Backburner Management For Network Rendering 3DS MAX

Assist. Prof. Ahmed Y. Mahmoud
Faculty of Engineering & Information Technology,
Information Technology Department,
Al-Azhar University, Gaza, Palestine,
ahmed@alazhar.edu.ps

Suheir H. ALmurshidi
Faculty of Engineering & Information Technology,
Information Technology Department,
Al-Azhar University, Gaza, Palestine,
soher.hassan88@gmail.com

Abstract— Nowadays, demands on animations rendering and modeling are increasing for various industries and merchandise. Autodesk 3DS max may be a skilled 3D computer graphics program for creating 3D animations, models, games and pictures. Network rendering may be a means that of mass-processing multiple rendering tasks or jobs. In this manuscript we use Autodesk Backburner with 3DS to achieve rendering tasks through network. Experiments show that Backburner efficiently coordinating job assignments.

> *Keywords—3DS MAX; Autodesk Backburner; Network rendering*

I. INTRODUCTION

The recent advance of production animation and high consumption on it leads to find new resources for rendering and modeling these enormous images and frames [1]. Accordingly, the main concern was to investigate algorithm, tools to achieve the greatest benefit in communication via network rendering. Increasing productivity and working with high-resolution face significant challenges. A novel approach is the one based on 3DS MAX [2].

In recent years, 3DS MAX have been considered by many authors (see for example, [3] - [6] and the bibliography therein).

Recently, 3DS MAX becomes one of the most important tools in network rendering. 3DS Max is one of the important tools that used to produce animation and modeling with high quality. Moreover, 3DS Max provides several benefits like XRef Renovations, Camera Sequencer, Easier Revit and SketchUp Workflows etc. The importance of 3DS Max comes from the revolution in the communication networks , the proliferation of applications which use internet, provided us the opportunity to use Backburner Management for Network Rendering due to the urgent need of increasing productivity without decreasing the resolution for images, video and modern media.

In this manuscript , we propose a model for Backburner Management for Network Rendering; the manuscript is organized as follows. In Section II, we recall some basic concepts, notations. In Section III, we give a description of the network rendering and its configuration. In Section IV, the security of network rendering is discussed. In Section V, we conclude with some remarks.

II. OVERVIEW OF RENDERINGEase of Use

In this section, we recall some standard concepts and introduce some definitions from network, which will be used throughout this manuscript. Most of them can be found in [5] and [6].

A. Network Rendering

Network rendering allows the user to take advantage of his/her network's computer resources to be used for rendering images, animations, and video frames. The tasks are organized into a queue that all users can see. Additionally, tasks can be sent from the queue to network rendering. It can be executed by several software; one if those software named Backburner. Backburner is a free software application packaged and installed with 3DS MAX which is responsible for coordinating how job assignments are processed [7]. Also it can queue up tasks from any number of cameras that assigned to process and render. In order to pass batch-rendering; the tasks are put in queue of rendering tasks to the Network Job Assignment dialog, then backburner using Batch Render tool.

B. Network played

Network played is an essential role for connecting PCs and usage their resources, one of these benefits network rendering; that means using network for processing multiple rendering tasks and jobs, it uses multiple computers that connected over a network to perform rendering task and rendering whatever is set up in scene.

III. CONFIGURATION OF NETWORK RENDERING

In this section a description of the network rendering configuration for manager and servers is given [8], [9], [10]. It includes the steps of network configuration. Figure 1 shows Backburner Manager General Properties; the configuration initialize the properties and set as default for the first time. Step 1 explain the settings for starting rendering across a

network; it assigns one machine to work as Manager, and then it specifies any number of machines to work as Servers. The configuration procedure is explained through several steps (1-11) and Fig. 1-4.

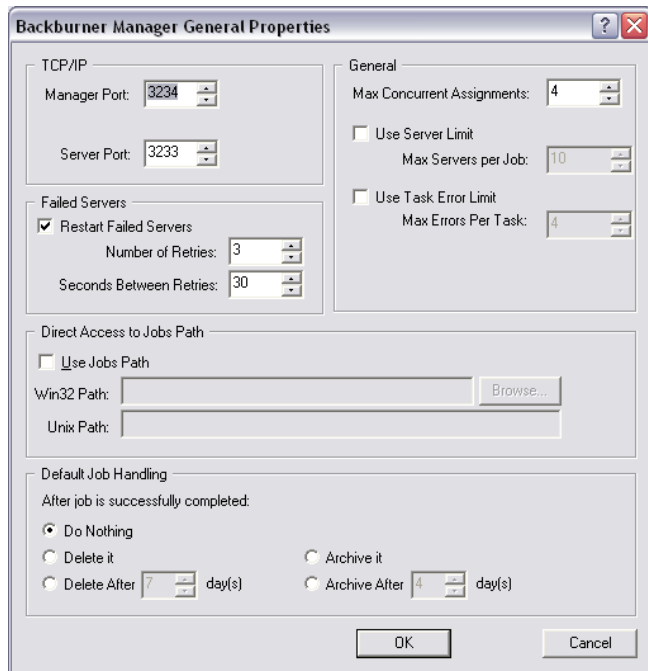**Step 1.** Go to Start menu ➤ Programs ➤ Autodesk ➤ Backburner and choose the ⬡ Manager menu item.

Fig. 1. *Backburner Manager General Properties dialog.*

Step 2. Click OK to accept the default settings, the Backburner Manager dialog displays.

Step 3. On the same computer, go to Start menu ➤ Programs ➤ Autodesk ➤ Backburner and choose the ⬡ Server menu item.

Figure 2 shows the Backburner Server General Properties dialog box with default settings, when you run Server for the first time.
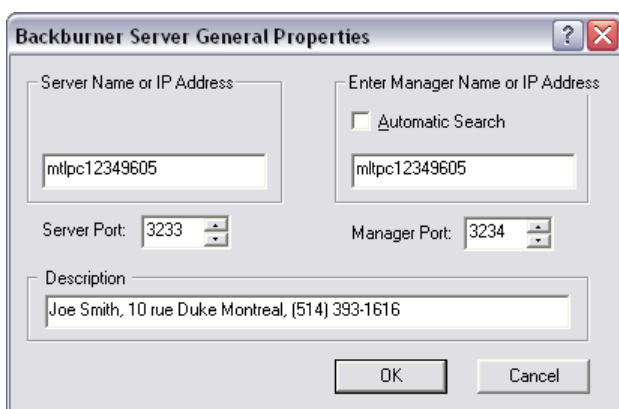

Fig. 2. *Backburner Server General Properties dialog.*

Step 4. Click OK to accept the default settings. The Backburner Server dialog displays. After a few moments, messages appear in both the Server and Manager windows indicating that the Server has successfully registered with the Manager (Figure 2).

Step 5. Repeat steps 3 to 4 on all the Server systems you intend to make available for your rendering jobs.

Step 5. Return to the Manager system, start 3ds Max, and load the first scene you want to render.

Step 6. From the Rendering menu, choose Render Setup, or on the main toolbar, click (Render Setup).

Step 7. Set the rendering parameters and specify an output path and file name.

Step 8. From the Render Setup dialog ➤ Target drop-down list, choose Submit To Network Rendering (Figure 3)

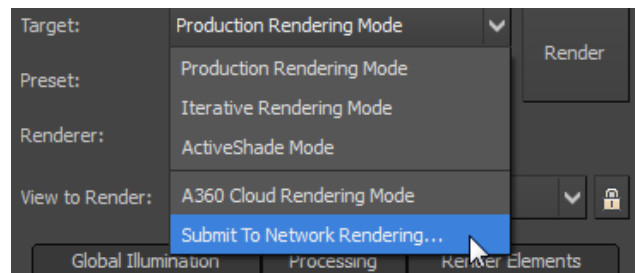Figure 3 shows The Network Job Assignment dialog appears.

Fig. 3. *The Network Job Assignment dialog appears.*

Figure 4 shows Network Job Assignment dialog showing all the server station plus the manager station that is running Server.
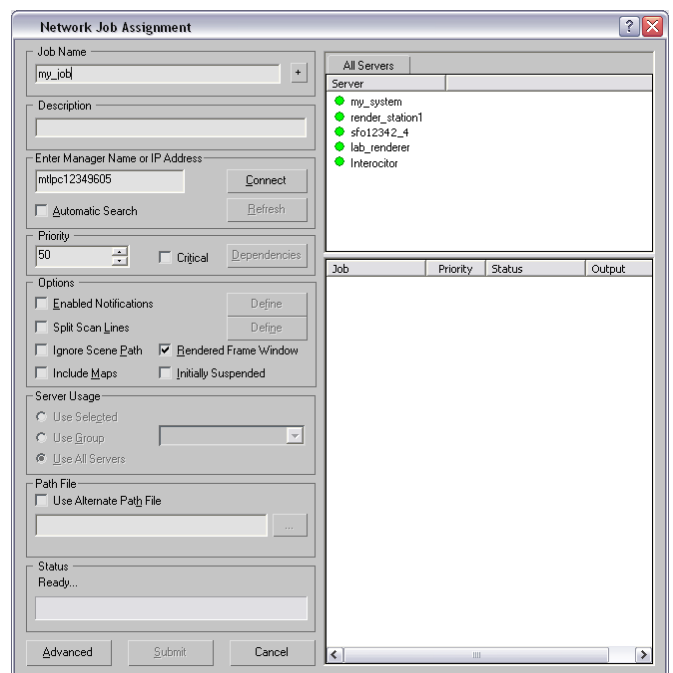

Fig. 4. *Network Job Assignment dialog.*

Step 9. Enter a job name (it's a good idea to change the default name) and then click the Connect button. The names of all Servers appear in the Server window to the right. Those with green dots next to their names are ready to start rendering as shown in Figure 4.

Step 10. Click the Submit button. The Manager submits the job to the Servers, which begin rendering.

The user assign the files in Backburner and later on he will collect the result. Then the user can start to review and use, etc.

IV. SECURITY OF BACKBURNER

Nowadays, security is considered as the most important topic in different fields. The proliferation of digital data show the urgent need of security [11]. Security The security of storage, transmission of digital data and data authentication can be obtained by applying encryption algorithms (see for example [12] – [20]). Applying the software security requirements is an important issue will encourage the users to use the software and increase their confidentiality [21].

Backburner is using local network for rendering and this gives the administrator the ability to distribute the jobs between computers on the local network and here is the problem arise, when this administrator computer connected to an unsecure channels (un-trusted) via internet and network then the transmitted became vulnerable to attackers and can be hacked it, they will have the ability to use backburner to create a denial of service.

Backburner manager contains a stack-based buffer overflow vulnerability and the cmd-job utility in backburner allows remote attacker to execute arbitrary commands or codes [7]; the length of input command is not checked or restricted and this may have an overwrite for stack instructions. Figure 5 shows stack buffer overflow mechanism. By default Backburner manager process listens on TCP/UDP port 3234, and this could configure by the user to use another port.

Reducing the danger of using Backburner from attacker can be obtained by adding restrictions to access Backburner manager.exe on port which used, and identified allowed users and trusted network by a system administrations and firewalls.

Based on [8], we suggest a plan as shown in figure 6; the designed plan shows the share folder Backburner on a dedicated server; It focuses on setting up a working Backburner render farm from scratch. This plan is designed for users whom requesting to gain an edge on getting the most of your existing rendering resources or user requesting to begin using the included features of Autodesk Backburner for network rendering in 3ds Max and/or Combustion.

V. CONCLUSION

In this manuscript, we show the network rendering configuration for manager and servers. The increasing demand on 3DS MAX with high resolution and fast delivery is considered; We implement Backburner to achieve network rendering and managing it by making a queue for jobs and assignments which effectively reduce the cost of time and resources for rendering. The security of Backburner also is considered and highlighted.
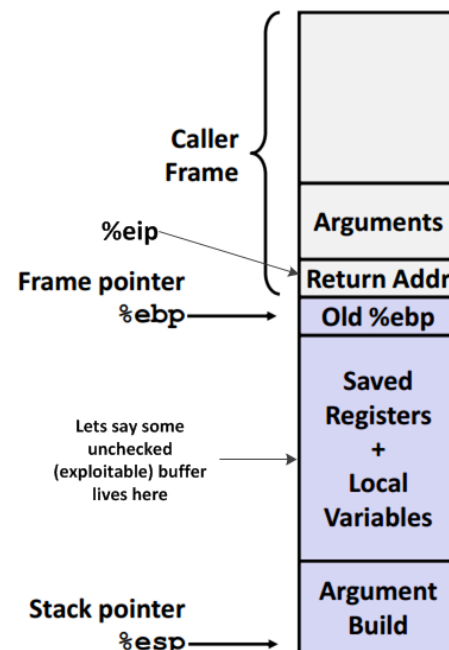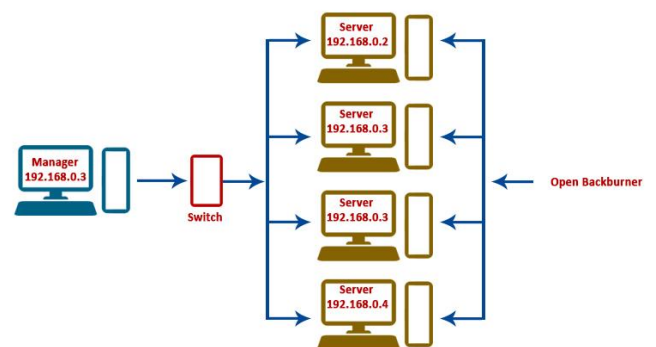

Fig. 5. *Stack based buffer overflow*

Fig. 6. *Share folder Backburner on a dedicated server*

REFERENCES

[1] Kayvon Fatahalian (PhD. theses) Evolving The Real Time Graphics Pipeline for Micropolygon Rendering, [access date 31/11/2016] http://graphics.stanford.edu/~kayvonf/papers/kayvonf_dissertation.pdf

[2] http://www.autodesk.com/products/3ds-max/ [access date 31/11/2016]

[3] Engel, K., Hastreiter, P., Tomandl, B. and et.al (2000). Combining local and remote visualization techniques for interactive volume rendering in medical

applications. In Proc. of IEEE Visualization 2000, 449–452

[4] Yasuhide Okamoto , Takeshi Oishi Katsushi Ikeuchi, (2011) Image-Based Network Rendering of Large Meshes for Cloud Computing, International Journal of Computer Vision ·

[5] Madeleine Taylor, Anne Whatley, and Julia Co"man, Network Evaluation in Practice: Approaches and Applications, , International Journal of The Foundation review, vol 7(2), 2015, p.p. 22-37, http://www.networkimpact.org/wp-content/uploads/2015/07/TFRVol7Is2_NetworkEvaluationinPractice.pdf [access date 31/12/2016]

[6] https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/3DSMax/files/GUID-C3CD6E31-72F5-4646-A400-971E2DA1CFED-htm.html [access date 23/10/2016]

[7] http://security.stackexchange.com/questions/17075/buffer-heap-overflow-register-of-what-is-executed [access date 17/09/2016]

[8] http://help.autodesk.com/view/3DSMAX/2016/ENU/?guid=GUID-50FCDD2A-2D68-427C-B75E-65A0717FD2FC [access date 17/09/2016]

[9] http://security.stackexchange.com/questions/17075/buffer-heap-overflow-register-of-what-is-executed [access date 17/09/2016]

[10] https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-2344 [access date 17/09/2016]

[11] Mahmoud, Ahmed Y. and Barakat, Mohammed S. and Ajjour, Mahmoud J , Design And Development Of Elearning University System, Journal of Multidisciplinary Engineering Science Studies (JMESS), vol. 2, 5, 2016, pp. 498-504

[12] Mahmoud, A. Y., & Chefranov, A. G. (2009, October). Hill cipher modification based on eigenvalues hcm-EE. In Proceedings of the 2nd international conference on Security of information and networks (pp. 164-167). ACM.

[13] Chefranov, A. G., & Mahmoud, A. Y. (2010, September). Elgamal public key cryptosystem and signature scheme in GU (m, p, n). In Proceedings of

the 3rd international conference on Security of information and networks (pp. 164-167). ACM.

[14] Mahmoud, A. Y., & Chefranov, A. G. (2010, May). Secure Hill cipher modifications and key exchange protocol. In Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on (Vol. 2, pp. 1-6). IEEE.

[15] Mahmoud, A. Y., & Chefranov, A. G. (2014). A Hill Cipher Modification Based on Eigenvalues Extension with Dynamic Key Size HCM-EXDKS. International Journal of Computer Network and Information Security, 6(5), 57.

[16] Doukhnitch, E., Chefranov, A. G., & Mahmoud, A. (2013). Encryption Schemes with Hyper-Complex Number Systems and Their Hardware-Oriented Implementation. Theory and Practice of Cryptography Solutions for Secure Information Systems, 110.

[17] Mahmoud, A., & Chefranov, A. (2014). Hill cipher modification based on pseudo-random eigenvalues. Appl. Math, 8(2), 505-516.

[18] Mahmoud, A. Y., & Chefranov, A. G. (2012). Secure hill cipher modification based on generalized permutation matrix SHC-GPM. Information Sciences Letters, 91-102.

[19] Mahmoud, A. P. A. Y., & Mahdi, A. O. (2016). Comments on Multi-Window Against Mobile Application Lock. Journal of Multidisciplinary Engineering Science Studies (JMESS), 2, 494-497

[20] Mahmoud, A. Y. A. (2012). Development of Matrix Cipher Modifications and Key Exchange Protocol.

[21] Abdelwahed, A. S., Mahmoud, A. Y., & Bdair, R. A. (2017). Information Security Policies and their Relationship with the Effectiveness of the Management Information Systems of Major Palestinian Universities in the Gaza Strip. International Journal of Information Science and Management, 15(1), 1-26.