ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

August 2013, 20(Suppl. 1): 35–39
www.sciencedirect.com/science/journal/10058885

The Journal of China Universities of Posts and Telecommunications

http://jcupt.xsw.bupt.cn

QoS-aware computational method for IoT composite service

ZHOU Ming (✉), MA Yan

Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract

This paper integrated the quality of service (QoS) requirements of Internet of things (IoT) composite services, and put forward effective decomposition and optimization methods of the QoS metrics. The authors break down the complex QoS calculation model into four basic models, and each model is given a computational method. Combined with the QoS technology of the composite service, the authors adopt the algorithm to find the sub-optimal service with an acceptable cost under the QoS constraints. The proposed algorithm can quickly provide QoS computing services than other two algorithms for large-scale IoT compose services.

Keywords QoS, IoT, composite service, backtracking algorithm, wireless sensor network (WSN)

1 Introduction

In 1999, IoT was first used by Ashton [1]. In 2005, the ITU formally defined the concept of the IoT and published ITU Internet Report 2005: Internet of Things. This report introduced forms, features, technology, opportunities and challenges of IoT. It could be summarized that IoT is the integration of information space and physical space. Currently there are 9 billion interconnected devices and it is expected to reach 24 billion devices by 2020 [2]. Therefore the related research on IoT has major economic significance to society, people's livelihood and technological innovation.

With the further development of IoT, IoT technology has been widely used in practice. In enterprise applications, simple service cannot accomplish a complex task alone. Therefore, the demand for combined multiple services in accordance with rules to generate new services in order to complete complex requirement grows gradually. In the cloud computing era, many Web services were published on the Internet. To meet the user's QoS requirements, people need efficient QoS model and calculation methods for a variety of compose services. ISO 8402 and ISO 9000 [3–4] made the following description of quality: The totality of features and characteristics of a product or service bear on its ability to satisfy stated or implied needs. Generally, QoS is a group of indicators that reflects the non-functional properties of services.

This paper, based on the IoT environment, discuss the QoS computational methods, and find a QoS-aware service selection algorithm for IoT composite service. Sect. 2 introduces the related research projects. Sect. 3 describes the QoS computational method of IoT. Sect. 4 describes a QoS-aware service selection algorithm for IoT composite service and Sect. 5 draws the conclusions.

2 Related work

The implementation of compose service can be divided into three phases: first, through manual or automatic methods, service processes are generated using workflow-based pattern. The various parts of composite service contents are defined, and subtasks are determined. Second, is to find the corresponding specific services for the subtask in previous stage. Third, is the selection process of services. A subtask may have a lot of service with the same functions. These services can meet the target task, but have the difference QoS metrics. Therefore, how to select the appropriate service according to the QoS

Received date: 31-07-2013
Corresponding author: ZHOU Ming, E-mail: zhouming@buptnet.edu.cn
DOI: 10.1016/S1005-8885(13)60252-6

metrics is a key issue to research.

In the object-oriented Web services environment, the compose service has become an inevitable trend in order to meet the various needs of users. However, due to the uncertainty of the environment and the randomness of the service, the QoS of Web services also has a strong uncertainty. QoS was first telecommunications concept. ITU-T modified the earlier definition in E.800, and redefined the concept of QoS in the E.860 [5]. The following definition of QoS is used. The degree of conformance of the service delivered to a user by a provider with an agreement between them. In the process of composite services, QoS research has also seen considerable development. W3C puts forward the QoS requirements and the possible way utilize Web services [6], and gives the 13 possible generic QoS metrics, including Performance, Reliability, Scalability, Capacity, Robustness, etc. Cardoso did a lot of pioneering work in Ref. [7], and gives QoS assessment models and evaluation indicators, plus specifics for service cost, service time and reliability. In Ref. [8], Maximilien et al. provides a range of QoS metrics instead of the average. In Ref. [9], Hwang et al. describes the value of the QoS metrics using the probability distribution. In order to choose the service composition for the global optimal execution plan, In Ref. [10], Li et al. put forward a QoS-based fuzzy multi-attribute decision model, especially for heterogeneous QoS data. It specifically describes the heterogeneous QoS data, and then evaluates the QoS of the real number, interval number, and language-based data. In short, QoS optimization of service composition has been studied in-depth through existing research. These researches are utilized in the study of IoT services in this paper.

WSN include a variety of dynamic distributions of the sensor and actuator which are used to display the physical state or execute an order. The application requirements determine the diversity of the wireless sensor network's QoS requirements. At present, QoS research in WSN are sequential assignment routing (SAR), minimum cost forwarding, SPEED, energy aware routing, multi-path multi-speed protocol (MMSPEED), ReInForM, Mobicast [11]. This research proposed QoS-aware routing protocols for WSN to provide fine QoS in the energy and bandwidth constraints of the individual circumstance. However, their perspective is limited to how to provide QoS in WSN.

In the MiLAN project [12], Heinzelman et al. proposed that Middleware configure of the network and nodes to meet the application QoS requirements. In this article, the cooperation between the middleware the application is needed. MiLAN needs a state diagram to develop each scene which is relatively more complicated. In Ref. [13], Anastasi et al. presented service-oriented architecture (SOA) middleware for QoS configuration and management of WSN. Also, it signed a contract negotiation scheme based on service level agreements (SLA) to ensure the QoS. However, the calculation of the QoS parameter is not discussed in detail.

The work in this paper mainly focuses on the QoS computation of IoT compose services. The authors put forward an effective decomposition and optimization method of total QoS metrics, and find an available algorithm to select services with QoS constraints in IoT environment. The algorithm is faster enough to fit the real-time requirement in IoT.

3 QoS computation

Composite service is composed of multiple simple and available services, which combine according to certain rules and form the new service. The QoS metrics of these simple services are different from each other. Therefore, the system needs to break down the total QoS metrics of the composite services to their simple services and calculate the corresponding range of indicators.

According to the different workflow, the mode of service composition can be divided into four basic forms, which are serial mode, parallel mode, branch mode and circulation mode such as shown in Fig. 1. More modes supported by Web services business process execution language (WSBPEL) can be composed using these basic types of models. If s is replaced by the sensor and network in Fig. 1, the four basic modes can be used in QoS calculation of IoT.

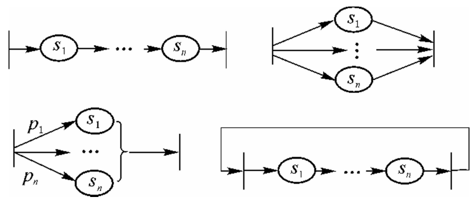

Fig. 1 Four kinds of composite modes

The four composite modes' QoS are calculated using the following methods:

$$G = \{V, E, Q\} \tag{1}$$

$$V = \{s_i\};\quad i = 1,2,\ldots,n \tag{2}$$

$$E \subseteq V \times V \tag{3}$$

$$Q = \{q_t, q_r, q_a\} \tag{4}$$

G is the WSN directed graph of WSN. In V, s_i is the calculation unit of WSN generally including a single sensor and its connecting network. E is the connection between each s_{i}. Q is the QoS metrics. q_{t} is the response time in milliseconds. q_{r} is the reliability indicator, the value range is [0,1]. q_{a} is the availability indicator, the value range is [0,1]. The values of q_{t}, q_{r}, q_{a} are determined by the specific indicators and transmission of the sensor network.

1) Serial mode

System requirements for the services $\{s_1,s_2,\ldots,s_n\}$ perform according to a certain order.

In this example, the following formulae are used to calculate the indexes.

$$q_{\mathrm{t}}(s) = \sum_{i=1}^{n} q_{\mathrm{t}}(s_i) \tag{5}$$

$$q_{\mathrm{r}}(s) = \prod_{i=1}^{n} q_{\mathrm{r}}(s_i) \tag{6}$$

$$q_{\mathrm{a}}(s) = \prod_{i=1}^{n} q_{\mathrm{a}}(s_i) \tag{7}$$

2) Parallel mode

In the parallel mode, the system requires the services $\{s_1 \ldots s_n\}$ to execute tasks at the same time.

The following formulae are used with parallel modes.

$$q_{\mathrm{t}}(s) = \max\{q_{\mathrm{t}}(s_i)\} \tag{8}$$

$$q_{\mathrm{r}}(s) = \prod_{i=1}^{n} q_{\mathrm{r}}(s_i) \tag{9}$$

$$q_{\mathrm{a}}(s) = \prod_{i=1}^{n} q_{\mathrm{a}}(s_i) \tag{10}$$

3) Branch mode

The detection of $\{s_1,s_2,\ldots,s_n\}$ is the same in branch mode as in the parallel mode. As long as a successful detection is finished, the system can complete the task. This mode can be divided into two models in actual use.

For the first model, if the sensor is energy constrained, it needs to extend the usage time of the WSN as much as possible. It is assumed that each branch is able to perform with probability p_i. Of course, p_i is defined according to user requirements. For example: If the WSN is set in accordance with the minimum time optimization, the probability of a branch with the minimum delay is set to 1, the other branch is 0. When this branch's energy is exhausted, then the minimum latency of the branch probability is set to 1. Also, it is defined that the value of p_i in accordance with the round-robin method.

These are calculated using the following formulae:

$$\sum_{i=1}^{n} p_i = 1$$

$$q_{\mathrm{t}}(s) = \sum_{i=1}^{n} p_i q_{\mathrm{t}}(s_i) \tag{11}$$

$$q_{\mathrm{r}}(s) = \sum_{i=1}^{n} p_i q_{\mathrm{r}}(s_i) \tag{12}$$

$$q_{\mathrm{a}}(s) = \sum_{i=1}^{n} p_i q_{\mathrm{a}}(s_i) \tag{13}$$

The second model gives priority to the availability and reliability indicators. $\{s_1,s_2,\ldots,s_n\}$ are executed at the same time. When the first detected data is received, the task is completed and reported to the system. The disadvantage of this method is a serious waste of sensor energy and network bandwidth.

These are calculated using the following formulae:

$$q_{\mathrm{t}}(s) = \min\{q_{\mathrm{t}}(s_i)\} \tag{14}$$

$$q_{\mathrm{r}}(s) = 1 - \prod_{i=1}^{n} (1 - q_{\mathrm{r}}(s_i)) \tag{15}$$

$$q_{\mathrm{a}}(s) = 1 - \prod_{i=1}^{n} (1 - q_{\mathrm{a}}(s_i)) \tag{16}$$

4) Circulation mode

The services $\{s_1,s_2,\ldots,s_n\}$ are running in series and each cycle time is performed according to cycle time L.

These are calculated using the following formulae:

$$q_{\mathrm{t}}(s) = L\sum_{i=1}^{n} q_{\mathrm{t}}(s_i) \tag{17}$$

$$q_{\mathrm{r}}(s) = \left(\prod_{i=1}^{n} q_{\mathrm{r}}(s_i)\right)^{L} \tag{18}$$

$$q_{\mathrm{a}}(s) = \left(\prod_{i=1}^{n} q_{\mathrm{a}}(s_i)\right)^{L} \tag{19}$$

In general, the formula of QoS metrics of composite services and WSN are calculated based on the breakdown of the four methods described above. In each specific instance, the types of WSN sensors are varied, so the QoS parameters of the different sensors need to be determined separately.

4 QoS-aware service selection algorithm

4.1 The QoS definition of combination service

Define 1 Atomic services provide the basic functions of services. In the IoT environment, an atomic service can contain multiple sensors.

Define 2 Composite service is composed of multiple atomic services, which combine according to certain rules and form the new service. Atomic services are relative to composite services, and a composite service in another composite service may be an atomic service.

Define 3 Many atomic services which have the same functionality but different QoS extract the common characteristics to form an abstract service.

Composite service consists of n components of abstract service. $s=\{s_1,s_2,\dots,s_n\}$

Each abstract service s_i binds m atomic services s_i, $s_i=\{s_{i1}, s_{i2},\dots,s_{im}\}$.

The selected variables of abstract service s_i is $D_i=\{d_{i1}, d_{i2},\dots,d_{im}\}$, $d_{ij}\in\{0,1\}$, binding the kth atomic service, $d_{ik}=1$, $d_{ij}=0, j\neq k$.

The QoS metric of atomic service is $q_s=\{q_{s1}, q_{s2},\dots,q_{sr}\}$.

The composite service has r QoS metrics. $Q=\{q_1, q_2,\dots,q_r\}$.

And its QoS weights is $W=\{w_1, w_2,\dots,w_r\}$, $\sum_i w_i = 1$.

The composite service has r QoS constraints. $C=\{C_1, C_2,\dots,C_r\}$. These constraints are generally constant. If a QoS constraint does not exist, the $+\infty$ take the place of it.

The implementation of composite service is $s=\{s_1, s_2,\dots,s_n\}$.

$$\left.\begin{aligned}&\max\{\text{QoS}(s)\} = \max\{W\times Q^T\}\\&\text{s.t.}\\&\quad Q^T \leqslant C^T\end{aligned}\right\}\tag{20}$$

The validation process of composite service s, in fact, is an abstract service binding atomic service process. The problem is a typical NP problem. Two commonly algorithms are often used: integer programming (IP) and genetic algorithm (GA) [14]. There are many ready-integer programming solver that can be used. However, when the difference is large between the optimal solution and the original solution, the convergence process will become very slow, and it is also difficult to deal with large-scale processing. The genetic algorithm is a heuristic algorithm, which finds an approximate solution through continuous evolution. The method of iterations and time is relatively large, which is not suitable for real-time solving process. The algorithm is adopted with an acceptable cost, to find the sub-optimal service under the QoS constraints. It makes more sense to spend a high price to obtain the optimal solution. Based on this idea, a heuristic backtracking algorithm (BT) is used in this paper.

Algorithm 1 Heuristic backtracking algorithm

```
Input: Abstract service set AS. Atomic services set CS.
Output: The Atomic service set to meet the QoS.
Begin
    i=0;
  AbstractService   AS[];//Abstract service set
  AtomService    CS[][];//Already sorted set of atomic services.
  QoSLimit       QL; //QoS constraint
  AtomService    RS[];//The Atomic service   set to meet the QoS
  int   Tag[];//The sequence number to the atomic services.
  int i=j=0;
  bool endflag = true;
  while(!endflag && i<AS.length) {
      if(CalculateQoS(AS,CS,Tag) < QL){
        j=Tag[i];
        RS[i] = CS[i][j]; // The current atomic services to meet the
QoS.
        i++;
      } else{
        i--;
        Tag[i]++;
      }
  }
  return RS[];
End
```

First, the QoS of all the atoms services are calculated according to Eqs. (1)~(20), and the results are queued according descending order in each abstract service. Then, the system used backtracking search algorithm to test the atomic services of each abstract service, and sought solutions to meet the QoS constraints. If not, just to test the next atomic services of the prior abstract service. Finally, the solution is got to meet the conditions.

4.2 Analysis of experimental results

In this experiment, the authors calculated the classic integer programming, genetic algorithm and backtracking algorithms, and compared their solving time and error. The test environment is: Quad-Core AMD Opteron (TM) Processor 2378, 2 GB memory.

The candidate atomic services of each abstract service are 20. The QoS indicators are three and their weights are $w_1=0.4$, $w_2 = 0.3$, $w_3 = 0.3$. Each group randomly generated 100 instances, and test the average solution time. In addition, the definition of a relative error ratio $\eta = (Q_{\text{opt}}$

– Q_{now}) / Q_{opt}, is used to compare suboptimal solution and optimal solution. In the Table 1, we also compares the computational time of the three algorithms.

Table 1 Analysis of the QoS results

No.	IP/ms	GA/ms	BT/ms	Optimized rate/(%)	Relative error/(%)
2	3.2	3.3	3.6	− 11	0
4	7.9	6.1	5.7	38	0.13
6	14.2	25.2	7.3	94	0.34
8	79.4	37.4	10.4	663	0.84
10	445.2	55.2	15.4	2 790	1.84
12	1 071.3	89.3	18.6	5 659	1.96
14	3 894.5	149.8	22.1	17 522	2.08
16	12 467.4	233.7	29.8	41 737	2.14
18	58 450.2	284.5	33.2	175 954	2.34
20	125 182.6	364.3	41.7	300 098	2.5

In Fig. 2, the ordinate is the time in *ms*. The horizontal axis represents the number of combinations of abstract service. The optimal solution given by integer programming is used as a reference to calculate the error of backtracking algorithm. As shown in Fig. 2, with the increase of the abstract service scale, the computer time of integer programming increases exponentially, the time of genetic algorithm increases faster and the backtracking algorithm requires relatively slow growth of computer time. Compared with the optimal solution of integer programming, the error of backtracking algorithm increase slowly with the number of abstract services grows.

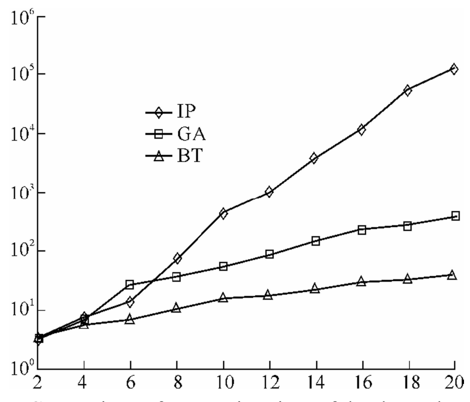

Fig. 2 Comparison of computing time of the three algorithms

As shown in Table 2, we can see that compared to the other two algorithms, backtracking algorithm is more suitable for large-scale services and real-time calculation of QoS. Therefore, we chose the backtracking algorithm as the method of calculating the QoS of IoT compose service.

Table 2 Comprehensive comparison algorithm

Algorithm	Efficiency	Solution	Service scale
IP	Low	Optimal	Small
GA	Normal	Better	Medium
BT	High	Normal	Large

5 Conclusions

In this article, the QoS requirements of IoT composite services is integrated, and it is put forward that effective decomposition and optimization method of total QoS metrics. The authors break down the complex QoS calculation model into four basic models, and gave each model a computational method. In addition, a QoS calculation method is proposed which suited for IoT characteristics, and made analysis and comparison with IP and GA algorithm. As shown in the results, the algorithm in this paper is more suitable for large-scale IoT services. Further research is needed to improve the calculation algorithm of the QoS of IoT compose service in a variety of application scenarios. The uncertainty analysis of QoS is the next focus of the research.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61003282), and the National Basic Research Program of China (2009CB320505).

References

1. Ashton K. That 'internet of things' thing. RFID Journal, 2009
2. Gubbi J, Buyya R, Marusic S, et al. Internet of things (IoT): a vision, architectural elements, and future directions. Future Generation Computer Systems, 2013, 29(7): 1645–1660
3. ISO 8402. Quality management and quality assurance – vocabulary. 2003
4. ISO 9000. Quality management systems–fundamentals and vocabulary. 2005
5. ITU-T E.860. Framework of a service level agreement. 2002
6. Lee K, Jeon J, Lee W, et al. QoS for Web services: requirements and possible approaches. W3C Working Group, 2003
7. Cardoso J, Sheth A, Arnold J, et al. Quality of service for workflows and web service processes. Web Semantics: Science, Services and Agents on the World Wide Web, 2004, 1(3): 281–308
8. Maximilien E, Singh M. A framework and ontology for dynamic Web services selection. IEEE Internet Compute, 2003, 8(5): 84–93
9. Hwang S Y, Wang H J, Tang J, et al. A probabilistic approach to modeling and estimating the QoS of web-services-based workflows. Information Sciences, 2007, 177(23): 5484–5503
10. Li Z, Yang F C, Su S. Fuzzy multi-attribute decision making-based algorithm for semantic Web service composition. Journal of Software, 2009, 20(3): 583–596
11. Bhuyan B, Sarma H, Sarma N, et al. Quality of service (QoS) provisions in wireless sensor networks and related challenges. Wireless Sensor Network, 2010, 2(11): 861–868
12. Heinzelman W, Murphy A, Carvalho H, et al. Middleware to support sensor network applications. IEEE Network, 2004, 18(1): 6–14
13. Anastasi G F, Bini E, Romano A, et al. A service-oriented architecture for QoS configuration and management of wireless sensor networks. Emerging Technologies and Factory Automation (ETFA) IEEE Conference, 2010, 1–8
14. Canfora G, Penta M D, Esposito R, et al. An approach for QoS-aware service composition based on genetic algorithms. Proceeding GECCO'05 Proceedings of the 2005 conference on Genetic and evolutionary computation, 2005, 1069–1075