APPEX
NETWORKS

LotServer™ Deployment Manual
Maximizing Network Performance, Responsiveness and Availability

1. Introduction

LotServer is a ZetaTCP™ powered software product that can be installed on origin web/application servers or cache servers to accelerate content delivery to end users, requiring no software or plugins on client side.. It supports all mainstream Linux and Windows server distributions. LotServer is even more effective for delivering content to users that are widely distributed across the Internet. Users that are accessing content across long network distance, multiple carrier networks or from wireless devices can see dramatic improvement in their user experience which increases the usability of the service. Also improved is the stability of the network connections, lowering network access failures by a large margin.
LotServer is used in some of the largest networks in the world to increase stability and enhance the user experience.

2. How Does It Work

LotServer is a software module which implements the latest generation of Zeta-TCP Learning Based algorithms, running in the kernel of the Operating System. Zeta-TCP improves upon traditional TCP in the following ways:

2.1 Intelligent Congestion Window and Recovery algorithms

Traditional TCP's congestion avoidance algorithms, such as the most commonly used NewReno and its later variations, are largely based upon the overly simplified assumption that all loss is an indication of network congestion. This assumption is misplaced in today's networks, especially for wireless networks where loss and varying latency are very commonplace. Other more modern TCP algorithms like TCP Vegas, base their congestion predication solely on end-to-end latency, which is over simplified and often wrong. Sometimes this oversimplified approach can cause over aggressive re-transmits which actually makes thw situation worse and fails to effectively back off when the congestion happens for real.
Zeta-TCP introduces a number of intelligent algorithms to measure the level or likelihood of the network congestion based on both latency and the loss rate. This is a more accurate way to determine whether the congestion is really happening and apply different recovery algorithms under different situations to maximize the end-to-end throughput.

2.2 Better and more accurate loss-detection algorithm

Zeta-TCP has its own unique, simple but comprehensive packet-loss detection algorithm.. Traditional TCP frequently makes mistakes in detecting packet loss. In complex packet-loss scenarios, traditional TCP tends to either identify reordered packets as lost, triggering a resend and in turn occupying extra bandwidth, or waits a relatively long time for more feedback from the peer to confirm a lost packet, leaving bandwidth idle. Zeta-TCP's loss detection overcomes these issues by tracking the loss probability for each packet, ensuring the shortest response time and best bandwidth utilization.

In addition to the major enhancements above, Zeta-TCP has a number of other improvements over traditional TCP. Tested and tuned by years of large-scale field deployment, the asymmetric TCP acceleration of Zeta-TCP is now a proven technology.

Naming Conventions

LotServer executable file: acce-{engineVersion}-{Linux distroVersion_kernelVersion}
LotServer license file: apx-{expirationTime}.lic

3. Installation

You can choose from two installation methods offered by LotServer web distribution: automatic or manual. Regardless of which method you use, you will need root privileges of your servers to successfully install LotServer.

3.1 Automatic Installation

We recommend automatic installation as it is the easiest method to setup LotServer. When choosing automatic installation, the installer will connect back to LotServer web distribution over the internet to authenticate your account, license your installation and download the LotServer executable package bound with the license file. Therefore to use automatic installation, your server must have internet access. If not, please choose the manual installation instead.

The detailed instructions for automatic installation are found at: http://download.appexnetworks.com/w.do?m=lsl

The installation script will display messages about the final result of the process upon exit. Usually there are three different outcomes:

1) Success: after a successful installation, the installer should have created the directory "/appex" under the file system root. All the LotServer files and directories are created beneath this directory. There are 3 sub-directories under "/appex": "bin", "etc" and "log". Directory "bin" stores the executable files and scripts. Directory "etc" stores the LotServer license and configuration files. Directory "log" should be empty.

2) The automatic installer failed to locate the LotServer software compatible with the server platform. The installer will exit after showing the error message.

3) The account has run out of licensed number of copies to install. The installer will exit after showing the error message.

3.2 Manual Installation

If your servers don't have internet access for some reason, you can choose manual installation to download LotServer to your PC that has access to the download site, then copy the file to the server to install and run.

Detailed instructions can be found at: http://download.appexnetworks.com/w.do?m=lslm

4. Running LotServer

4.1 Directories and File

Name	Type	Description
/appex/	Dir	The root of the AppEx LotServer installation. All the LotServer related files and directories are created under it.
/appex/bin/	Dir	Stores LotServer executable binaries and scripts.
/appex/bin/acce-xxx	File	LotServer executable binary.
/appex/bin/lotServer.sh	File	LotServer start script.
/appex/etc/	Dir	Stores LotServer license and configuration files.
/appex/etc/apx-xxx.lic	File	LotServer license file.
/appex/etc/config	File	LotServer configuration file.
/appex/log/	Dir	Stores LotServer runtime logs and captures files for diagnostic purposes.

4.2 Common Parameters

You can modify file /appex/etc/config to set or alter LotServer's parameters. The following are the most commonly used ones and their usages:

1) *accif="eth*"*
The network interface to be accelerated. eth* should be the interface that the server offers network services, e.g., eth0, eth1, ... Command "ifconfig" can be used to obtain the names of the interfaces.

2) *acc="1"*

Whether to enable TCP acceleration. To enable the acceleration, set this parameter to 1. Otherwise 0.

3) *csvmode="1"*
The "Conservative Mode" enabler. Setting it to 1 enables the Conservative Congestion Control Mode, and 0 disables it. When TCP acceleration is enabled, the effective data rate may drop under certain patterns of heavy packet losses. Enabling the Conservative Mode will prevent such degradation from happening. However, it may also result in slightly lower acceleration ratio.

4) *wankbps="1000000"*
The outbound bandwidth of the accelerated network interface. This value should be the maximum bandwidth for the traffic flowing from the server to the internet. The unit here is Kbit/s. e.g., if the maximum bandwidth to the internet is 1Gbit/s, set this parameter to 1000000.

5) *waninkbps="1000000"*
The inbound bandwidth of the accelerated network interface. This value should be the maximum bandwidth for the traffic flowing from the internet to the server. The unit is Kbit/s. e.g., if the maximum bandwidth to the server is 1Gbit/s, set this parameter to 1000000.
Ideally, wankbps and waninkbps should be set to the same values of the real bandwidth allocated to the server. If it is hard to determine the real bandwidth, or the bandwidth is dynamically shared with other servers or devices, then they can be simply set to the throughput that the accelerated NIC offers. e.g., if the NIC is a full-duplex Gigabit Ethernet card, then these parameters can be set to 1000000. If these parameters are set lower than the real bandwidths, then the traffic flows in and out the server will be bound by the values configured, i.e., the throughput will be lowered.

6) *apxexe=" /appex/bin/acce-***"*
The LotServer executable binary's file name.

7) *rsc="0"*
"0": Feature OFF, "1" Feature On. Enable this feature for the NICs that support RSC(Receiver-Side Coalescing). Otherwise the inbound traffic may not be properly processed.

8) *gso="0"*
"0": Feature Off, "1": Feature On. Enable this feature when the GSO/TSO(Generic Segmentation Offload/TCP Segmentation Offload) is enabled on the NICs.

9) *engineNum="0"*
"0": Create the number of engine according to the number of your CPU cores, greater than "0": create the number of engine according to parameter value. If you see a pop up message "Cannot allocate memory", when you launch LotServer, it means your server doesnot have enough memory to create many engines. You can reset the parameter value to smaller than the number of your CPU cores.

10) *shortRttMS="0"*
"0": Feature Off, greater than "0": Bypass the flows with RTT lower than the value. Secifies the threshold of the RTT of the flows to be bypassed. If the RTT is lower than the given threshold, then the flow will be bypassed. Otherwise accelerated.

11) *pmtu=""*
"0": Feature Off, "": Feature On. Null value means enable PMTU function of LotServer.

12) *apxexe="/appex/bin/acce-***"*
The path of LotServer executable binary file.

13) *apxlic="/appex/etc/apx-***.lic"*
The path of LotServer license file.

4.3 Starting / Stopping / Reconfiguring LotServer

Starting LotServer: */appex/bin/lotServer.sh start*
This command loads up the LotServer kernel module and starts the Zeta-TCP engine, applying the configurations in file /appex/etc/config.

Stopping LotServer: */appex/bin/lotServer.sh stop*

This command stops the Zeta-TCP engine and unloads the LotServer kernel module. It's best to make sure no processes are accessing directory /proc/net/appex/, for example, a command console with current directory set to /proc/net/appex/.

Reconfiguring LotServer: */appex/bin/lotServer.sh reload*

This command does not unload the LotServer module. After you have modified file /appex/etc/config, you need to run this command to have the changes take effect.

NOTE: It is very important to ensure that TSO (TCP Segmentation Offload) and GSO (Generic Segmentation Offload) are turned off before running LotServer. These features will prevent LotServer from intercepting network packets and therefore break network connections. You can query the TSO and GSO status with command "ethtool –k eth*". Use "ethtool –K eth* tso off" and "ethtool –K eth* gso off" to turn off these features. When you launch lotServer.sh to start LotServer, it will automatically run ethtool first to turn off TSO and GSO. If ethtool returns failure, the script will stop loading LotServer.

4.4 Verifying If LotServer Is Running

You can run the Linux command "*lsmod*" and check if a module named "appex0" exists to ensure that LotServer has been successfully loaded. Then "*cat /proc/net/appex/stats*" to verify that LotServer is successfully running.

AppEx Networks Corporation
1601 McCarthy Blvd.
Milpitas, CA, 95035

+1 408-973-7898
contact@appexnetworks.com

More information can be found at:
www.appexnetworks.com

For a Free LotServer trial:
download.appexnetworks.com