International Journal of Applied Computer Technology and Information Systems: Volume 4, No.1, April 2014 - September 2014

Task Allocation for Service Provider using Difference Equation Work Load Status Consideration

Somchai Prakancharoen
Department of Computer Science and Information
Faculty of Applied Science King Mongkut's University of Technology North Bangkok
1518 Pracharaj-1 Bangsue Bangkok Thailand
spk@kmutnb.ac.th

Nimol Dith
Audio Visual Unit, Court Management Section
Extraordinary Chambers in the Courts of Cambodia (ECCC)
National Road 4, Chaom Chau Commune, Porsenchey District, Phnom Penh, Cambodia
dith.nimol@eccc.gov.kh

***Abstract*—Task allocation algorithm topic in distributed computing system, such as cloud computing, is now an interesting issue. Service Manager (SM) has to consider which Service Provider (SP) is suited to service data processing or even data storage for a Service Request (SR). Normally, SP also has to handle its own tasks thus new allocated tasks should not consume more CPU's workload status which could make SP decrease its base performance in processing. This research tries to invent the task allocation algorithm by studying the status of the CPU usage according to the time period then transforming it to a Transition Probability (TP) matrix. This TP should be used to predict the SP workload status, using the Markov chain technique at a period of given time. These information were transformed to difference equations in order to simplify calculation handlings. Another information was gradient of SP's work load which should trajectory the increasing or decreasing of SP's workload. These information were used to make a decision by SM, whether this SP is suited to service request from SR or not. The experiment is presented in practical calculations. It could point out that some request of SR is more suitable for sending to which one of SPs.**

Keywords – Task allocation algorithm; distributed system; difference equation; gradient

I. Introduction

Service Manager: SM is a server which is responsible for decision making in sending task from Service Request (SR), which is initiated from a specific client-SR, to be processed by a suitable Service Provider (SP). There are many designed algorithms used to help SM make a good decision. A good algorithm should utilize CPU, RAM or even Data storage of SP facilities usage. The allocated task may not tightly disturb or corrupt SP's ordinary its own handled tasks. A good algorithm should also support SR in task service processing with rapid response-finishing time.

The objective of this research is to design a simple concept of task allocation from SR to SP under consideration by SM.

II. Related Research and Theory

A. Related Research

First come first served [1] is a technique that each task is allocated to one of SPs that is managed in queue of services. This algorithm is the easiest one but it may cause a serious problem to SP which is in a very busy state and could delay the finish time of that new allocated task processing. This method is suitable for SP that is mostly available to service provision.

Ant colony [2] is an optimization technique that is applied to task allocation algorithm. Application is split into many tasks which are considered for distribution to cloud of SP under a behavior migration of each SP. This algorithm is designed to prevent work overloading which may occur on candidate SP. This technique is very complex and designed to consume plenty of processing time resulting in a bottle neck at SM.

Load balancing [3] is the most presented algorithm which tries to study SP's prior knowledge of its workload so that SM could use such information to consider which SP is an optimized SP in processing a request task of SR.

The theme for good task allocation algorithms is not yet finished. It is still being invented and contributed by many researchers. Thus, this research is also following the same direction in relation to [2] and [3] but with an objective of reducing its complexity to become a simple task allocation algorithm. A Markov chain technique is chosen to be a major technique for this research.

B. Markov Chain Technique

Markov chain [4] is a first order of differential equation. The states of some event in time "t" are changed to new state at time "t+1". This state changing of all associated events must be defined in the Transition Probability (TP) matrix. Typically, all of the states covered in each event must be calculated so that the result is equal to the value of "1.0". The

Markov chain is used to represent the behavior of each event that is changed during its life time. However, the state of each event should become a fixed value and not be subject to any change as time goes by. This point is called Steady Point. For example, event A and B are in their own state until transferred to another event as shown in table I.

TABLE I. Transition Probability or P Matrix

Time=t	Time=t+1	
	A	B
A	P_{AA}	P_{AB}
B	P_{BA}	P_{BB}

P_{AA} is a probability of state transformation from event A to B. If the start state of A is [1 0] row matrix, then state of A at time t+1 will be calculated by multiplying the start state matrix with the transition state matrix.

$$(1 \quad 0)\begin{pmatrix} P_{aa} & P_{ab} \\ P_{ba} & P_{bb} \end{pmatrix}$$

If we would like to find out the next state the previous result row matrix will be used to repeat the calculation of time period by time period. On the n^{th} month in which the value of its state is not changed, this point is called a Steady State, or the expected probability of all events in this application domain.

C. Difference equation [5]

Markov chain analysis is calculation time consume and do not show the direction of SP's workload situation. Markov chain then was reform to be difference equations in order to eliminate previous time element, as shown in equation (1),(2)

$$y_t = \alpha y_{t-1} + \beta \tag{1}$$

$$y_t = \alpha^t * A + B \tag{2}$$

D. Gradient [5]

Gradient is the slope of equation or time series equation. It can be calculated from $\frac{dy_t}{dt}$. If the gradient value is positive this mean that y is increasing status according to t and decreasing if gradient value is negative. If the gradient value is equal to '0' then it is in stationary state.

E. Criteria of Decision

SM has to define a criteria that could be used to determine which SP is best allocated for the required task. This research suggests some condition that represents a workload status on each SP. If average fraction of SP's load situation on fair and busy is larger than some specific threshold such as 0.50 then this SP was appreciated to be a candidate in task allocated. Another reason on decision improvement was the gradient or slope of SP's work load, if the value of it is negative this inform SM that SP's work load is in decreasing state. It should be good if the fraction of SP's load situation on fair & busy state is larger and its gradient value is negative.

III. Task Allocation Algorithm Design

The designed algorithm of task allocation begins with related data preparation, calculation of SP information then making a decision.

A. Definition of CPU Usage

In experiment for this research, CPU usage is categorized in three statuses: Ready (R), Busy (B) and Very Busy (VB). The status is defined as shown in table II.

TABLE II. SP Status Definition

SP Status	CPU-performance
Fair:F	0≤25%
Busy:B	26%<and≤60%
Overload:O	61%<and≤100%

B. Transition Probability Matrix

As an example in this research, we assume the SP-status data of SP. The SP status is collected from a period of time t_0 to t_n which is numbered from t_0 to t_{59} or about 60^{th} minutes. The average values of the SP-state transition probability are presented in table III.

TABLE III. SP- StateTransition Matrix (or P) of SP

SP	F	B	O
F	0.2	0.7	0.1
B	0.5	0.4	0.1
O	0.1	0.6	0.3

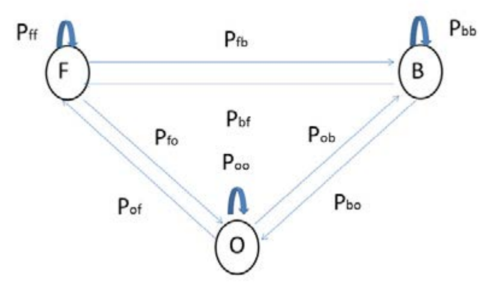

Figure 1 The Makov model of SP state

While, for example, p_{ff} represents a transition probability of transition from state f to state f (equal to 0.2 as in table 3).

C. SP's status difference equation

SP's status has to be calculated according to expected allocation time such as during start time: t_r though stop time: t_s. Difference equation was used to find out these by summation each piece of time probability load status as shown in equation (3),(4) and (5) for 'Fair', 'Busy' and 'O' respectively.

$$F_t = \alpha_F{}^t * A_F + B_F \quad (3)$$

$$B_t = \alpha_B{}^t * A_B + B_B \quad (4)$$

$$O_t = \alpha_O{}^t * A_O + B_O \quad (5)$$

D. F&B fraction

In this research, the status of SP that summed state in fair and busy should be considered as appropriate condition to allocate task for it. That is the fraction of F&B ought to be a large proportion. Since F&B fraction were summed from s to r piece of status so the average value of this fraction will be further used.

$$\overline{FB}_{tr-ts} = \frac{1}{(ts-tr)} \sum_{t=r}^{t=s} (F_t + B_t) \quad (6)$$

It should be find out O – fraction from equation (7).

$$\overline{O}_{tr-ts} = \frac{1}{(ts-tr)} \sum_{t=r}^{t=s} (O_t) \quad (7)$$

The last equation (8) represent fraction of F & B from all fractions.

$$\overline{F \,\&\, B}\ fraction = \frac{???}{??? + ???} \quad (8)$$

E. F&B gradient

Gradient of (8) was defined from (9).

$$Gradient = \frac{d}{dt}[\overline{FB}_{t11-t15}] \quad (9)$$

F. Practical Example

Transition probability matrix-P from table 3 was used to construct difference equations. The numerical results were presented as following details

Determinant = -16.67

Eigen values λ_1=5, λ_2=-3.33, λ_3=1.

Eigen vector for λ1, λ2, λ3 were (0.14, 0.14, -9.8), (0.69, -0.56, 0.44), (0.57, 0.57, 0.57) respectively.

Diagonal matrix $\Lambda = \begin{pmatrix} 5 & 0 & 0 \\ 0 & -3.33 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. And the matrix C which composed of corresponding eigen vectors (C) is

$$C = \begin{pmatrix} 0.14 & 0.69 & 0.57 \\ 0.14 & -0.56 & 0.57 \\ -0.98 & 0.44 & 0.57 \end{pmatrix}.$$

According to start SP's probability status vector (Π_0) as shown here.

$$\Pi_0 = \begin{bmatrix} P_F^0 \\ P_B^0 \\ P_O^0 \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.5 \\ 0.2 \end{bmatrix}$$

The difference equation of SP's status F, B and O were presented as equation (10), (11) and (12).

$$F_t = \alpha_F{}^t * A_F + B_F \quad (10)$$

$$B_t = \alpha_B{}^t * A_B + B_B \quad (11)$$

$$O_t = \alpha_O{}^t * A_O + B_O \quad (12)$$

Supposing that tr= 11 and ts = 15 time unit in second then the average F&B status was $\overline{FB}_{t11-t15} = ???$, $\overline{O}_{t11-t15} =$??? and $\overline{FB}_{t11-t15} = \frac{1}{(15-11)} \sum_{t=11}^{t=15} (\overline{F}_{tr..ts} + \overline{B}_{tr..ts})$

$$= \frac{1}{(15-11)} \sum_{t=11}^{t=15} (??? + ???)$$

Thus the result of average F&B fraction was

$$\overline{F \,\&\, B}\ fraction = \frac{\overline{FB}_{t11-t15}}{\overline{FB}_{t11-t15} + \overline{O}_{t11-t15}}$$

$$= \frac{???}{??? + ???}$$

The last interested feature is the slope or gradient of F&B status. This value could be found from

$$Gradient = \frac{d}{dt}[\overline{FB}_{t11-t15}]$$

$$= ???$$

SP’s status represented by average about F&B, O, F&B-fraction and F&B fraction were shown in table 4. In case of cloud computing scheme, other candidate SPs should be compiled in the same calculations so that SM could easily make decision in SPs task allocation selection.

TABLE IV. SP’s Congestion index On Average Status

Item(average)	SP#1	SP#2*
F&B	???	…
O	???	…
F&B fraction	???	…
Gradient/slope	???	…

(*note: the result were calculated from unshown CPU performance of candidate SP#2)

G. SM Decision Making

After table 4 considering, SM has to make decision which was the best SP to allocate task for it. SP#1 has larger F&B fraction while its gradient was decrease thus SP#! Was a better one than SP#2 that has small value of F&B fraction and a increasing gradient

IV. Summary and Suggestion

The simplified design of task allocation algorithm in this research is based on the difference equation and gradient technique. The transition probability matrix must be predefined if there are available data from each SP. SM is responsible for performing such activities.

In case of SM complicate decision making, the gathered feature should be used as primary or hidden causes in visible decision activity. Thus hidden makov model (HMM) might be chosen to more practical operation in further research.

References

[1] Armstrong P., Cloud scheduler: a resource manager for distributed compute clouds, University of Victoria, Canada, 2010.

[2] Ratan Mishra, Ant colony optimization: A solution of load balancing in Cloud, International journal of Web& semantic technology vol.3 no 2. April, 2012.

[3] Nidhi jain kansal, Existing load balancing techniques in cloud computing: a systematic review, Journal of information systems and communication, vol.3 issue 1, 2012.

[4] David Anderson, Markov chains, University of Wisconsin Madison, 2013.

[5] Earl D. Rainville,Phillip E. Bedient,Richard E., Elementary differential equation edition 8, Prentice Hall, 1996.