The Performance Inversion in Edge Computing System

1. Introduction

Cloud computing and edge computing are both widely applied technologies in many areas like IoT, AI, and autonomous driving. Edge computing is always regarded as more efficiently compared with cloud computing as edge servers are usually closer to users. However, the latest research work indicates that in certain situations, the edge system has worse performance than the cloud system, especially in high utilization[1]. In order to explore and verify the performance inversion problem, this paper designed and implemented the simulator of edge and cloud computing systems, and verified the performance inversion problem as well as relative theories through intensive experiments from different aspects.

2. Theoretical Analysis

Assuming that the edge system has k servers located on different sites, each server accepts a request queue, so an edge system holds k independent request queues. A cloud system is a cluster consisting of K servers that accept a single queue as input, the requests in the queue will be sent to each idle server for further processing. Both systems share the same scheduling algorithm—FCFS (First Coming, First Serving). The edge system and cloud system can be both illustrated and analyzed by queue theory.

According to Kendall notation rule in queue theory, a queueing model is usually marked as symbol X/Y/Z, X denotes the probability distribution of the time interval of the request queue, Y denotes the probability distribution of service time of requests, Z denotes the number of servers in the system. Given that the theories need to be verified aims to explain the situation of the requests queue and service time are both general distributions, so the cloud system can be represented as a single G/G/K model, and the edge system can be represented as k G/G/1 models.

The end-to-end latency of a typical application usually consists of three parts. 1. Round trip latency *n*, waiting time in the queue *w*, and execution time *s*. The whole latency of the edge system and cloud system can be represented by equation 2-1 and equation 2-2.

$$T_{edge} = n_{edge} + w_{edge} + s_{edge} \tag{2-1}$$

$$T_{cloud} = n_{cloud} + w_{cloud} + s_{cloud} \tag{2-2}$$

Performance inversion means the total latency of edge system is larger than the total latency of cloud system.

$$n_{edge} + w_{edge} + s_{edge} > n_{cloud} + w_{cloud} + s_{cloud} \tag{2-3}$$

In this experiment, assuming that edge server and cloud server share the same hardware configuration , then $S_{edge} = S_{\text{cloud}}$, so equation 2-3 can be simplified to equation 2-4.

$$n_{edge} + w_{edge} > n_{cloud} + w_{cloud} \tag{2-4}$$

let $\Delta n = n_{cloud} - n_{edge}$, equation 2-4 can be simplified to equation 2-5.

$$\Delta n < w_{edge} - w_{cloud} \tag{2-5}$$

2.1 Workload balance

In equation 2-5, the waiting time can be represented by the expectation of waiting time. According to Allen-Cunneen approximation method. The expectation of waiting time of G/G/1 queuing system can be calculated by equation 2-6

$$E(w) \approx \frac{\rho}{\mu(1-\rho)} \bullet \frac{{c_A}^2 + {c_B}^2}{2} \tag{2-6}$$

Respectively, the expectation of waiting time of G/G/K queuing system can be calculated by equation 2-7：

$$E(w) \approx \frac{P_s}{\mu(1-\rho)} \bullet \frac{{c_A}^2 + {c_B}^2}{2k} \tag{2-7}$$

In equation 2-7, ρ denotes the system utilization which can be represented by the ratio of arrival rate to service rate, and P_s is given by equation 2-8

$$P_s \approx \begin{cases} \dfrac{\rho^k + \rho}{2}, if\ \rho > 0.7 \\ \rho^{\frac{s+1}{2}}, if\ \rho < 0.7 \end{cases} \tag{2-8}$$

Since Allen-Cunneen approximation method is more accurate in the condition of high utilization, so the situation of ρ>0.7 in equation 2-8 is adopted in this experiment. Substituting equation 2-8, 2-7, 2-6 into equation 2-5, we can get equation 2-9 which is the first theory that needs to be verified in this experiment.

$$\begin{aligned} \Delta n < {} & \rho_{edge} \frac{1}{\mu_{edge}(1-\rho_{edge})} \frac{{c_{edgeA}}^2 + {c_{edgeB}}^2}{2} \\ & - \frac{{p^k}_{cloud} + p_{cloud}}{2} \frac{1}{\mu_{edge}(1-\rho_{cloud})} \frac{{c_{cloudA}}^2 + {c_{edgeB}}^2}{2k} \end{aligned} \tag{2-9}$$

2.2 Workload Skew

The analysis in section 2.1 is based on a certain situation that each edge server has the same arrival rate. However, the arrival rates at different edge sites are dynamic, which is the so-called workload skew. In the situation of skewed workload, each edge server accepts an arrival rate of λ_i, and $\lambda = \sum_i \lambda_i$. The expectation of waiting time of each edge server is given by equation 2-10

$$E(w_i > 0) = \frac{\sqrt{2}}{1-\rho_{edgei}} \tag{2-10}$$

The expectation of waiting time of cloud system is

$$E(w \mid w > 0) = \frac{\sqrt{2}}{(1-\rho)\sqrt{k}} \tag{2-11}$$

The expectation of waiting time of edge system is the weighted sum of expected waiting time of each edge server, and weight of each edge server is calculated by $w_i = \lambda_i/\lambda$, substituting equation 2-10 and 2-11 into equation 2-5

$$\Delta n < \sqrt{2}\left(\sum_i \frac{w_i}{1-\rho_{edgei}} - \frac{1}{\sqrt{k}(1-\rho_{cloud})}\right) \tag{2-12}$$

Equation 2-9 and 2-12 are the important theories about performance inversion that need to be verified in this report. They give the critical condition of performance inversion in two typical scenarios of edge computing system—workload balance and workload skew. This report will design reasonable and intensive experiments to verify these theories.

3. Experimental Objective

This report aims to explore the performance inversion problem of edge computing versus cloud computing. There are three main objectives in this experiment.

1. Design and build a simulator of both edge and cloud system for intensive simulation experiments.
2. Verify the performance inversion problem.
3. Verify the theories mentioned in section 2 (Equation 2-9 and 2-12) could predict the accurate Critical condition (cutoff utilization) about when performance inversion could happen.

4. Method

4.1 Simulator Design and Implementation

Based on the analysis of section 2, the simulator used in this experiment consisted of four main components.

1. *Data generator*: Since the request arrival time interval and service time could be arbitrary distribution, the data generator in this simulator should cover several common distribution types and generate corresponding data according to a series parameter.
2. *Cloud system model and edge system model*: Both models accept the data generated by component 1 and conduct simulation following the FCFS rule, then return the average end-to-end latency of request queue and find the cutoff utilization in simulation.
3. *Theoretical value calculator*: For each data generated by component 1 and simulated by component 2, theoretical value calculator could give the theoretical cutoff utilization for further comparison.
4. *Visualization and analysis*：For the results returned by the component 2 and 3, draw the line chart and make analysis

Figure 4-1 illustrates the architecture of this simulator. User could specify a series parameter, data generator would generate input data and system models would be initialized according to the input parameters, then the system models would start the simulation. Finally, the simulation results and input information would be sent to visualization module and theoretical value calculator for further

analysis.

Mixed programming solution was adopted in the implementation of the simulator. Specifically, user interface, data generator, definition of system models, visualization, and analysis modules were implemented by python. FCFS schedule module is the kernel part of this simulator which could be time-consuming when the simulation duration is too long or number of servers is set to a big value, so this module was implemented by C++ with multi-threading acceleration metric and coupled with system models by Pybind[2]. Because too many parameters need to be set for each simulation result in lots of manual effort, a graphical user interface was also implemented which could improve the efficiency of experiments.

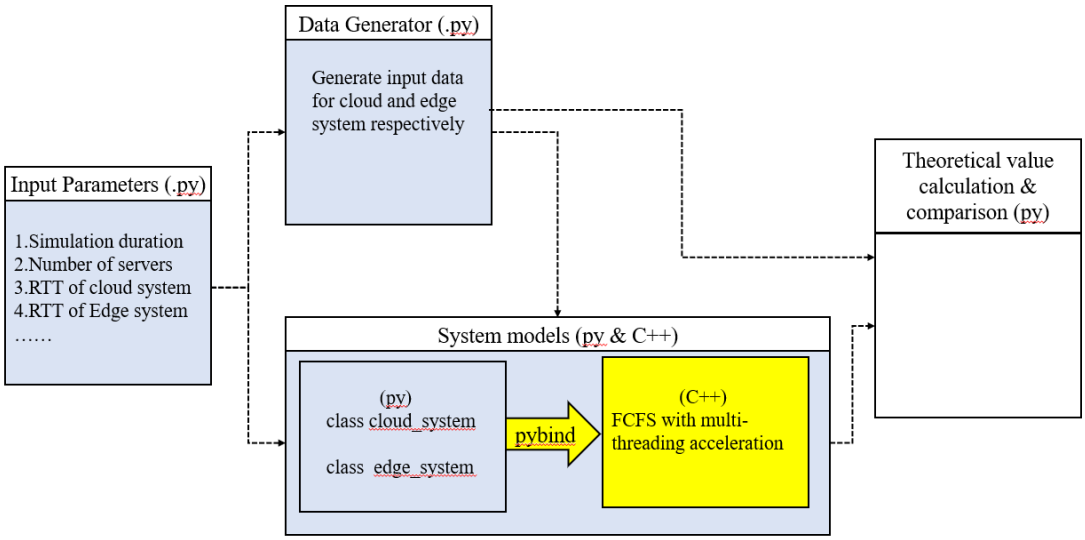


Figure 4-1: Simulator architecture

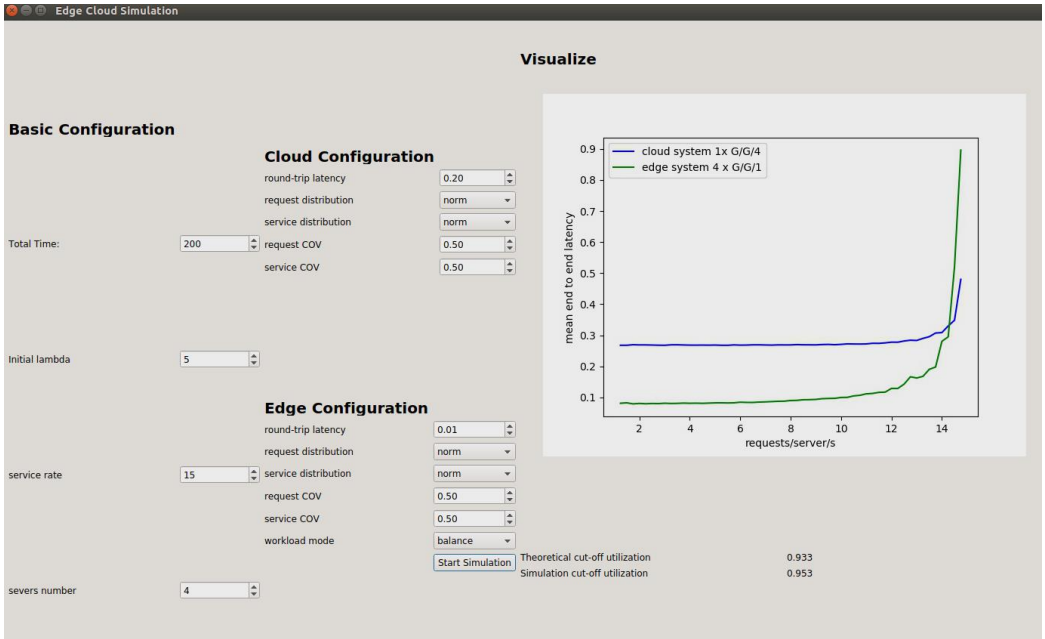


Figure 4-2: GUI of the simulator

4.2 Experimental Design

Based on the simulator implemented in section 4.1, intensive experiments could be conducted to verify the theories of 2-9 and 2-12. The experiments consisted of several simulations, and the request arrival rate would increase until reaching the max value (max value = number of servers * service rate) in each simulation.

From equation 2-9, there are two main factors could influence the critical condition of performance inversion. In order to explore the relationship between performance inversion and system utilization, variable-controlling approach was introduced in this experiment. During the simulation, the Cov (coefficient of variation) values of both request arrival distribution and service time distribution were set to a fixed value, as well as other parameters except request rate. Then the simulation-based cutoff utilization value and theory-based cutoff utilization value would be compared under different input distribution types.

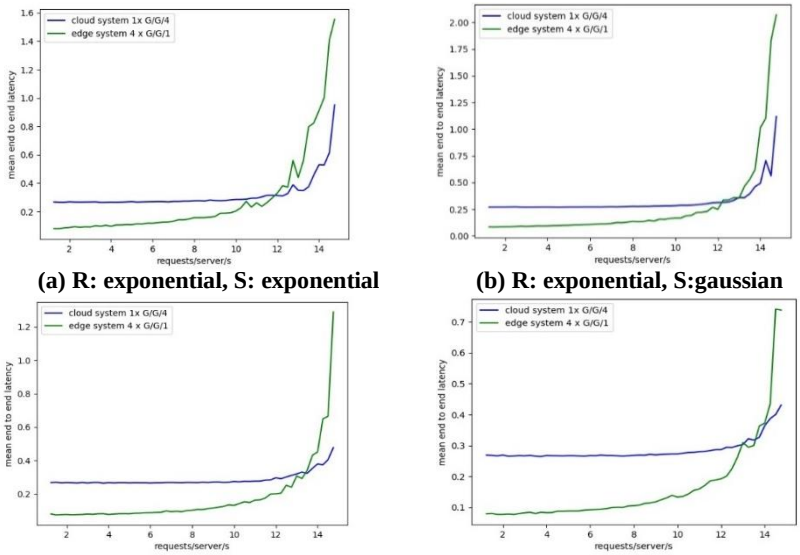
From equation 2-12, the weights of edge servers in each simulation are different. In order to verify the theory, the utilization that corresponds to the performance inversion happened for the first time would be regarded as the cutoff utilization. The verification would also consider different input distribution types.

5. Results

The experiments could be classified into two categories. 1. G/G/K cloud system model and k x G/G/1 edge system model with balanced workload. 2. G/G/K cloud system model and k x G/G/1 edge system model with skewed workload.

5.1 G/G/K cloud system model and k x G/G/1 edge system model with balanced workload

According to the analysis in section 4.2, the simulation experiments were performed based on four different combinations of request arrival distribution types and service time distribution types: (a). Request arrival distribution: exponential, service time distribution: exponential. (b). Request arrival distribution: exponential, service time distribution: gaussian. (c). Request arrival distribution: K-erlang, service time distribution: gaussian. (d). Request arrival distribution: gaussian, service time distribution: exponential. Other parameters were also fixed during simulation: simulation time is 200, service rate is 15, the round-trip-latency of cloud system is 0.2, the round-trip-latency of cloud system is 0.01, number of servers is 4. The initial request rate was the only variable set to 5 and increased gradually in all simulation iterations, the simulator would return the simulation cutoff utilization and theoretical cutoff utilization for each request rate.



(a) R: exponential, S: exponential **(b) R: exponential, S:gaussian**

(c) R: K-erlang, S: exponential **(d) R: gaussian, S: exponential**

Figure 5-1: Mean end-to-end latency in different combination of arrival distribution and service distribution (edge workload in a balance mode)

Figure 5-1 illustrates how the mean end-to-end latency changed as the request arrival rate changed for both cloud system and edge system. we could observe 2 phenomena: 1. The mean end-to-end latency would increase with the increase of system utilization. 2. Performance inversion would occur at high utilization level.

(a) R: exponential, S: exponential								
simulation	77.1%	77.5%	75.4%	74.9%	74.4%	71.5%	77.2%	76.0%
theory	77.8%	77.8%	77.8%	77.8%	77.8%	77.8%	77.8%	77.8%
difference	0.70%	0.30%	2.40%	2.90%	3.40%	6.30%	0.60%	1.80%
(b) R: exponential, S: gaussian								
simulation	81.6%	82.4%	83.9%	82.9%	82.7%	80.8%	83.0%	81.0%
theory	84.9%	84.9%	84.9%	84.9%	84.9%	84.9%	84.9%	84.9%
difference	3.30%	2.50%	1.00%	2.00%	2.20%	4.10%	1.90%	3.90%
(c) R: K-erlang, S: exponential								
simulation	92.8	93.5%	93.5%	93.0%	93.1%	95.0%	93.6%	92.8%
theory	95.1%	95.1%	95.1%	95.1%	95.1%	95.1%	95.1%	95.1%
difference	2.30%	1.60%	1.60%	2.10%	2.00%	0.10%	1.50%	2.30%
(d) R: gaussian, S: exponential								
simulation	89.7%	87.5%	87.6%	84.8%	86.3%	89.9%	89.0%	88.4%
theory	84.9%	84.9%	84.9%	84.9%	84.9%	84.9%	84.9%	84.9%
difference	4.80%	2.60%	2.70%	0.10%	1.40%	5.00%	4.10%	3.50%

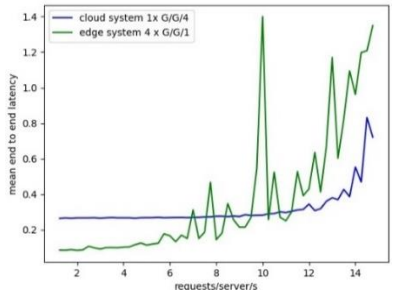
Table 5-1. Comparison of simulation cutoff utilization and theoretical cutoff utilization

Table5-1 illustrates the accuracy of theoretical model (equation 2-9), the average difference between simulation result and theoretical result of experiment (a) is 2.3%, 2.61% for experiment (b), 1.69% for experiment (c) and 3.03% for experiment (d). All the average difference values were calculated

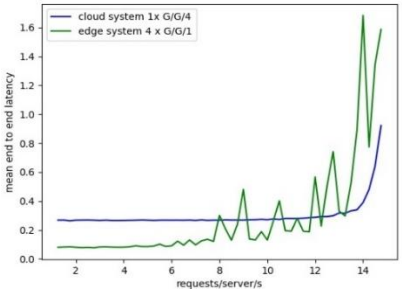
by the absolute value of the difference of each iteration. From the comparison result, we can think that the difference between simulation value and theoretical value has slight gap, thus theory 2-9 has been verified to be correct.

5.2 G/G/K cloud system model and k x G/G/1 edge system model with skewed workload.

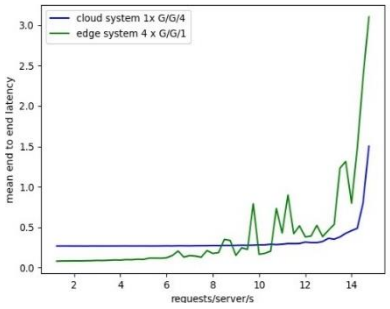
Experiment 5.2 has almost the same configuration as the experiment in section 5.1 except for the workload mode of edge system. In experiment 5.1, the request rate was distributed equally across all edge servers which means a balanced workload of edge system. In order to verify theory 2-12, the request arrival rate was distributed unequally across all edge servers to simulate workload skew in real scenarios.



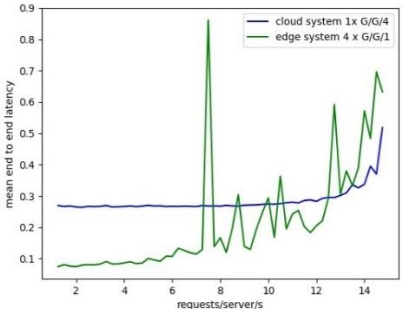
(a) R: exponential, S: exponential



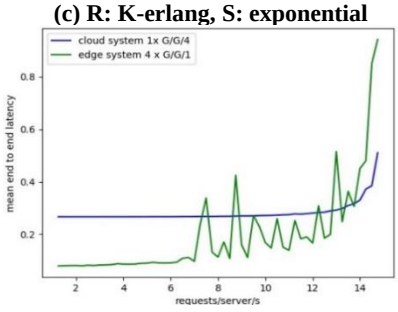
(b) R: exponential, S:gaussian



(c) R: K-erlang, S: exponential



(d) R: gaussian, S: exponential



(e) R: general, S: general

Figure 5-2: Mean end-to-end latency in different combination of arrival distribution and service distribution (edge workload in a skew mode)

Figure 5-2 shows how the mean end-to-end latency altered with the increase of the request arrival rate for both cloud system and edge system. We can observe three phenomena: 1. Performance inversion would also occur when edge system has skewed workload. 2. The cutoff utilization tends to be lower compared with balanced mode which has the same system parameter setting. 3. The

fluctuating range of mean end-to-end latency of edge system is higher than edge system in balance mode. Compared with experiment 5.1, this experiment has a different distribution combination: R: general, S: general. In this simulation, general data was generated by random data without any distribution formula specified but has a fixed mean value.

(a) R: exponential, S: exponential								
simulation	52.3%	56.2%	49.9%	54.3%	48.1%	52.0%	46.2%	48.0%
theory	45.0%	46.7%	53.3%	48.3%	48.3%	43.3%	46.7%	41.7%
difference	7.30%	9.50%	3.40%	6.00%	0.20%	8.70%	0.50%	6.30%
(b) R: exponential, S: gaussian								
simulation	50.2%	63.6%	63.6%	45.6%	48.9%	53.6%	49.3%	54.4%
theory	43.3%	50.0%	55.0%	43.3%	46.7%	45.0%	50.0%	43.3%
difference	6.90%	13.6%	8.60%	2.30%	2.20%	8.60%	0.70%	11.10%
(c) R: K-erlang, S: exponential								
simulation	51.6%	46.7%	61.2%	54.7%	61.4%	53.6%	53.1%	53.8%
theory	45.0%	46.7%	53.3%	50.0%	48.3%	45.0%	46.7%	48.3%
difference	6.60%	0.00%	7.90%	4.70%	13.1%	8.60%	6.40%	5.50%
(d) R: gaussian, S: exponential								
simulation	50.0%	50.8%	49.0%	48.7%	56.7%	49.4%	55.1%	46.6%
theory	52.7%	46.7%	48.3%	50.0%	46.7%	43.3%	45.0%	46.7%
difference	2.70%	4.10%	0.70%	1.30%	10.0%	6.10%	10.1%	0.10%
(e) R: general, S: general								
simulation	48.8%	50.0%	68.7%	64.8%	50.8%	60.4%	54.0%	52.8%
theory	48.3%	45.0%	46.7%	50.0%	45.0%	48.3%	50.0%	48.3%
difference	0.50%	5.00%	22.0%	14.8%	5.80%	12.1%	4.00%	4.50%

Table 5-2. Comparison of simulation cutoff utilization and theoretical cutoff utilization with edge system in skew mode

Table5-2 illustrates the accuracy of theoretical model (equation 2-12), the average difference between simulation result and theoretical result of experiment (a) is 5.24%, 6.75% for experiment (b), 6.6% for experiment (c), 4.39% for experiment (d) and 8.59 for experiment (e). Compared with experiment 5.1, the theory 2-12 used for predicting cutoff utilization when edge system has skewed workload presented lower accuracy but still has enough high precision even for the real engineering application. Thus, the theory 2-12 has been verified to be correct.

6. Conclusion

In this report, we investigate the performance inversion problem in edge and cloud computing systems. Two main works were conducted in the experiment. 1. Designed and implemented an effective simulator with mixed programming strategy and multi-threading acceleration, a graphical user interface was also implemented for efficient intensive experiments. 2. Investigated the performance inversion problem under different request arrival distribution types and service time distribution types, as well as different edge system workload modes (balance and skew), and verified the two theories about predicting cutoff utilization to be correct. Based on the simulator implemented in this report, we can explore more problems in edge-cloud systems, not only limited

to performance inversion issue.

7. References

[1] Ali-Eldin, A., Wang, B., & Shenoy, P. (2021, November). The hidden cost of the edge: a performance comparison of edge and cloud latencies. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-12).

[2] https://github.com/pybind/pybind11