Real-Time Resource Prediction Engine for Cloud Management

Christofer Flinta[1], Andreas Johnsson[1], Jawwad Ahmed[1], Farnaz Moradi[1], Rafael Pasquini[3,4], Rolf Stadler[2,4]
[1] Ericsson Research, Sweden, Email:{christofer.flinta, andreas.a.johnsson, jawwad.ahmed, farnaz.moradi}@ericsson.com
[2] ACCESS Linnaeus Center, KTH Royal Institute of Technology, Sweden, Email: stadler@kth.se
[3] Faculty of Computing (FACOM/UFU), Uberlândia, MG, Brazil, Email: rafael.pasquini@ufu.br
[4] Swedish Institute of Computer Science (SICS), Sweden

Abstract—Predicting resource requirements for cloud services is critical for dimensioning, anomaly detection and service assurance. We demonstrate a system for real-time estimation of the needed amount of infrastructure resources, such as CPU and memory, for a given service. Statistical learning methods on server statistics and load parameters of the service are used for learning a resource prediction model. The model can be used as a guideline for service deployment and for real-time identification of resource bottlenecks.

I. Introduction

Predicting the resource requirements of cloud services is intrinsically hard. One approach is to model the various layers of hardware and software using analytical models and to develop an overall model of the system. However, this approach requires thorough understanding of the functionalities of various components and their interactions, and the resulting system model becomes highly complex. In this demonstration we present an alternative approach, based upon machine learning, whereby the behavior of the service is learned from observations. While a large amount of observational data is needed, no detailed knowledge about the system components and their interactions is required.

In previous papers [1, 2] we have explored how to predict service quality from server statistics of the underlying infrastructure, such as CPU and memory usage. In this demonstration we turn the question around and aim for predicting the amount of resources needed in the infrastructure, based on service quality requirements and service load parameters. The objective is to build a predictive model for each resource type that maps service load to resource usage, given that the service quality is acceptable. The setup is illustrated in Figure 1.

Specifically, we collect statistics from a Linux kernel in a server machine and quality metrics from a video-streaming service, using VLC media player (videolan.org). The displayed video frame rate is used as the quality metric, while the resource levels to predict are CPU utilization and memory usage. The service load is represented by the number of concurrent users, which is estimated from the number of open TCP sockets in the server machine. This model can easily be extended to a larger service-load space.

These types of models can be used in different ways. One scenario is resource allocation when deploying a cloud service, where the model estimates the amount of resources needed for the service, e.g. CPU, memory, storage, accelerators, and networking. The estimates are based on expected system load, such as maximum number of users and their session lengths.

Another scenario is to identify resource bottlenecks for a running service, in case of SLA violation. The cloud operator and/or the tenant uses the model in real time to predict the requirements in terms of resources for the current load activity. The resource predictions that diverge from the actual resource usage hint towards the lack of those resources. Actions for the operator or tenant could be to increase the deficient resource if possible, such as adding more CPU or memory to the service or alternatively reduce the service load, such as limiting the number of users. Similar models can also be built for other resources, e.g. network and disk usage. In the considered scenario, each model must be able to make predictions periodically from real-time service-load data. In our demonstrator, new predictions are made each second.

II. Testbed

The testbed setup for experimentation and data generation for building the predictive models includes three parts, namely a server cluster that provides the video-streaming service from VLC servers over HTTP (using progressive download), a main VLC client machine that runs video-streaming sessions, and a load generator that creates an aggregated service demand from a set of background VLC clients, as illustrated in Figure 2. Each server machine runs a sensor that reads out and stores Linux kernel statistics periodically once per second. The main VLC client is instrumented with a sensor that reads out and stores the number of displayed video frames during each second, which is correlated to the quality experienced by the user. The load generator machine dynamically spawns and terminates the background VLC clients, according to specified load patterns, such as periodic load and flash-crowd load. In the demonstrator we use a periodic-load pattern where clients are started according to a Poisson process, starting at 70 clients/minute and varying between 20 and 120 clients/minute following a sinusoid function with a period of 60 minutes.

During the training phase, system-generated data is used to build a prediction model for each service resource of interest. The trained model is then applied in the test phase, where each data sample collected from the testbed gives a prediction of the current resource needs.

III. Demonstration

The demonstration shows the predictions of CPU and memory usage together with the actual CPU and memory usage over time, as illustrated in Figure 3. A difference in actual and

predicted resource usage signals an anomaly in terms of resources. In the picture, we can see that at certain points in time, when the SLA is violated, there is a discrepancy between actual and predicted CPU. This knowledge can be used as an indicator for root-cause analysis. The displayed values are calculated from traces collected at the testbed, where CPU and memory data is stored together with VLC video frame rate and number of active TCP sessions.

REFERENCES

[1] R. Yanggratoke, J. Ahmed, J. Ardelius, C. Flinta, A. Johnsson, D. Gillblad, and R. Stadler, "Predicting service metrics for cluster-based services using real-time analytics," International Conference on Network and Service Management (CNSM 2015), Barcelona, Spain, November 2015.

[2] R. Yanggratoke, J. Ahmed, J. Ardelius, C. Flinta, A. Johnsson, D. Gillblad, and R. Stadler, "Predicting Real-time Service-level Metrics from Device Statistics," International Symposium on Integrated Network Management (IM 2015), Ottawa, CA, May 2015.

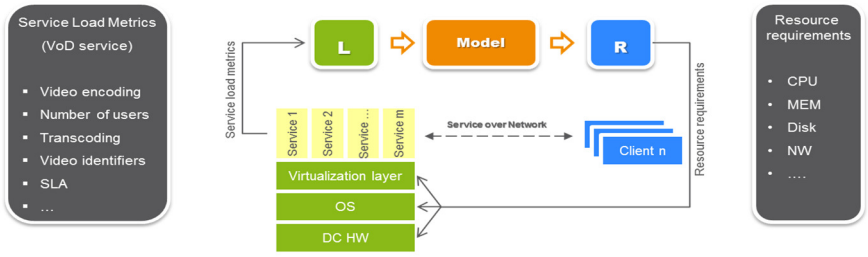

Fig. 1. The modeling for a video-streaming service. The model takes the load parameters of the service (L) as features and outputs estimates of resource requirements (R).

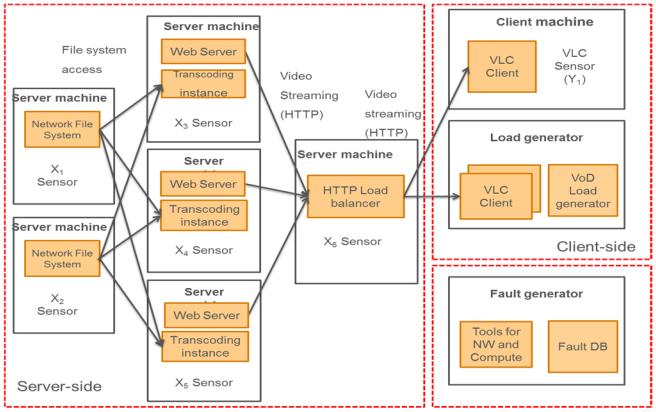

Fig. 2. The testbed setup for video-streaming service experiments

Fig. 3. Predictions for resource needs and actual resource consumption in terms of CPU and memory utilization for a video-streaming service over-time.