A Hybrid P2P-based Rendering System on WAN

Yizhen Cao[1, a], Yilin Lin[2, b] and Yan Wang[3, c]

[1, 2, 3]School of computer, Communication university of china, Beijing, China

[a]caoyizhen@cuc.edu.cn, [b]linyilin1003@163.com, [c]wingstory@hotmail.com

Keywords: Hybrid P2P, Rendering System, Super Peer, Reputation

Abstract. On the basis of cluster rendering, we present a WAN-based distributed animation rendering system. According to the distribution of resource on WAN and the characteristics of computational nodes, this paper presents how to construct a distributed rendering system on WAN. Combining with P2P technology gives a novel solution to solve the problems such as the performance bottleneck in networks and single point of failure. The system presented in this paper divides the network into multiple clusters according to geographical-proximity principle; each cluster elects a super peer by using Reputation-Aware Selection Algorithm. Hence this method can reduce the communication time of rendering on WAN, improve the robustness of rendering network, and ensure good scalability of the system.

Introduction

Animation rendering requires much computation time. To solve the bottlenecks in manufacturing process, the development of rendering technology had passed four phases [1]: CPU-Based rendering, GPU-Based rendering, network rendering, and cluster rendering. The main clue of rendering technology is developing distributed computing which allows multiple computational nodes to achieve a shared rendering task in cooperation. Besides, rendering structure based on grid environment is proposed to support distributed rendering in a larger scale [4].

With the performance of PC (Personal Computer) being improved obviously, it is feasible to execute a task of rendering on PC [2]. Therefore, the researches for WAN-based rendering system are made. Since there are plentiful computing resources in WAN, this kind of systems is able to support most rendering tasks without the need for those high performance resources. However, there still exist some difficulties and problems:

1. Computational nodes are widely distributed, and their on-line states are unsteady, thus it can't ensure that the rendering won't be interrupted in a long time like the render farm.
2. It has low bandwidth, and the transmission time for rendering material accounts for a large proportion of total processing time.

In view of the second problem, a lossless 3D compression algorithm is devised [3], and the technology of splitting the rendering scene to make it easy to distribute is also a good solution [4]. In this paper, we propose a hybrid P2P-based rendering system which organizes personal computers to construct rendering clusters on P2P networks.

System Structure

The construction is on the basis of hybrid P2P network. It makes full use of computational nodes in WAN to participate in the process of rendering, including preprocessing, distribution, calculation, and management. In this way, it can reduce rendering time and the communication cost and data transmission between nodes and the center servers, avoid the management center becoming the bottleneck, and enhance the scalability of system capacity.

This section emphasizes on components and functions of the system. The core of the system is the control center and the rendering clusters constructed on P2P networks.

Fig.1 shows the structure of system. The storage, transmission, and rendering calculation of the tasks are executed in the P2P networks. Rendering portal is the main entrance for users to access system, in which users can apply for rendering, fill in the rendering information, and upload

Published by Atlantis Press, Paris, France.

source files (including scene, texture and other related files). The system starts to perform scheduling and rendering after submitting the task successfully. During rendering, users are allowed to check the processing state and the progress, and send commands, such as stop, delete, and resubmit, at the same time. The thumbnails of the result can be previewed on the portal and the resultant files are available to be downloaded after rendering is completed.

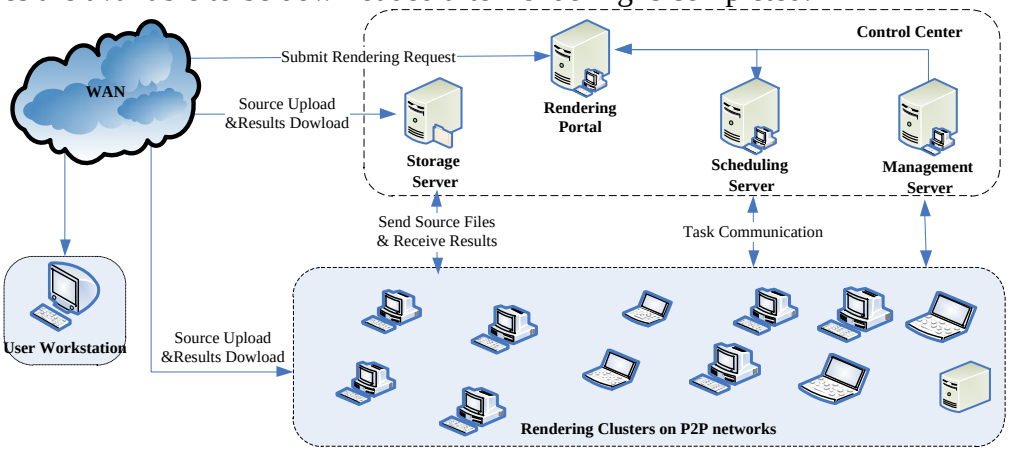

Figure.1 Structure of Rendering System on WAN

By splitting a complete huge task into many small size subtask in the preprocess stage, we can get the estimation for computational complexity of every subtask. The rendering scheduling server gets the real-time performance data of current available network resources from management server, then divides the task into some sub-tasks which are assigned to different rendering nodes. Since P2P network is a kind of mass distributed storage system, huge scene, and files like maps don't need to be transmitted by control center. Therefore, the storage server is more like a temporary storage area which only stores some temporary data. For instance, if a rendering task submitting while there is no proper computing nodes, the system will send the task to the storage server.

Fig.2 presents the network structure of the hybrid P2P-based rendering system. The essence is to build an overlay network on WAN, and divide it into multiple rendering clusters. The peers within close geographical proximity will be added to the same cluster, and each cluster will elect a super-peer (SP) and a backup super-peer (BSP). SP and BSP take charge of the peer management and the requests for rendering in their own cluster cooperatively.

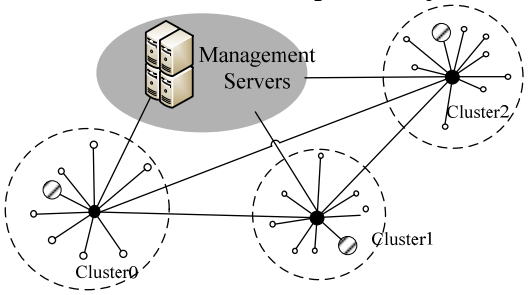

Figure.2 Hybrid P2P-based rendering clusters. Black nodes represent super-peers, white nodes represent ordinary peers, and striped nodes represent backup super-peers

There are four entities in this P2P network:

MS (Management Server): A server in charge of monitoring and management in the P2P networks, and maintaining information of clusters and SPs.

SP (Super Peer): A node in a peer-to-peer network that operates both as a server to a set of clients, and as an equal in a network of super-peers [5]. SP is the peer with the best reputation. The duties of a SP are to maintain peers information within its own cluster, communicate with the control center, and exchange task info.

BSP (Backup Super Peer): A SP becomes a single point of failure for its cluster, and a potential bottleneck. When the super-peer fails or simply leaves, all its clients become temporarily disconnected until they can find a new super-peer to connect to [5]. To provide reliability, each cluster elects a BSP automatically. As soon as SP fails or leaves, BSP takes over the management of

the cluster instead of SP, and a new BSP will be elected at the same time.

OP (Ordinary Peer): An OP is installed with a render that is capable of executing rendering. Each OP saves the information of MS and the cluster it joined to. When the cluster re-elects, it is possible that an OP becomes a BSP.

Construction Method of Rendering Cluster

To reduce the communication costs between SP and OP, we organize the peers within close geographical proximity into a cluster. The migration of tasks is implemented in cooperation by those SPs without republishing the tasks from MS.

Cluster Creation. We assume the number of clusters in the system is K, and limit the maximum cluster number to K_{max}. It is assumed that number of peers in the cluster is N, and the maximum node number is N_{max}. So there will be K_{max} SPs to communicate with MS at most, thus the peer capacity in the entire system will be $\sum_{i=1}^{K_{max}} N_i$, and the maximum peer capacity will be $K_{max} \times N_{max}$.

The main steps of cluster creation are as follows:

(1) On the initialization, there is no cluster exist, P_0 which just registered to the rendering system will become SP of the first cluster. Then, the first cluster is built up.

(2) When a peer is going to join the rendering system, the system searches information of all clusters, including all the SP address, and calculates the distance from this peer to each cluster:

$$cluster = \min(d\left|P_n - SP_i\right|) \quad i = 0,1,2\ldots k-1 \qquad d\left|P_n - SP_i\right| = ARTT_i < \alpha \tag{1}$$

P_n is a per-join peer, SP_i is the SP in the cluster of number i, and $d|P_n\text{-}SP_i|$ is the distance between P_n and cluster (that is Average RTT), α is a specified response threshold.

(3)If $d|P_n\text{-}SP_j| > \alpha$ (i=0,1,2…j j<k-1)， then P_n applies to establish a new cluster $Cluster|_{j+1}$, and becomes the SP of that cluster.

(4)When the number of clusters reaches k (j=k-1), the addition of clusters is not allowed in the system. Peers can continue to be added if appropriate readjustment is made for α to increase the value of N dynamically ($N<N_{max}$).

With increment of the peers, it should increase k. When cluster is changed, it will start the process of electing SP and BSP. To reduce the fluctuation, it is feasible to adjust the cluster scale by cluster merging and cluster splitting.

Rendering Peer Joining and Leaving. Firstly, P_n sends a request of joining to MS, from which it acquires a clusters list. After figuring out its $Cluster|_{nearest}$ by using Eq.1, P_n can communicate with the SP in $Cluster|_{nearest}$, and apply to join in. After the success of joining, P_n will keep alive to SP. A newly joined peer will never replace the current SP with the purpose of reducing the frequent fluctuation in the network. However, BSP will be reconsidered. If P_n has higher reputation, it will replace current BSP, and register to be new BSP. To reduce network influence, peers are allowed to perform a few operations before normal exit. When SP exits, BSP would update to be SP and take in charge of the cluster. When BSP exits, there would start electing a new BSP in the cluster. But we must consider the unusual situations such as power off, broken network, and dead halt.

If SP exits abnormally, as soon as the BSP detects the abnormal exit of SP, it applies to be SP at once. If the original SP could not be detected by MS, BSP will upgrade to SP and send notification to the whole cluster. The abnormal exit of SP is transparent to OPs, so the processing procedure of OPs is the same as the procedure of SP's normal exit. If BSP exits abnormally, election algorithm will run again to elect a new BSP. The peers in the same cluster are within geographical proximity, so SP and BSP are possible to be in the same physic network. If both SP and BSP are invalidated simultaneously, the OPs should be re-initialized and choose one cluster to join in.

Figure.3 State Transition Graph between Peers

We can present the relation between OP, BSP and SP in Fig.3. Becoming BSP is a precondition

of being SP as for OP except the first peer of the cluster. As the peer increases, BSP is possible to be downgraded to OP. In this way, SP is hard to be replaced, and the stability of SP is guaranteed by reducing a lot of fluctuation brought by the election in the cluster. And the comprehensive index of BSP like performance, reputation can maintain best or second best.

Election of SP and BSP. Fig.3 shows that the OP has no ability to become a SP directly, only BSP has the right to upgrade to be SP. In fact, BSP Election Algorithm chooses the optimum OP to be the BSP, and upgrades the original BSP to perform duties of SP.

BSP Election Algorithm is the key part of the entire P2P protocols we proposed. A node will not be super node, unless it is accessible, credible and highly stable [6]. In this paper, we use the Reputation Evaluation Algorithm [7] to evaluate each peer. In this way, a BSP is elected in the cluster (BSP will replace SP as long as the SP is inactive, so there is no need of election for SP).

Because SP and BSP both have the most complete information about peers in the same cluster, election algorithm is calculated by them. What OP should do is to report its own information to SP and BSP. We adopt a kind of marking standard in which each peer has a reputation [7]. The reputation presents the service capability and reliability of the peer. Service capability is obtained from peer's CA (Computing Ability) and NA (Network Ability); Reliability mainly refers to the peer's AOT (Average Online Time).

We set the proportion parameters of CA, NA and AOT at μ, ν and λ, and $\mu+\nu+\lambda=1$, $\text{Score}|_{\text{CA}}$ is the evaluated score of CA, $\text{Score}|_{\text{NA}}$ is for NA, while $\text{Score}|_{\text{AOT}}$ is for AOT. Hence, the reputation of a peer is as below:

$$Reputation|_{peer}=\mu\times\text{Score}|_{\text{CA}}+\nu\times\text{Score}|_{NA}+\lambda\times\text{Score}|_{\text{AOT}} \quad (2)$$

The following formula gives the evaluation of CPU and RAM as outcome:

$$Evaluation_{CPU}=W_{CPU}\times(1-CPU_{load})\times\frac{CPU_{speed}}{CPU_{\min}}$$

$$Evaluation_{RAM}=W_{RAM}\times(1-RAM_{usage})\times\frac{RAM_{size}}{RAM_{\min}}$$

$$Score|_{CA}=\frac{Evaluation_{CPU}+Evaluation_{RAM}}{W_{CPU}+W_{RAM}} \quad (3)$$

In Eq.3, W_{CPU} is the weight of CPU, CPU_{load} is the load of current CPU, CPU_{speed} is actual speed of CPU, and $CPU_{\min}$ is the required minimum speed of CPU. W_{RAM} is the weight assigned for RAM, RAM_{usage} is the usage rate of current RAM, RAM_{size} is the initial capacity of RAM, and $RAM_{\min}$ is the required minimum capacity of RAM.

CPU calculation velocity is decided by many factors, such as frequency, L2 & L3 cache, and the core architecture, the amount of cores. We can set a default value for CPU_{speed} according to the evaluation score of the mainstream CPU. We could also evaluate CPU_{speed} by the way of comparing the time different CPU uses to rendering the same scene file.

When we transmit rendering scene and result on WAN, the uplink bandwidth and downlink bandwidth from peer to storage server should be taken into consideration of evaluation.

$$\text{Score}|_{\text{NA}}=\beta\times BW_{uplink}+(1-\beta)\times BW_{downlink} \quad (4)$$

BW_{uplink} is the uplink bandwidth from peer to storage server while $BW_{downlink}$ is the bandwidth from storage server to peer, and $0<\beta<1$. We take BW_{uplink} for example to illustrate calculation method. Due to the great fluctuation on the change of bandwidth, it is necessary to consider average bandwidth $BW_{uplink\_average}$ and bandwidth deviation $BW_{uplink\_deviation}$, it is calculated as below:

$$BW_{uplink}=BW_{uplink\_average}+2\times BW_{uplink\_deviation}$$

We use BW_{uplink_sample} as the latest uplink bandwidth from each sample, and the formulas for calculation of average bandwidth and bandwidth deviation are as follows:

$$BW_{uplink\_average} = (1-g) \times BW_{uplink\_average} + g \times BW_{uplink\_sample}$$

$$BW_{uplink\_deviation} = (1-h) \times BW_{uplink\_deviation} + h \times \left| BW_{uplink\_sample} - BW_{uplink\_average} \right|$$

Set $g = 0.125, h = 0.25$ as the initial value, we can make an adjustment to the values at runtime according to the practical situation.

The way for calculation of $BW_{downlink}$ is similar to the method above.

$\text{Score}|_{\text{AOT}}$ is the ratio of on-line time length and up-line times which recorded by the software automatically. The formula for calculation is as follows:

$$\text{Score}|_{\text{AOT}} = \frac{OnlineTime}{UpTime} \tag{5}$$

$\text{Score}|_{\text{CA}}$, $\text{Score}|_{\text{NA}}$ and $\text{Score}|_{\text{AOT}}$ are changing dynamically. Hence, each peer has to calculate its reputation at intervals of t, and attaches the reputation to the heartbeat message, and then sends the message to SP. Since BSP has complete information of pees in the cluster as well, BSP will upgrade to SP as soon as SP leaves, and elect a new BSP according to election algorithm later.

Conclusion

It is a very complicated process of constructing a P2P network to meet the need of rendering on WAN. The composition and working mechanism of rendering cluster are introduced in this paper, and the main work of research is the study on the quantitative evaluation of the service capability and reliability of the peer, by which SP in cluster is elected. We also discuss the merging and splitting problem of the system in dynamic process to ensure the best utilizes of resources. Further research will go on for the evaluation of rendering computational node's capability and reputation and the optimization for rendering scheduling strategy. Furthermore, security schemes will be studied to prevent attack on SP and SP cheating.

Acknowledgement

This paper is supported by "Research and Demonstration of Broadcast Control Platform Based o n Three Networks Convergence Integration" (2011AA01A107)

Reference

[1] Y.B.Wang: *A Survey of Render Farm under Network Environment*, Microelectronics & Computer. Vol.25 No.9. (2008)

[2] B.Wylie, C.Pavlakos, V.Lewis and K.Moreland: *Scalable Rendering on PC Clusters*, IEEE Computer Graphics and Applications. (2001).

[3] A.Chong, A.Sourin and K.Levinski: *Grid-based computer animation rendering*. Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia, (2006), p. 39-47.

[4] Y.B.Wang, Z.G.Hong and Y.Z.Cao: *A Method of WAN-oriented animation rendering task decomposition and system implementation*. China, H04L29/08; G06T13/20. (2011).

[5] B.Yang, H.Garcia-Molina: *Designing a Super-Peer Network*, 19th International Conference on Data Engineering, (2003), p. 49-60.

[6] V.Lo, D.Y.Zhou, Y.H.Liu, C.GauthierDickey and J.Li: *Scalable Supernode Selection in Peer-to-Peer Overlay Networks*, Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems, (2005) , p.18-27.

[7] Y.M.LIU: *The Research of Reputation-Aware SuperNode Selection Algorithm in P2P System*. Journal of the Graduate School of the Chinese Academy of Sciences. Vol25 No.2 (2008).