M-003

# On the Quality of Service for Fully-distributed Event Processing System

シューショートケオ　サンヤナン† 山口 弘純† 東野 輝夫†
Sunyanan Choochotkaew Hirozumi Yamaguchi Teruo Higashino

## 1. Introduction

In this paper, we propose an approach for distributing continuous event processing tasks based on the quality of service significance in a self-organized-and-fair way. Among Quality of Service (QoS) attribute definition and taxonomies [1][2], the efficiency and reliability are mainly but flexibly taken consideration for task-agent assignment and data collection process in event processing systems especially in fully-distributed environment owning to heterogeneous, highly dynamic and mobile environments [2][3][4][5]. Nevertheless, there is no fairness concern among those tasks. We add the availability attribute to normalize efficiency and reliability into the same metric. Since event processing is a never-end query task, we need to assign tasks to the agent permanently without arrival order. We propose a QoS-oriented task selection algorithm for each processing agent to independently pick the task to process according to its capability and decision of other agents. The experiment shows our superior to existing approaches from total and average throughput.

## 2. Method

To formulate a service-quality tradeoff function in fully-distributed event-processing systems sharing equally among multiple users, we quantize the service quality attributes in the term of selection probability in the task assignment process. Firstly, we discuss on attributes that are usually in concern in such a dynamic and mobile environment. Then, with the weighted values among those attributes given by query invokers (users), we compute a selection probability on each task at each processing agent and perform adaptive selection algorithm. The probability can be different for the same tasks on a different agent. For instance, the efficiency-intensive tasks have a selection probability on the agents with high performance higher than that on the agents with low performance.

### 2.1 Quality of Service Attributes

Fully-distributed networks usually face challenges from dynamicity and mobility of the node components. In addition to the most common service quality attribute like *efficiency*, *reliability* is also required to ensure processing success.

#### 2.1.1 Efficiency

Commonly, the efficiency of processing is the preferable attribute of any services. The efficiency refers to throughput and latency per each query, which are directly affected by agent performance. In Ref. [2], the authors classify QoS metrics into two groups: time-based and content-based. We consider content-based metrics as strict specification, applied as parameters in query processing and time-based metrics as flexible specification, used for factors of task assignment algorithm. *Tasklet*, a middleware for distributed applications, applies Quality of Computation (QoC) providing execution guarantees on multiple attributes on task execution by assigning the requirement-matching agents for each task unit [3]. Nevertheless, in terms of speed, users have to specify the exact minimum spec of the agent. That might not cause unavailability of the service. One of the efficiency-intensive applications is a traffic analysis from stationary cameras at the crossroad, where we can set-up the always-available and powerful agent nearby.

#### 2.1.2 Reliability

In highly dynamic-and-mobile networks, the reliability of services is a primary concern, as found in [4]. The reliability indicates the success rate of query processing and delivering. Adding replications increases the reliability in dynamic and mobile networks. However, it comes out with resource consumption tradeoff. In *tasklet*, users can specify the number of replication in the query requirement [3]. However, in a custom way of routing area, the number of replication is adaptable regarding the spreading algorithm. In our previous work, we proposed an algorithm to minimize the number of replications while considering the delivering probability [5]. Nevertheless, the higher number of replications in continuous stream processing incurs a higher risk of unavailability of the whole system. While reliability-intensive tasks consume a high amount of resources, we should treat the rest tasks running concurrently with fairness. One example of high-reliability-needed is car accident detection from volunteering driving recorders, which are highly dynamic and mobile.

#### 2.1.3 Availability

The availability is an attribute for indicating the ability to conform to the efficiency and replication requirements. While giving weight to availability attribute will increase an opportunity to be selected equally at all processing agents, the chance of selection from reliability will decrease after assigning the tasks to some agents as a tradeoff. Consequently, the availability-intensive tasks will always have an equal or higher selected chance compared to the query with high reliability. Likewise, efficiency weight puts a more selected chance to high-performance agents by dragging that off from the lower-performance agents as a tradeoff. As a result, the availability-intensive tasks will have a higher selected chance than the query with high efficiency at middle and low-performance processing agents. In conclusion, the availability attribute outperforms in heterogeneous networks of numerous devices.

† 大阪大学情報科学研究科 Graduate School of Information Science and Technology, Osaka University

## 2.2 QoS-oriented Task Selection Algorithm

We assume that we can categorize any agents into one of three categories: high, medium, and low performance according to some standard benchmarks. The QoS-oriented task selection algorithm is to find the many-to-many mapping from a set of QoS-specified tasks (τ) to a set of performance-classified processing agent under their resource constraints. The resource constraint checking function could be in many terms. Some are fixed thresholds like the computation and communication capacity. Meanwhile, some are varied like the guarantee of processing time estimation. The discussion about the constraint function is out of focus on this paper. The QoS attributes considered are efficiency (*e*), reliability (*r*), and availability (*a*).

### 2.2.1 Selection Probability Computation

We denote the selection probability at high (*h*), medium (*m*), and low-performance (*l*) agents by attributes $\theta \in (e, r, a)$ with $p_{h\theta}$, $p_{m\theta}$, and $p_{l\theta}$. The total selection probability from all attributes must be equal at initial state, when there is no assignment for any tasks: $\sum_{c\in(h,m,l)} p_{ce} = \sum_{c\in(h,m,l)} p_{cr} = \sum_{c\in(h,m,l)} p_{ca}$ .

The probability from efficiency is larger in higher-performance agents: $p_{he} > p_{me} > p_{le.}$ Meanwhile, there is no difference in availability, $p_{ha} = p_{ma} = p_{la,}$ considered as $p_{neutral}$. For each task *i,* users give weight, $w_{i\theta}$, for each attribute, where $\sum_{\theta\in(h,m,l)} p_{ca} = 1$. With a maximum replication number, *M*, when any task *i* has $w_{ir} = 1.0$, the probability of reliability on all agents with any performance *c* decreases relatively to the current number of accepting agents $\mu \in [0, M)$ as a tradeoff for resource usage:

$$p_{cr} = (M - \mu)/M \times p_{neutral}$$

The probability of task *i* on agent with performance *c* ($\pi_{ic}$) is:

$$\pi_{ic} = \frac{\sum_{\theta\in(e,r,a)} w_{i\theta} p_{c\theta}}{\sum_{j\in\tau, j\neq i} \sum_{\theta\in(e,r,a)} w_{j\theta} p_{c\theta}}$$

### 2.2.2 Selection Algorithm

The goal of selection is to map all the tasks and their designated replication to the agents. The selection algorithm performs at each agent in a non-synchronized manner due to the dynamicity of the networks. Each agent will exchange its decision as a proposal with a score, which is the accumulated probability of selection. The higher score means higher priority towards assignment consensus. For making a decision, each node follows the following steps:

*Step 1:* to compute in-required tasks from all task information and other higher-score proposals and generate the base pool and the required pool from all task information with in-required marks and its capability

*Step 2:* to select a set of tasks under capacity constraints from the required pool then base pool when no required left

*Step 3*: compute the score, validate, and exchange the decision

Agents need to re-compute the invalid proposal. The proposal will be valid if the selected tasks cover or are all in the in-required list of tasks from Step 1.

### 2.2.3 Algorithm Evaluation

The evaluation has conducted logically with three kinds of tasks: efficiency-intensive, reliability-intensive, and availability-intensive. We set $p_{neutral} = 1$, $p_{h\theta}$:$p_{m\theta}$:$p_{l\theta}$ = *2.4:0.6:0,* and *M*=10.

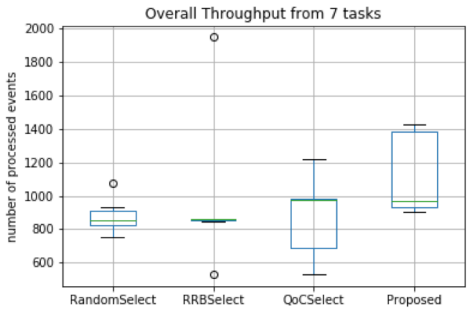


Fig 1 Throughput Comparison Result

We emulate three levels of processing agents: high, medium, and low. The processing times for all kinds of tasks on each type are 0.02s, 0.05s, and 0.1s, respectively. Intuitively, the maximum capacity of tasks is 5, 2, and 1. We balance the number of each agent as 2, 5, and 10. As one reliability-intensive task requests at least 10-agents, the dense-and-balance ratio of each type of tasks is 3:1:3, sampling every 0.2s. The testing time is 60 seconds. We compare our proposed method with random, round robin (RRB), and QoC selections. Since random and our proposed approach depend on probability, we use the average value from 10 times. The result has shown in Fig. 1. Our proposed gains the highest amount of throughput in total and average, specifically, 29.4%, 17.6%, and 31.3% increased from random, RRB, and QoC methods, respectively. Furthermore, our method serves the QoS specification well as the efficiency-intensive tasks get high-performance agents and high throughput, and the reliability-intensive task gets the specified replications. Meanwhile, the availability-intensive tasks get the maximum number of agents.

## 3. Conclusion

Towards fully-distributed event processing systems, quality of service management is a key to serve multiple kinds of applications with satisfaction and fairness in a heterogeneous and dynamic environment. We propose a novel approach of service quality requirement specification considering tradeoff over efficiency, reliability, and availability. With users' specified weights, we compute a selection probability of each task on each processing agent regarding its performance and capacity constraints. The emulation tests have been conducted to show the superior to potential approaches, including random, round robin, and quality of computation selection algorithm.

### Reference

[1] C. Bisdikian, L. M. Kaplan, and M. B. Srivastava. "On the quality and value of information in sensor networks," ACM Trans. Sen. Netw. 9, 4, Article 48 (2013).

[2] D.-I. Sven Schmidt, "Quality-of-service-aware data stream processing," (2007).

[3] D. Schafer, J. Edinger, J. M. Paluska, S. VanSyckel, and C. Becker, "Tasklets: better than best-effort" computing," International Conference on Computer Communication and Networks (ICCCN) (2016).

[4] T. P. Yaxita Patel, "A survey on dtn routing protocols," International Journal of Engineering Development and Research (IJEDR), vol. 3, pp. 817–819 (2015)

[5] S. Choochotkaew, H. Yamaguchi, and T. Higashino, "BALANCE: A Robust Routing Protocol in Self-Organized Civilian DTN," International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), (2018)