Advanced Science and Technology Letters
Vol.42 (Mobile and Wireless 2013), pp.89-94
http://dx.doi.org/10.14257/astl.2013.42.21

A Load Balancing Scheme for Games in Wireless Sensor Networks

Hye-Young Kim[1]

[1] Major in Game Software, School of Games,
Hongik University,
Chungnam, Republic of Korea
hykim@hongik.ac.kr

Abstract. We propose load balancing scheme for games in wireless sensor networks which is based on allocate load on wireless sensor nodes proportionally to each of the agent's capacity. This proposed scheme is combined dynamic provisioning algorithm based on greedy graph and user-oriented load balancing scheme for maintain of the performance and stability of distributed system in wireless sensor networks.

Keywords: load balancing, greedy graph algorithm, user-oriented, hot-spot, wireless sensor networks

1 Introduction

A sensor network consists of a large number of tiny sensors with limited computation, memory space of storage, communication range, and battery power.[1] Without load balancing, a coordinator with excessive number of mobile modes may lead to more energy consumption and experiences data reception delay as compared to the other coordinators. In a wireless sensor networks, the frequent changes in large scale network topology lead to a challenge for delay sensitive data reception. When a the nodes which are associated with the same coordinator send data at the same time, there will be a delay or a failure of data received by the coordinator. Wireless sensor nodes have maximum utility when they can be used "anywhere at any time"[2]; additionally, some of the ability to provide a more abundant user experience by running heavy desktop based games. With the explosion of mobile games and user demand, wireless sensor networks are born to integrate cloud networks in to the mobile environment. There is an important issues when consider a load balancing scheme in wireless sensor network.it must be established within a short time to adapt with the change in wireless nodes locations, and the parameter that triggers the action must relate to the changes in the topology.

We show a general architecture of wireless sensor networks as shown in Fig. 1. Mobile nodes typically connect to the access network via various wireless access entrances. There exists a mobile network and mobile cloud platform. The issues of this layer vary from node's mobility control and computation task offloading management

ISSN: 2287-1233 ASTL
Copyright © 2013 SERSC

to security control. Also, wireless sensor networks bring unprecedented opportunities and challenges associated with developing mobile user-oriented specific mobile games. To apply cloud networks to mobile environment, there are still many obstacles involving service availability, mobility, security, privacy and load balancing.

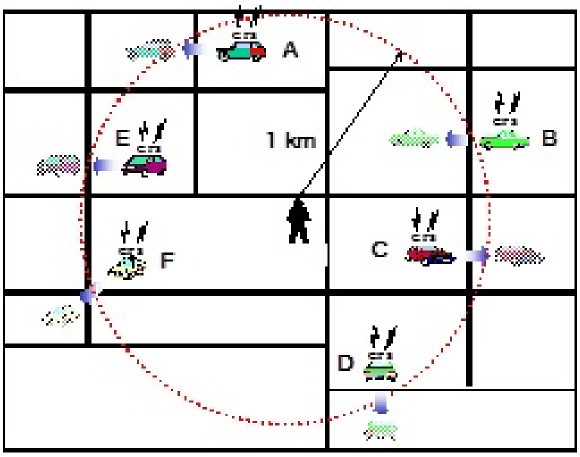

Fig. 1. Architecture of wireless sensor networks

In the wireless sensor networks, the mobile games such as RPG(Role Playing Games) in which a large number of players can interact with each other in the same world at the same time[3], so it is the large number of players interacting simultaneously, reaching the number of tens of thousands [4]. Each of the players remotely controls one or several in game characters called avatars. States of the avatars are simulated by the hosting agent, with updates of the avatars states broadcasted to other hosting agents and clients, so that all avatars can be properly rendered on their client side. For example, if the avatars of two players are too distant from each other, probably there will be no interaction between them and therefore the agent needs only to update each of one with the result of theirs own actions. However, if these avatars are close to each other, each player should be updated not only his own actions, but also of the actions of the other player.[5]

For this reason, it is not enough just to divide the players between agents, even if this division is proportional to the resources of each one of them. First, in some cases the large of the mobile nodes' bandwidth may be square to the number of players, while in others it may be liner in wireless sensor networks. Moreover, there is another important topic is the overhead of the distribution. As the agents need to communicate with one another, there must be a way to minimize this traffic, reducing the waste of resources of the agent system for mobile games must prevent the presence of hotspots from degrading the quality of the game beyond a tolerable limit.

Therefore, we propose a scheme is combined dynamic provisioning algorithm based on greedy graph and user-oriented load balancing scheme for maintain of the performance and stability of distributed system in wireless sensor networks. The rest of the paper is structures as follows. Section 2 describes our proposed architecture and provides the detailed a load balancing scheme for mobile games in cloud networks. Also, section 3 constitutes a summary of our proposal.

 Copyright © 2013 SERSC

2 Proposed load balancing scheme

To define our load balancing approach, we cannot ignore that the traffic generated by player is not simply linear, but square for each cluster of mobile game players. And another pointer is the overhead. This overhead must be taken into account, no matter which load balancing algorithm is being used.

Table 1. Our proposed scheme

```
    cell has a position coordinates(x,y);
    weight_to_divide <- 0;
    free_capacity <- 0;
    for each region R in region_list do
        weight_to_divide    <- weight_to_divide + Wr(R)
          free_capacity <- freee_capacity + P(S(R))
          calulate distance between cells
          S(cx, cy)    <- cell(M)    //s(cx, cy): agent's coordinates, cell(xm, ym):
cell's mean
          cell_list    <- list of cells from R in increasing order of cell`s distance
          while fracCr(R) > fracCp(S(R)) do
                          if cell(x,y) is not adjcent of S(cx,cy)
                            c <- first element from cell_list
                            remove C from R
                            remove C from cell_list
                   endif
          end while
    sort region_list in increasing order of u(S(R))
    for each region R in region_list do
        weight_share <- weight_to_divide*Pp(s(R))/free_capacity
   while Wr(R) < weight_share do
          if there is any cell from R neighboring a free cell then
              R <- R U {neighbor free cell with the highest Intc(C)
     else if there is any free cell then
          if cell adjacent to Agent
          R <- R U {cell}
      else
         R <- R U{heavist free cell}
       endif
    end while
    end for
```

When wireless sensor gaming users are within visual range of each other, it is best to locate them in one mobile agent. To manage wireless sensor gaming users like the above, both the scope of interactions and probable interactions have to be taken into account and then set the neighboring gaming users area to about 1.5~2 times the visual range with the gaming users at the center. All wireless sensor gaming agent or anchors have the same capacity, each one being able to handle same wireless sensor gaming users. When new wireless sensor gaming users process to log-in, first check whether the existing wireless sensor gaming users are in the nearest gaming user area and if they are, then distribute the new wireless sensor gaming users to the nearest wireless sensor gaming user areas. It is considered that an initial section of the virtual environment has already been made. Each mobile agent or agent should then check regularly if there is an imbalance and trigger the algorithm. Although the overhead resulting from the distribution of the virtual environment is part of the workload on mobile agent or agents, there is no way to know it beforehand without executing the repartitioning first.

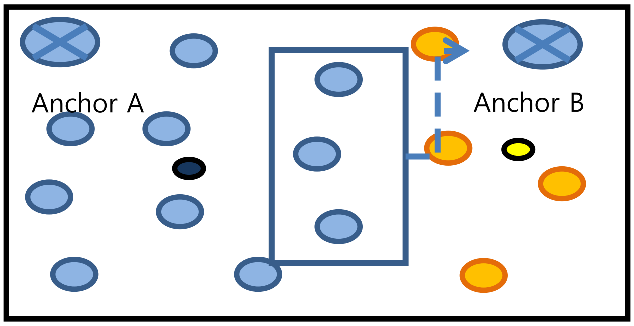

Fig. 2. Selection Anchor of wireless sensor gaming nodes

For load-balancing, each the center of each gaming agent gets updated at the load-balancing agent and this point is equated by the average coordinates of the game users. Load-balancing agents distribute gaming users to the agent which is closest to the center of the agent when new game users log-in and when all gaming users of the game agents are not included in the nearest section. Fig. 2 shows the selection anchor node of the wireless sensor gaming nodes.

The proposed scheme is follows.

- select the group of local regions: the most of the largest cell allocate to the largest agent
- calculate the weight of each of the cell and represent all of the cells to the graph
- allocate coordinate of agent to average of cells
- decision of the cell adjacent of the agent through the coordinate of the cell
- balance these regions, assigning to each one a weight which is proportional to the power of its agent
- refine the partitioning, reducing the overhead

Let us consider N wireless sensor nodes connected to a distributed game session aggregating a total of H regions. Here, we use parameters such as Table 2 for simulating of the load balancing in the wireless sensor networks.

Table 2. The parameters of the simulation

parameters	*Descriptions*
N	number of nodes
H	number of regions

Copyright © 2013 SERSC

parameters	*Descriptions*
I	total number of interactions
BE	number of moving node
AE	number of active entities
C	number of connected wirelss sensor nodes
IC	number of avatas interacting with any other entities
Ti	consumed time for each computation
T_u	update of entity states received from/sent to another nodes
P_{ci}	percentage of CH
P_{ei}	average number of interactions as a percentage of BE
P_{ui}	ratio between time necessary for one entity update

The load of the wireless sensor gaming node, L_{net} is

$$L_{net} = (C*d_{in} + (N-C+BE-AE)*d_{update})/B$$
(B: Bandwidth, d_{update}: bining update of wireless sensor nodes)

In the Fig. 3, we illustrate of the load of the wireless sensor nodes using our proposed scheme. We found that the load is kept constant in the wireless sensor nodes on our proposed scheme.

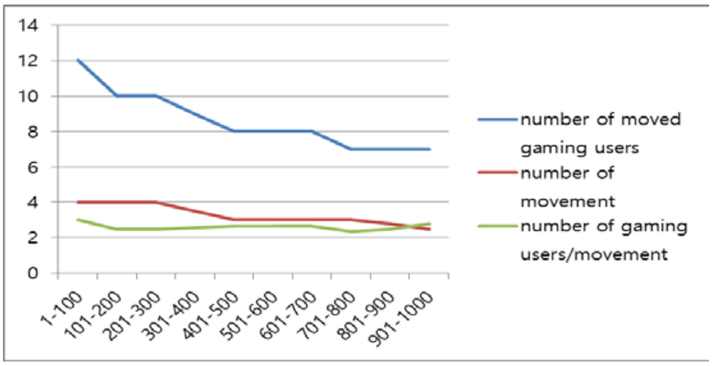

Fig. 3. load of the wireless sensor gaming nodes

3 Conclusion and future works

We propose a scheme which combined dynamic provisioning algorithm based on greedy graph and user-oriented load balancing scheme for maintain of the performance and stability of distributed system in wireless sensor networks. Finally, we simulate the load of the wireless sensor gaming nodes using the parameters in Table 2. We are going to consider a real-world environment through the simulation and performance analysis of our proposed scheme mathematically.

Acknowledgments. This work was supported by the Korea Foundation for the Advancement of Science & Creativity (KOFAC), and funded by the Korean Government (MOE).
This work was supported by 2013 Hongik University Research Fund.

References

1. Kyu-Han Kim, Sung-Ju Lee, Paul Congdon: On Cloud-Centric network architecture for Multi-Diemensional Mobility. In: 1st MCC workshop in mobile cloud computing,pp.1-6, (2012)
2. Pradeep Kumar Sinha, Sunil R Dhore: Multi-agent Optimized Load Balancing Using Spaning Tree for Mobile Services. In: International Journal of Computer Applications, vol. 1, pp.33-40, (2010)
3. Hye-Young Kim: An Efficient Access Control Scheme for Online Gaming Agent. In: Computer Science and Convergence, Lecture Notes in Electrical Engineering 114, pp. 259~267, (2011)
4. G. Schiele et al.: Requirements of Peer-to-Peer-based Massively Multiplayer Online Gaming. In: proceedings of the IEEE International Symposium on Clister Computing and the Grid, CCGRID, 7, pp.773-782, IEEE (2010)
5. Calos Eduardo B. Bezerra, Joao L. D. Comba, Claudio F.R.Geyer: A Fine Granularity Load Balancing Technique for MMOG Agents Using a KD-Tree to Partition the Space. In: Games and Digital Entertainment(SBGAMES), 2009. , pp.17-26, Brazilian Symposium on 2009

 Copyright © 2013 SERSC