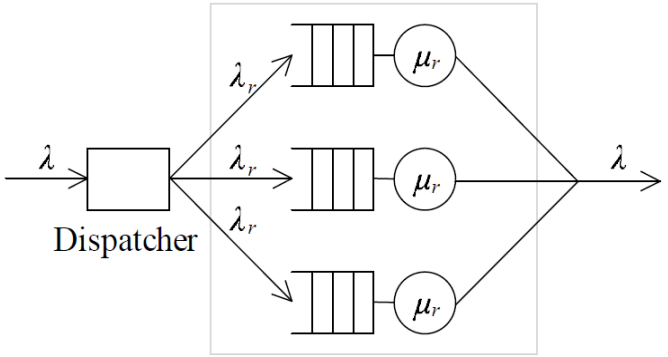
- Today's online-data-intensive (OLDI) applications must meet stringent request tail Service Level Objectives (SLOs) to retain customers
- A request tail SLO is generally expressed in the form of the p^{th}-percentile request response time of x_p milliseconds, e.g., $p=99$ and $x_p = 300$ milliseconds

Motivation (Cont'd)

- OLDI applications are generally scale-out by design, i.e., each request involves task partitioning and merging and the slowest task determines the request response time
- A task subsystem may involve multiple replicated servers for fault tolerance, load balancing, and task tail-cutting by redundant task issuing



A task-partition-merge system with task mapping to parallel subsystems



A task subsystem with one dispatcher and three replicated servers

Motivation (Cont'd)

Question: How to schedule tasks effectively at individual subsystems so that the request tail SLO will be met?

Challenge: There is a missing link between request tail SLO and task performance budgets, in terms of, e.g., mean and variance of task response time

Due to the lack of such a link, the approach taken today to meet tail SLOs is by resource overprovisioning

Outline

- Motivation
- Our design goal
- Prediction model
- Testing results
- Conclusions

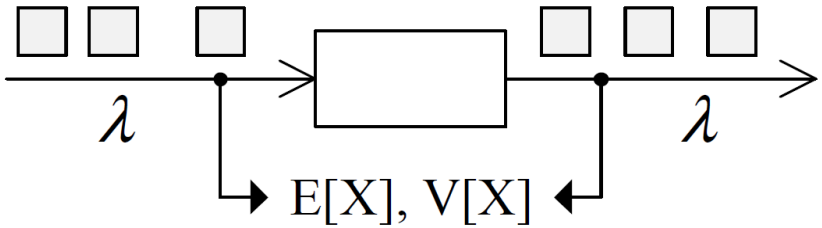
Our Goal

- Establish a link between request-level tail SLO and task response time budgets, independent of the underlying subsystems to be used
- The budgets are related to the low order statistics of the task response time, e.g., mean and variance, which can be measured easily, so that the budgets can be tested quickly

Achieving this goal will allow tail-SLO-aware request scheduling problem to be degenerated into a simple budget testing problem at the task level. Possible use cases are:

- Offline Resource planning
- Online distributed task scheduling

Prediction model



Main Idea:

1. Treat each task subsystem as a black box
2. Seek the possibility of constructing the task-response-time distribution function $F(x)$ as a function of E[X] and V[X] only

Then the extreme value theorem says that the request-response-time distribution function $F_{(N)}(x) = F^N(x)$ => a tail SLO can then be readily expressed as a function of E[X] and V[X]

Prediction model

- It turns out this goal can be achieved in a high load region, say 90% or higher, where the resource provisioning is desirable
- It is based on the central limit theorem of queuing systems:

The response time distribution F(x) will converge to an exponential distribution as the load increases, which is a function of E(X) only

Prediction model (cont'd)

- We postulate that in the high load region of practical interest, the task response time distribution can be adequately approximated as a generalized exponential distribution.
  - Distribution function

$$F_{ge}(x) = \begin{cases} (1 - e^{-\mu x})^{\alpha} & x > 0, \\ 0 & \text{otherwise}, \end{cases}$$

  - Mean and variance

$$\begin{aligned} E[X] &= \frac{1}{\mu}[\psi(\alpha + 1) - \psi(1)], \\ V[X] &= \frac{1}{\mu^2}[\psi'(1) - \psi'(\alpha + 1)], \end{aligned}$$

Prediction model (cont'd)

$$F_{(N)}(x) = \begin{cases} (1 - e^{-\mu x})^{N\alpha} & x > 0, \\ 0 & \text{otherwise}, \end{cases}$$

- The pth-percentile request response time of $x_{p,}$

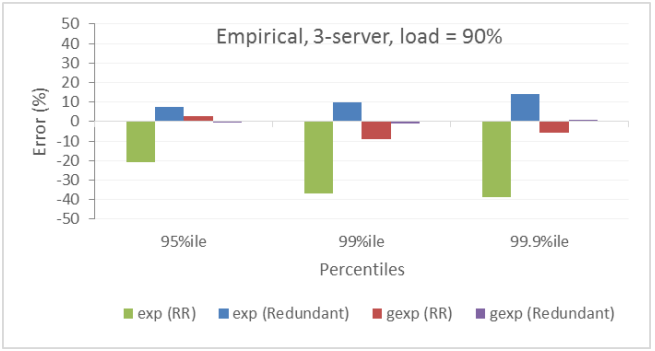
$$x_p = -\frac{1}{\mu} \log \left(1 - \left(\frac{p}{100} \right)^{\frac{1}{N\alpha}} \right)$$

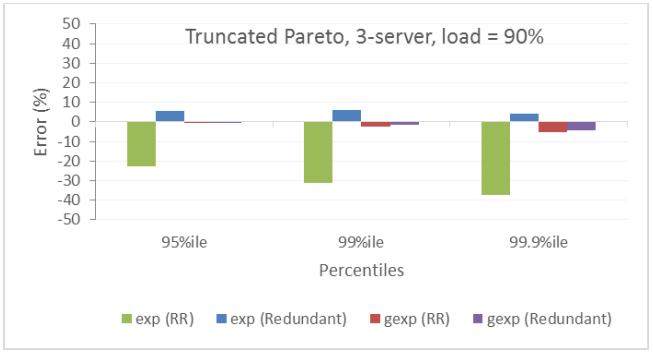
- A link between any given tail SLO in terms of x_p and p, and task budgets E[X] and V[X] is established.

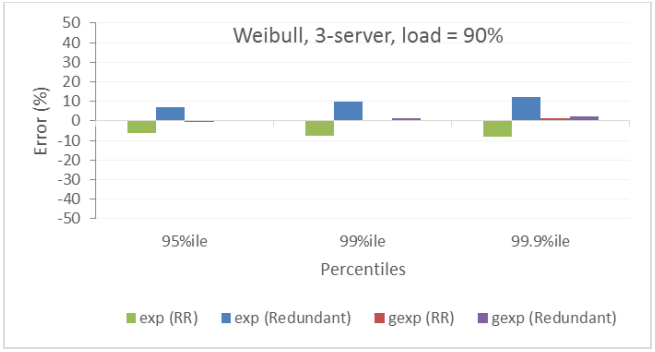
Subsystem tail latency prediction

- The accuracy of the prediction model is tested against different types of subsystems
  - Pure model-based subsystem
  - Hybrid measurement-and-model-based subsystem
  - Pure measurement-based subsystem
    - A Solr cluster of three Amazon EC2 m3.medium instances, each responsible for the same sample shard of the Wikipedia index.
- The service time distributions used in the experiments
  - Empirical distribution measured from a Google search test leaf node (CV= 1.12)
  - A heavy-tailed truncated Pareto distribution (CV = 1.20)
  - A heavy-tailed Weibull distribution (CV = 1.50)
  - All the distributions have the same mean service time as the empirical one (μ = 4.22 ms)

Subsystem tail latency prediction (cont'd)





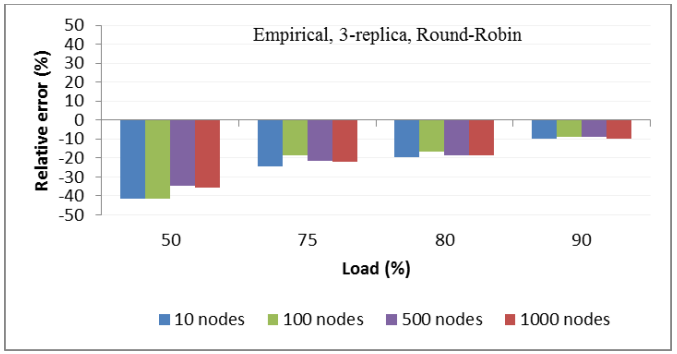


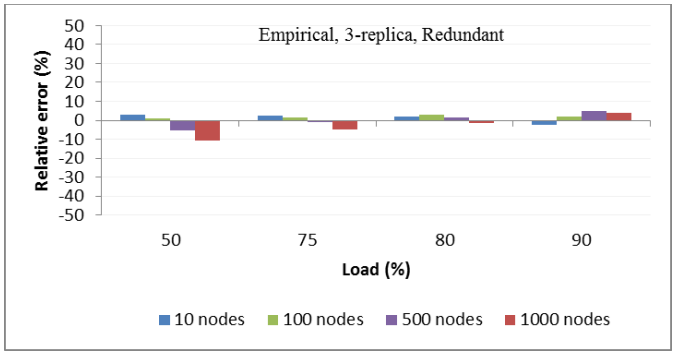
#clients	Percentiles		
	95th	99th	99.9th
20	-11.305	7.911	24.216
30	-3.233	5.295	13.429
40	-1.718	5.452	2.974
50	0.703	2.015	-1.381

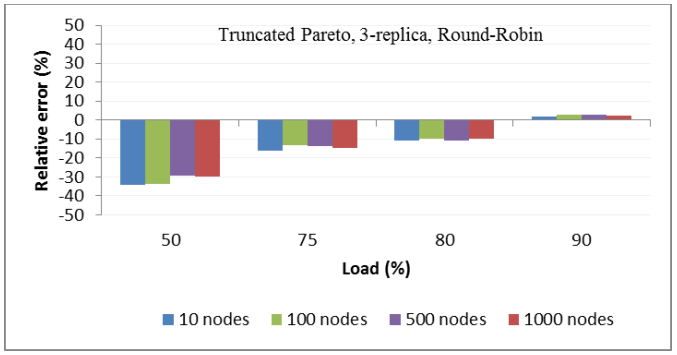
The prediction errors for the pure measurement-based subsystem

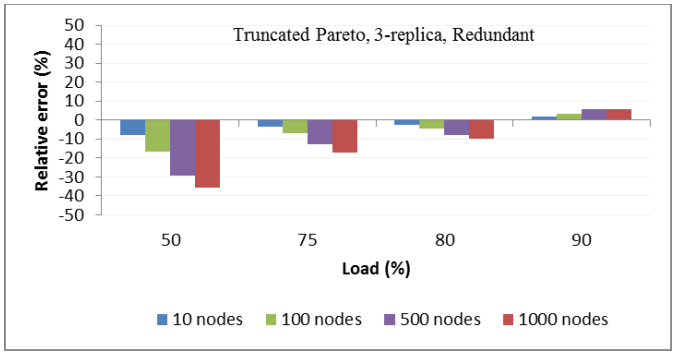
- The models with the generalized exponential distribution outperform those with the exponential distribution.
- For all the cases studied, the prediction errors are within 10% at the load of 90%.

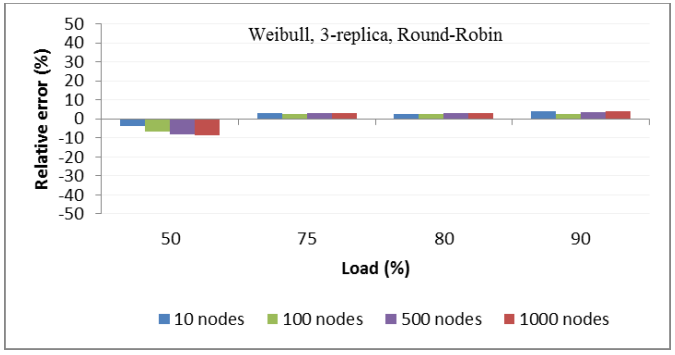
System tail latency prediction

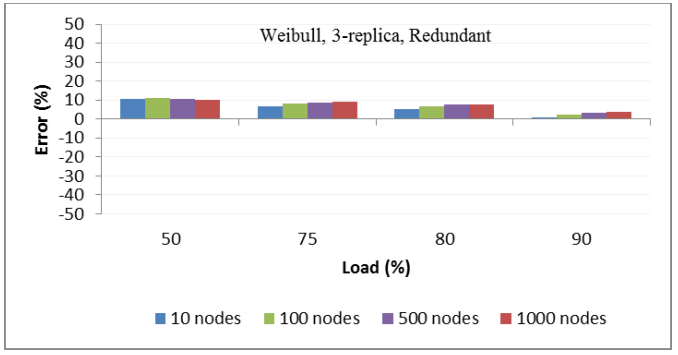












For all the cases studied, the prediction errors are within 10% at the load of 90%.

Facilitating resource provisioning

A Use Case: Given size of parallel database D to be searched and monetary budget C, whether the system to be deployed in a cloud may sustain R requests per second, while meeting the pth-percentile request response time of L ms.

- Build a replicated server cluster subsystem with m VMs by replicating a portion of the total database, i.e., D/N, to all the VM replicas;
- Measure the mean and variance of the task response time running a given task scheduling policy, at desired task rate $\lambda = R$;
- Find the parameters of the generalized exponential distribution using the measured mean and variance task;
- Estimate the 99^{th}-percentile request response time x_p;
- Finally, x_p is compared against L and the total cost for running N VM clusters with m each is compared against the associated budget C to see if both the tail SLO and monetary budget are met. If both are met, a feasible tail-constrained resource provisioning is found. Otherwise, the performance targets and/or budget are revised and then rerun the procedure.

Conclusions

- This paper proposed a simple model to predict the tail SLOs for OLDI applications, which requires only the mean and variance of task response time.
- The prediction model yields accurate prediction at the server loads of 90% or higher, which could be used to facilitate tail-constrained resource provisioning for OLDI applications.

Thanks!