*Available online www.jocpr.com*

**Journal of Chemical and Pharmaceutical Research, 2014, 6(5): 536-539**

**Research Article**

**ISSN : 0975-7384**
**CODEN(USA) : JCPRC5**

# The resources monitoring and optimization of scheduling research based on the cloud computing environment

**Shengjun Xie**

*Campus network management center, Southwest Univ. for Nationalities, Chengdu, 610041, China*

---

**ABSTRACT**

*This article focus on the problems during the performance degradation of SaaS software, this paper puts forward a cloud computing environment resource monitoring and optimal operation platform, with CPN performance simulation analysis, based on particle swarm optimization (pso) algorithm, keep excellent particle technology and tabu search technology to the cloud resource scheduling and optimization, the industrialization is of the good application value.*

**Key words:** SaaS; cloud computing; PSO

---

## INTRODUCTION

Recent interest in Cloud Computing has been driven by new offerings of computing resources that are attractive due to per-use pricing and elastic scalability, providing a significant advantage over the typical acquisition and deployment of equipment that was previously required. The effect has been a shift to outsourcing of not only equipment setup, but also the ongoing IT administration of the resources as well. The needed changes to applications, in order to take advantage of this model, are the same as those required for server consolidation – which had already been taking place for several years prior to the advent of the Cloud. The increased resource utilization and reduction in power and cooling requirements achieved by server consolidation are now being expanded into the cloud.

SaaS (Software - as - a - Service) in the long run, the wide application range, due to many systems platform Service layer and the change of the tenant application layer of a variety of complicated factors, resulting in a decline in the performance of the SaaS Software, may even make a serious loss of SaaS Software Service quality and the loss of availability. [1] A lot of reasons of SaaS software performance degradation condition, such as due to the changes of application and system environment caused by the service resource allocation imbalance, resulting in a decline in the quality of service. Due to the dynamic changing computing environment, the diversity of the network environment, results in the decrease of service reliability. Due to the running environment of hardware resources supply or malfunction, the tenant populations, within a certain period of time the use of an application service or business workflow exception is frequent, etc are likely to trigger a SaaS software performance degradation. In addition to the diversification and complication of user requirements, as well as the use of personalized and flexible, and can also lead to the unpredictability of workload changes.[2]

Monitoring for understanding the behavior of complex systems is necessary. Cloud computing as a large high performance distributed system, needs to be done a lot of monitoring data, such as fault detection, performance analysis, performance tuning and performance prediction and dispatching and other tasks. Monitoring the system work can realize the system running state, timely find the fault of the host, the analysis of system performance bottlenecks, and help the user find the error in the most short time ask for fault location, restore or adjust system, and achieve higher performance indicators as well. [3] Control the operation of the service object in the distributed

system, adjust the system to run. Monitoring data can also be used to predict trajectory of the system operation, dynamic load balancing strategy for cloud computing platform to provide reliable basis. In order to improve the reliability of cloud computing platform, ensure the quality of service, must be in the cloud computing platform real-time monitoring mechanism are introduced to understand the running status of the platform, make sure the platform is abnormal when the effect of early warning.

The resources of the cloud computing environment monitoring and optimization dispatching platform is designed to meet these requirements, the main task in general is monitoring information collection and based on the monitoring data analysis results of system maintenance, management and optimization. [4] Greatly improve the efficiency of cloud computing resources, improve the operational efficiency of the cloud computing resources, energy efficiency is improved. Let cloud computing users only through a management platform to manage, monitor all areas of cloud computing resources, unified control and operation, the use of monitoring and early warning mechanism to ensure the availability of the platform, has important academic significance. [5]Project research motivation comes from the project team currently bear the "fruit and vegetable farmers and supply chain of cloud services platform", "food safety of agricultural products and logistics back cloud services platform" SaaS software research and development projects, such as has urgent demand. Project research can solve the problem of performance degradation of SaaS software, optimization of system resource usage, provide higher quality service, has important application value.[6]

Along-running SaaS software often appear performance degradation (performance degradation). To solve the problems during the performance degradation of SaaS software, this paper puts forward a cloud computing environment resource monitoring and optimal operation platform, with CPN performance simulation analysis, based on particle swarm optimization (pso) algorithm, keep excellent particle technology and tabu search technology to the cloud resource scheduling and optimization. [7]Current system deployment optimization algorithm research work mostly on the platform layer (layer of cloud computing PaaS), in the SaaS service application layer deployment optimization algorithm research work less. And these work mainly adopts the traditional evolutionary algorithm, there exists a problem of premature convergence in the algorithm, does not guarantee to find the optimal deployment plan.

**OPTIMIZATION BASED ON PSO**

Cloud computing resource scheduling and optimization scheme can directly affect the performance of the software, through the analysis of the performance study of cloud computing resources and resource optimization scheduling model, revealing the resources optimization dispatching on the performance of the software response time, throughput and utilization effect mechanism, to form a SaaS software method of scheduling performance modeling and optimization theory.[8]

Based on particle swarm intelligence algorithm to predict the characteristics of monitoring data analysis, to the service response time, system throughput, resource utilization, three software performance indexes as a scheduling scheme take a weighted average of the three optimization index as the objective function. [9] To service component resource demand requirements, computing resources, storage resources constraints, parameter value scope requirements mainly constitute model constraints. With the improved particle swarm algorithm to solve the model, the result directly affect the task of improving the method and model of optimal operation. Through performance analysis, performance tuning, performance prediction and dispatching contain a variety of tasks, such as to get to the performance of the system of information analysis, to found that the system performance bottleneck on the basis of improving the system performance. [10]

Based on PSO:

Assuming a D-dimensional search space, there are N particles to form a community in which the ith particle is represented as a D-dimensional vector

$$X_i = (x_{i1}, x_{i2}, \cdots, x_{iD}), \ i = 1,2,\cdots,N.$$

"Flight" velocity of ith particle is also a D-dimensional vector, denoted

$$V_{\mathrm{i}} = (v_{i1}, v_{i2}, \cdots, v_{iD}), \ i = 1,2,\cdots 3.$$

Particle i found so far is called the optimal location of individual extreme, denoted

$p_{best} = (p_{i1}, p_{i2}, \cdots, p_{iD})$, $i = 1,2,\cdots,N$.

So far the search to the entire particle swarm optimal location for the global optimum, denoted

$$g_{best} = (p_{g1}, p_{g2}, \cdots, p_{gD})$$

Find the optimal values of these two particles to update their velocity and position according to the following equation (1) and (2):

$$v_{id} = w * v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

Where $c_1$ and $c_2$ are learning factors, also known as the acceleration constant; $r_1$ and $r_2$ are uniform random numbers in the range of [0,1]. Equation (1) on the right consists of three parts, the first part is "inertia" or "momentum" section, reflecting the movement of the particles' "habits", on behalf of the particles have a tendency to maintain their previous speed; the second part is "cognitive" section, reflecting the particle memory of their own historical experience or remembrance, on behalf of the particle has its own historical best position to approach the trend; the third part is "social" section, reflecting the collaboration between the particles of historical experience and knowledge sharing groups, on behalf of the particles to the community or neighborhood has the best location close to the historical trend, based on usual experience, $c_1=c_2=2$. i =1, 2, …, D. $v_{id}$ Is the velocity of the particle, $v_{id} \in [-v_{\max}, v_{\max}]$, $v_{id}$ is a constant set by the user to limit the speed of the particles. $r_1$ and $r_2$ are the random numbers between [0,1].

Algorithm process is as follows:

① initialized particle swarm, including population size N, position $x_i$ and velocity $V_i$ of each particle;

② calculate the fitness value $F_{it}[i]$ of each particle;

③ for each particle, comparing its adaptation value $F_{it}[i]$ with individual extremum $p_{best}(i)$, if $F_{it}[i] > p_{best}(i)$, with $p_{best}(i)$ replaced by $Fit[i]$;

④ for each particle, comparing its adaptation value $F_{it}[i]$ with its global maximum $g_{best}$, if $F_{it}[i] > p_{best}(i)$, with $g_{best}$ replaced by $F_{it}[i]$;

⑤ According to the formula (1) and(2), update the particle velocity and position;

⑥ If the end condition is met (error is good enough or reaching the maximum cycles) ,exit, otherwise return ②.

## EXPERIMENTAL SECTION

### EXPERIMENT AND ANALYSIS

In order to make test result reflect the characteristics of monitoring data analysis based on the particle swarm intelligence algorithm, this test fetched user survey data from project team currently bear the "fruit and vegetable farmers and supply chain of cloud services platform", "food safety of agricultural products and logistics back cloud services platform". Design data include classification, content and profile sample data set. Classification, content They are used for combining web data and analyzing hidden relationship of web users in order to meet requirements of user classification calculation in trust distinguish based on evolutionary game. Data file can be used for calculating selection model and parameter value of trust evolutionary trust function strategy probability transfer rate λε. Therein, classification sample files store characteristic information with evaluation data source in digital mode, according to requirements of our test, it includes 4 general varieties of type, time, country/region and other and detailed classification is given to each general variety.

**Table 1 NGWTCN application indexes**

| 1st grade indexes | 2nd grade indexes | 3rd grade indexes | Utilization rate |
|---|---|---|---|
| Trust application index of next generation weight trust control network (62.16) | Trust information obtaining index (76.7) | Trust online news | 80.1 |
| | | Trust search | 73.3 |
| | Trust entertainment index (71.7) | Trust online music | 83.5 |
| | | Trust online games | 68.9 |
| | | Trust online videos | 62.6 |
| | Trust interactive participation index (58.0) | Trust instant messaging | 70.9 |
| | | Trust E-mail | 56.8 |
| | | Trust blog update | 37.7 |
| | | Trust posting/reply | 30.7 |
| | Trust consumption index (25.2) | Trust online shopping | 28.1 |
| | | Trust online payment | 24.5 |
| | | Trust travel booking | 7.9 |

**Table 2 Comparison of NGWTCN application indexes**

| Index grade | Index name | Before using model | After using model |
|---|---|---|---|
| 1st grade index | Network source trust application index | 54.82 | 62.16 |
| 2nd grade indexes | Trust information obtaining index | 70.1 | 76.7 |
| | Trust entertainment index | 68.4 | 71.7 |
| | Trust interactive participation index | 45.5 | 58.0 |
| | Trust consumption index | 17.3 | 25.2 |

## CONCLUSION

SaaS software monitoring has an important influence on its performance. How to adaptively optimize the SaaS software deployment, improve service performance and is an important academic significance and application value. The improved algorithm achieve higher performance index and help the users for the recovery or adjusting system in the most short time.

**Acknowledgments**
The authors wish to thank “the Fundamental Research Funds for the Central Universities” of Southwest Univ. for Nationalities for contract 2014NZYQN48, under which the present work was possible.

## REFERENCES

[1] M. Dorigo and M. Birattari, “Ant colony optimization”, in Encyclopedia of Machine Learning, pp 36–39,Springer, **2010**.
[2] M. Dorigo and T. Stützle, “Ant colony optimization: overview and recent advances”, in Handbook of metaheuristics, , pp 227–263, Springer, **2010**.
[3] T. Stützle, M. López-Ibánez, P. Pellegrini, M. Maur, M. M. de Oca, M. Birattari, and M. Dorigo, “Parameter adaptation in ant colony optimization”, in Autonomous Search, pp 191–215,Springer, **2012**.
[4] M. Dorigo and G. Di Caro, “Ant colony optimization: a new meta-heuristic”, in Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, v. 2,**1999**.
[5] Z. Shanshan, W. Lei, M. Lin, and W. Zepeng, “An Improved Ant Colony Optimization Algorithm for QoS-Aware Dynamic Web Service Composition”, in Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on, pp 1998–2001, **2012**.
[6] Y. Li, W. Q. Tong, X. L. Zhi, and D. Ding, *Applied Mechanics and Materials,* vol 182, pp 2136–2141, **2012**.
[7] Z. Yang, C. Shang, Q. Liu, and C. Zhao, *Journal of Computational Information Systems*, vol 6, no 8, pp 2617–2622, **2010**.
[8] O. Baskan, S. Haldenbilen, H. Ceylan, and H. Ceylan, *Appl Res Comput*, vol 27, pp 573–576, **2010**.
[9] X. Tang, C. Jiang, and M. Zhou, *Expert Systems with Applications*, vol 38, no 10, pp 13024–13031, **2011**.
[10] Emir Imamagic, Dobrisa Dobrenic. Grid infrastructure monitoring system based on Nagios [C]. In: Proceeding s of the 2009 workshop on Grid monitoring. USA: ACM Press, 84, **2009**.