Synergy SYN304:
Best Practices for NetScaler Performance Tuning in the AOL Network

Presenters:
Kevin Mason - kem@aol.net
Tim Wicinski - tjw@aol.net

May 11, 2012

Audience: Sr Admin | CCNA level with responsibility for Netscaler configs

Agenda

- Goals
- Information on the AOL Production Network
- TCP Tuning Concepts
- TCP Tuning on the NetScaler
- AOL tcpProfiles
- Case Study: bps Improvements
- Case Study: Retransmission Reduction
- Case Study: Engadget Coverage of New iPad Announcement
- Troubleshooting and Monitoring NetScalers
- Appendixes

Today's Statistics

All the AOL statistics presented today are operational numbers. They should not be construed to represent page views or other types of site popularity data.

3

Performance Tuning Goal

Create the best possible experience for AOL Customers by continuously analyzing and engineering the fastest, most efficient network possible.

4

Definitions

Just to make sure we are on the same page

- VIP - Load Balancing (LB) and Content Switching (CS) VServers
- Service - Backend connectivity to Real Hosts
- Outbound - Traffic outbound from the VIP or Service
- Inbound - Traffic inbound towards the VIP or Service

5

AOL Production Network Stats

Load Balancing

70 HA pairs of NetScalers
10G, multi interface attached
19,642 VServers
27,214 Services (Host + Port)
13,462 Servers, majority cloud based

GSLB/DNS

Approx 18K DNS domains
Approx 4K GSLB configs
180M queries/day

As of April 18, 2012

AOL Production Network

Internal Tooling

Web based self service site managing all changes, using a mixture of SOAP and NITRO calls:

- Avg 250 Self Service changes / week.
- Avg 25K config command changes / week.

7

TCP Tuning Concepts

- Have a solid, technical understanding of the client's connection method.
- Get the bits to the customer quickly and cleanly.
- Where possible, send a large initial burst of packets.
- Ramp traffic up fast.
- Recover errors quickly, avoid slow start.

Melt fiber, smoke servers, choke routers

TCP Tuning Concepts

TCP Tuning is done per connection not the aggregate of the link.

Tuning Factors include:

- Latency or Round Trip Time (RTT)
- Minimum Bandwidth
- Packet Loss
- Supported TCP Options

Even though the propagation delay is low, BDP is still a limiting factor, especially with HTTP 1.1 pipelining.
Even a delay of 10ms can be impacting

- Servers can easily push 650 mb/s per flow
- 650mb/s x 10ms = 812,500 Bytes, WS = 5

TCP Connection Options

- TCP negotiates options to define how the connection operates.
- TCP Options are generally symmetrical, both sides have to support it, or the option is dropped.
- TCP Option values are NOT always the same on both sides. For example, Receive Windows, Window Scaling and MSS (Max Segment Size) are often different.

Option Negotiation is only done during the SYN, SYN/ACK, ACK exchange

!!! Caveat Emptor !!!

While we are happy to share AOL experience and configurations, be careful applying these settings in your network, they might cause problems.

When working with any parameters, have a rollback plan!

O'Toole's Comment on Murphy's Law: Murphy is an Optimist.

TCP Tuning on the NetScaler

Two major tuning knobs on the NetScaler:

- tcpParam - Global TCP settings
- tcpProfile - Per VIP or Service settings

We set the tcpParam for general purpose values, but since it is a OS supplied setting, it could be over written by an upgrade.

Using custom tcpProfiles is safer since they will not be overwritten.

12

tcpProfile Values

-WS
-WSVAL
-maxBurst
-initialCwnd
-delayedAck
-oooQSize

-maxPktPerMss
-pktPerRetx
-minRTO
-slowStartIncr
-SACK

-nagle
-ackOnPush
-mss (v9.3)
-bufferSize (v9.3)

If you do nothing else, enable:

-WS (Window Scale)
-SACK (Selective Ack)

13

TCP Window Scale (RFC 1323)

Window Scale is a TCP option to increase the receiver's window size, compensating for long and/or fat networks (LFN). The values used by each device in a connection are often asymmetrical.

- Enabled on the NetScaler with the -WS option.
- The value advertised by the NetScaler is set with -WScale.
- At the least, enable -WS to take advantage of the client and server advertised windows.

RFC1323 – may 1992 Windows 2000, enabled by default Vista
Linux 2.6.8 8/2004 enabled
os X – always enabled

TCP SACK (RFC 2018)

Selective Acknowledgement or SACK is a TCP option enabling a receiver to tell the sender the range of non-contiguous packets received.

- Without SACK, the receiver can only tell the sender about packets sequentially received.
  - This slows down recovery.
  - Can force Congestion control to kick in.
- Enabled with the -SACK option.

rfc 2018 – 8/1996
win98 – default
linux – can't find but imagine always

Citrix default tcpProfile

Settings appropriate for circa 1999.

Have stayed this way due to Citrix general policy of not changing default settings. Some significant changes were made to default on v9.2, see CTX130962 (10/7/2011).

- No Window Scaling or SACK.
  - Causes choppy data flow.
  - Slow to recover from packet loss.
  - Can trigger slow start congestion control.
- Very slow Congestion Window ramp up.
- Small -initialCWind & -maxBurst
  - See "An Argument for Increasing TCP's Initial Congestion Window" from Google for a deeper look.

Notes about CTX130962
-MSS added
-InitialCwnd range increased from 6 to 44
-MaxBurst from min from 2 to 1, max from 10 to 255
-oooQsize max from 512 to 65535
-maxPktPerMss max from 1460 to 512
-pktPerRetx from 100 to 512
-minRTO default fro 1,000 to 100
-bufferSize max to 4194304

Additional Citrix supplied tcpProfiles

Citrix supplies several predefined profiles

Depending on code level, Citrix supplies at least 7 additional tcpProfiles for you to try. This can be a good starting point, but since they are supplied with the O/S, they are subject to changes in the future.

AOL Custom tcpProfiles

18

AOL Custom tcpProfiles

VIP Profiles

- aol_vip_std_tcpprofile
- aol_vip_mobile-client_tcpprofile
- aol_vip_server-2-vserver_tcpprofile
- aol_vip_dialup-client_tcpprofile

Service Profiles

- aol_svc_std_tcpprofile

Complete configs included in Appendix

19

Standard Client to VIP - aol_vip_std_tcpprofile

Push data as fast as possible after the connection is established by increasing the packet burst and ramp rate, improve error recovery.

Assumptions

- Content is generally outbound out to client.
- The max bps per flow is 10 mb/s.
- RTT is 75 ms

Changes from default:

- Enable -WS (Window Scaling)
- Enable -SACK
- Increase -MaxBurst
- Increase -InitialCwnd
- Increase -slowStartIncr
- Increased -pktPerRetx
- Reduced -delayedAck

20

Assumptions

- The maximum bps per flow is 10 mb/s (1.25 MB/s)
- The mean propagation delay (RTT) is 75 ms (0.075 sec)

Mobile client to VIP - aol_vip_mobile-client_tcpprofile

Based on aol_vip_std_tcpprofile & aol_vip_dialup-client_tcpprofile with changes to improve the the mobile (3G, 4G) client experience by increasing tolerance for RTT shifts as the device moves. This reduces retransmissions that can flood a device.

Assumptions:

- RTT can shift from 300ms to 900ms
- bps is limited to < 3mb/s

Changes from aol_vip_std_tcpprofile:

- Increase the -minRTO
- Reduced -delayedAck
- Reduced –maxburst
- Reduced -slowstartincr

Results:

- Avg 30% reduction in retransmissions to mobile clients

21

Server to VIP - aol_vip_server-2-vserver_tcpprofile

Based on aol_vip_std_tcpprofile for vips that are handling internal server flows where the data is inbound to the vserver.

Assumptions

- The source host is in a data center or other attached space.
- The maximum bps per flow is 600 mb/s inbound to a VIP or vserver.
- The max propagation delay (RTT) is 20 ms (0.02 sec).

Changes from aol_vip_std_tcpprofile:

- Increase WScale
- Reduced -minRTO

22

Dial Client to VIP - aol_vip_dialup-client_tcpprofile

Based on aol_vip_std_tcpprofile with changes to improve the dial client experience by reducing retransmission and prevent terminal server flooding.

Assumptions

- Max bps per flow is 50 kb/s (KB/s).
- Avg RTT is 500 ms (0.5 sec).

Changes from aol_vip_std_tcpprofile:

- Increase -minRTO
- Reduced -delayedack
- Reduced –maxburst
- Reduced -slowstartincr

Results:

- Total page render time reduced from ~35s to ~16s.
- SSL handshake times reduced from 4+s to ~ 1s.

We are testing enabling Nagle and reducing mss with this profile

Server to Service - aol_svc_std_tcpprofile

Used on Services for traffic between the real host & NetScaler. Even though the propagation delay is low, BDP is still a factor due to per connection bps.

Assumptions

- Max bps per flow is 650 mb/s (81.25 MB/s).
- Avg RTT is 10 ms (0.010 Sec).
- Majority of flows are HTTP 1.1 or long lived TCP connections.

Changes from default:

- Enabled -SACK
- Enabled -WS
- Increased -WScale
- Reduced -delayedACK
- Reduced -minRTO

24

Results:

- Improved server efficiency

Case Study:
Bits per Second (bps) Improvements

25

Case Study: bps Improvements

Problem:

Need to improve the customer experience and increase efficiency.

- Analysis had shown more than 99% of incoming SYNs support Window Scaling and/or SACK, yet the NetScaler was not utilizing these options.
- Analysis also showed a large number of sessions advertising Zero Windows.
- Sessions were being dropped into slow start due to packet loss.

Blogsmith is the AOL Content delivery platform for sites such as engadget.com.

.

Case Study: bps Improvements

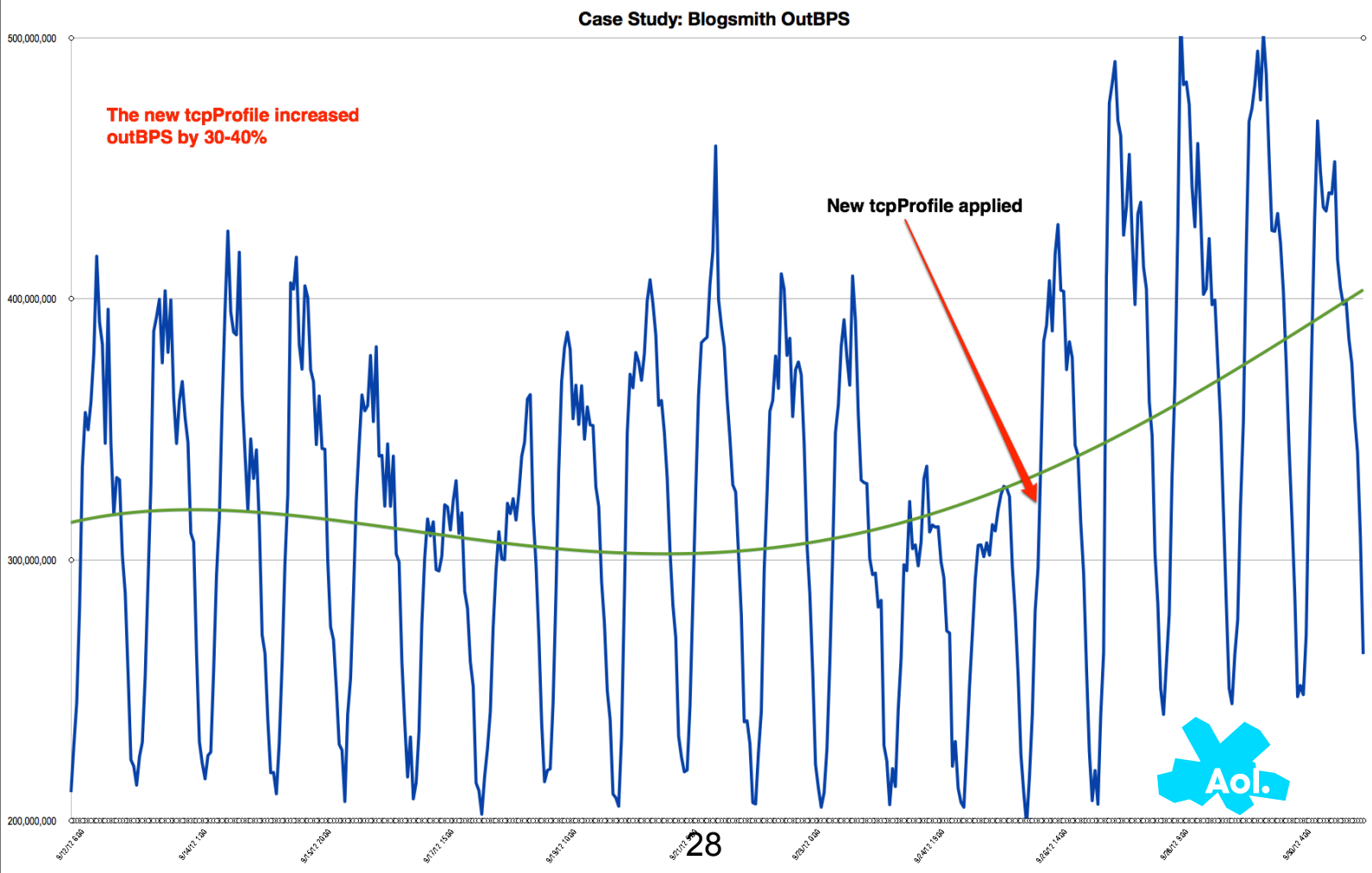
What did we change?

-WS Enabled
-SACK Enabled
-WSVal 0
-maxBurst 10
-initialCwnd 10
-delayedAck 100
-pktPerRetx 2
-slowStartIncr 4

This became the basis for the aol_vip_std_tcpprofile

Blogsmith is the AOL Content delivery platform for sites such as engadget.com.

Case Study : bps Improvements



28

Case Study: bps Improvements

Results:

- Immediate 30% jump in Outbound bps towards client.
- Retransmission rate did not increase.
- The increased in bps was sustained over the following weeks.
- Allowed higher server utilization and higher NetScaler efficiency (more bang for $$).

Blogsmith is the AOL Content delivery platform for sites such as engadget.com.

Case Study:
Retransmission Reduction

30

Case Study: Retransmission Reduction

Problem:

- While developing the Mobile and Dial tcpProfiles, we realized there were a large number of fast retransmission on the VServers.
- At times, retransmissions approached 15% of outbound traffic.
- This retransmission rate was sustained around the clock, in 3 data centers, eliminating network congestion as a cause.

Majority of traffic between 7am and 9am est is from the East Coast. This would give us an RTT of 40 ms, so the default – minRTO of 100 ms should have been fine

Case Study: Retransmission Reduction

Testing:

To reduce variables, we first tested on one pair of shared service NetScalers.

We tried to reduce the –delayedAck to 100ms

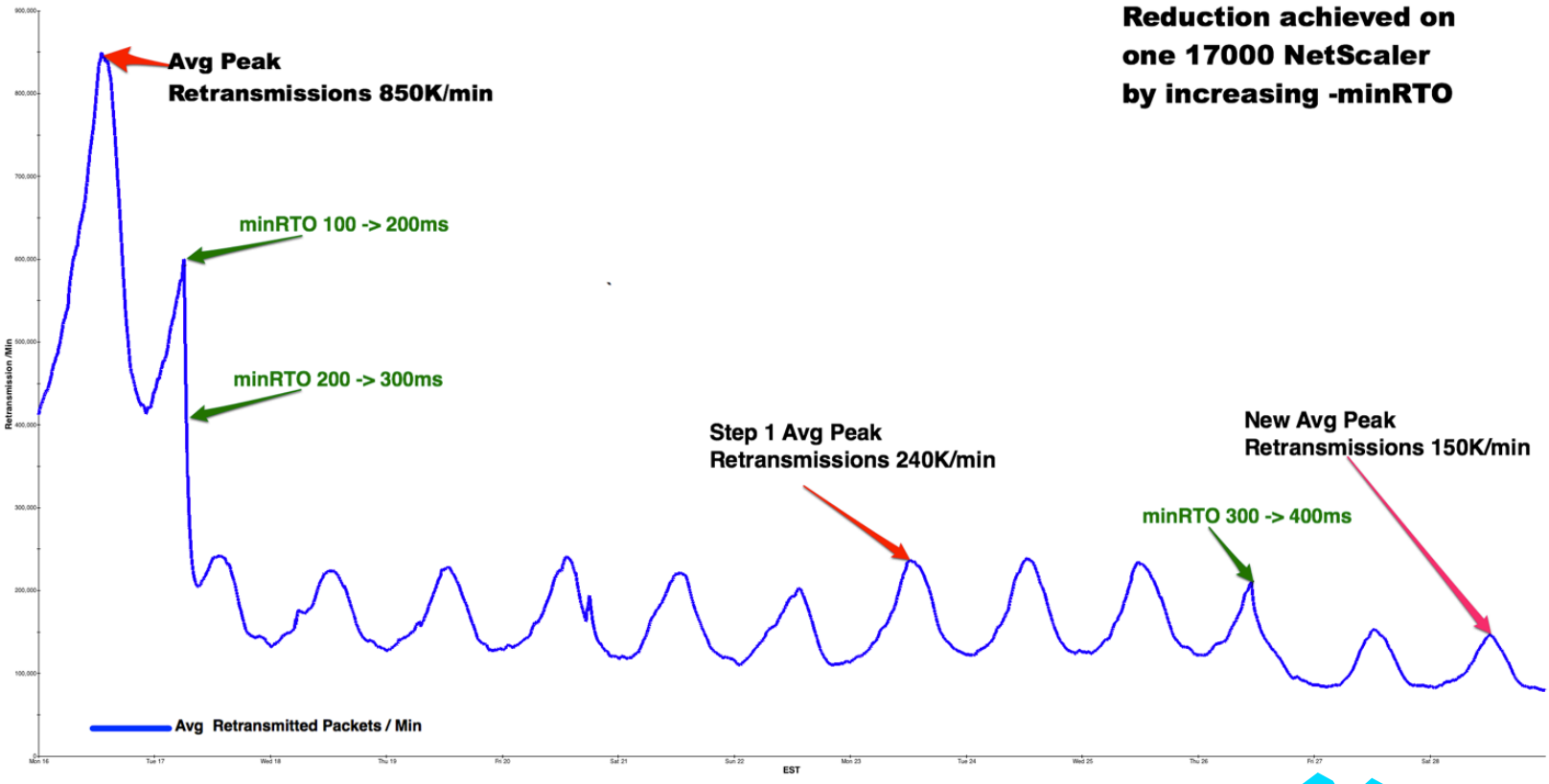
- No change was seen in the traffic over 24 hours

Next we tried increasing the –minRTO

- Instant drops in the number of Fast retransmissions, 1st - 7th retransmissions and RetransmissionGiveup.

32

Case Study: Retransmission Reduction



By simply increasing the minRTO from 100ms to 400ms
Total Retransmissions dropped from a peak of 850K/min to 150K/min
or seen another way, we dropped from from 10.2 Gb/min to 1.8 Gb/min
on one 10G link

Case Study: Retransmission Reduction

Change Details:

- Increasing the -minRTO from 100ms to 400ms
  - retransmitted packets dropped from 14.2k/sec to 2.5k/sec.
  - bps dropped from 102mb/sec to 30kb/sec on one link.
- Increasing -minRTO from 100ms to 300ms had the greatest effect on reducing retransmissions.
- When applied across AOL, total outbound bps dropped by 10%.
- However, increasing -minRTO to 500ms caused outbound bps to drop.

34

This change was added to aol_vip_std_tcpprofile

Case Study:
Engadget Coverage of New iPad Announcement

35

Case Study: Engadget Coverage of New iPad Announcement

On March 7, 2012 Apple held a Keynote event to announce the new iPad.

Traditionally, these events crush sites due to user traffic.

The iPhone 4S event in September 2011 overloaded Engadget and we decided that it would not happen again.

We used all the performance information presented here, plus other lessons learned to build a plant that could handle the projected traffic.

36

Case Study: Engadget Coverage of New iPad Announcement

Goals:

- Engadget **would not** fail during future Apple announcements.
- Customers would be able to access Engadget Live Blogs for the entire event.
- No event driven changes would be needed to the plant.

37

Case Study: Engadget Coverage of New iPad Announcement

Major Concerns:

- Keep the NetScaler CPU < 50%.
- Distribute the large initial connections per second.
- Prepare for competing sites to fail and those users shift to Engadget.
- Initially overbuild to guarantee service and measure the true capacity requirements.

38

Case Study: Engadget Coverage of New iPad Announcement

We succeeded!

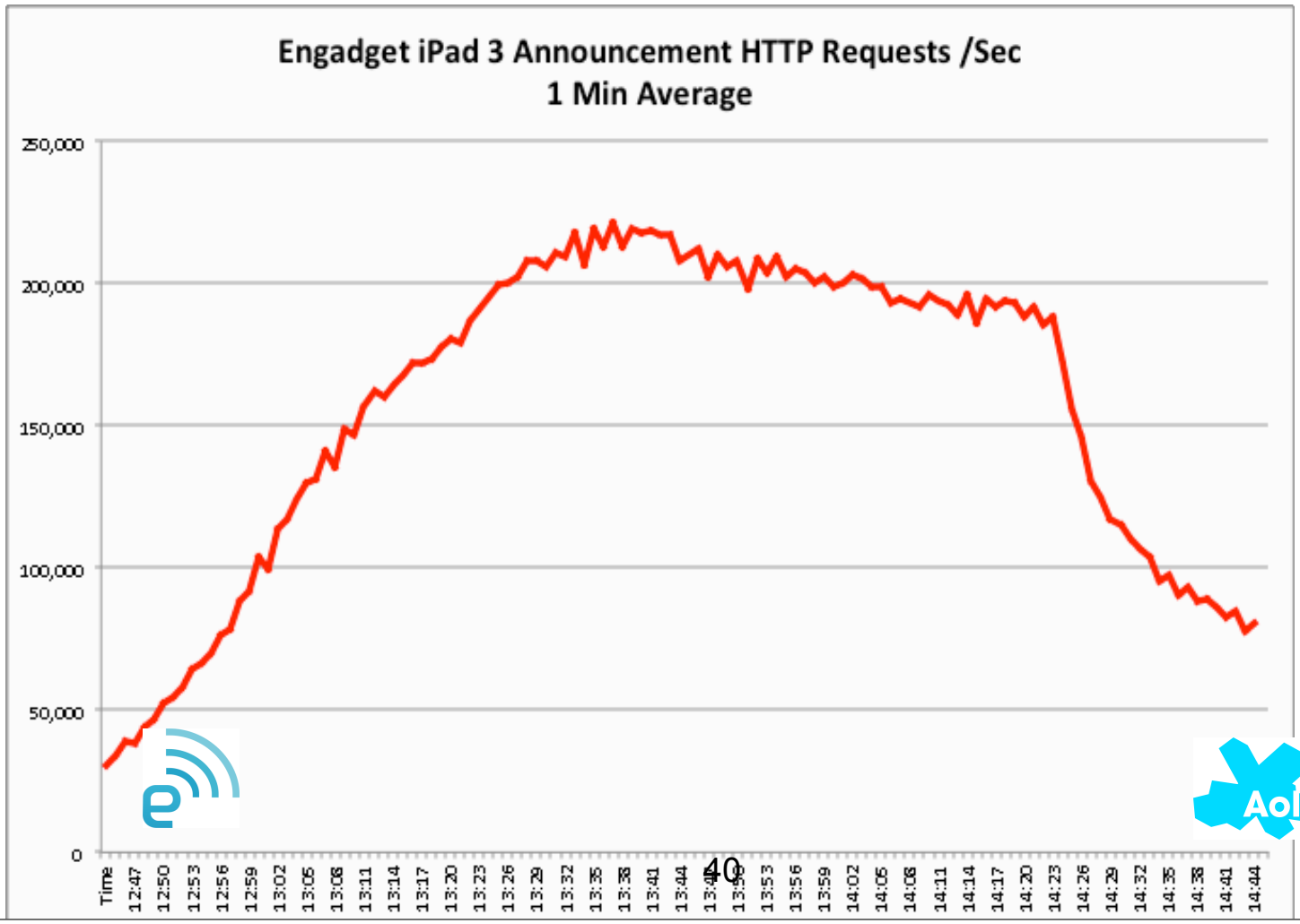
Gigaom.com 3/8/2012

"Engadget's Tim Stevens was one of the few tech media editors who only had to worry about covering the event itself, as opposed to managing the failure of his tools."

http://gigaom.com/apple/live-from-sf-sorta-why-apple-events-break-publishers/

39

Case Study: Engadget Coverage of New iPad Announcement



40

Case Study: Engadget Coverage of New iPad Announcement

Event Details:

- Event ran 120 minutes, including pre and post blogs.
- 3 second updates were served non-stop to the customers.
- A total of 1.14 Billion http requests served during the event.
- Sustained > 200K http requests /sec for 30 minutes
- Peak >220K http requests /sec over 10 min.

No changes or tweaks were needed to the Servers, Load Balancers or Network Devices during the event.

This refresh rate enables a customer to read an update before the blog scrolls.

Case Study: Engadget Coverage of New iPad Announcement

How we did it:

- Compression Bypass set for 20%
- Integrated Caching
- Flash Cache (1 sec)
- 14 non-dedicated NetScalers HA pairs to minimize connections rate, located in 5 data centers in the US and EU.
- AOL custom tcpProfiles.
- 600 cloud based, well tuned, virtual Centos servers.

This was done while maintaining normal AOL production services on the shared infrastructure!

42

It takes 10 min to deploy a cloud server, with a goal of decreasing it to 5 minutes.

Troubleshooting & Monitoring

43

Troubleshooting: General

Signs your NetScaler is getting overloaded:

- Watch for xoff frames on the switch port, indicates the NS is having trouble handling packets.
- Watch for buffer overruns, may indicate the need for additional interfaces to expand buffer pool.

nstrace.sh issues:

- Not always fully capturing packets.
- Relative frame timestamps may have a negative value due to capture mechanism.
- Additional threads added in 10.x may help, but is not a fix.

44

Monitoring: SNMP OIDs

Citrix SNMP OIDs to watch for signs of trouble

- ifTotXoffSent - .1.3.6.1.4.1.5951.4.1.1.54.1.43
- ifnicTxStalls - .1.3.6.1.4.1.5951.4.1.1.54.1.45
- ifErrRxNoBuffs - .1.3.6.1.4.1.5951.4.1.1.54.1.30
- ifErrTxNoNSB - .1.3.6.1.4.1.5951.4.1.1.54.1.31

ifTotXoffSent .1.3.6.1.4.1.5951.4.1.1.54.1.43 Number of XOFF pause frames sent on the specified interface.

ifnicTxStalls .1.3.6.1.4.1.5951.4.1.1.54.1.45 Number of system detected stalls in the transmission of packets on the specified interface. When a packet posted for transmission has not been transmitted in 4 seconds, the NIC is said to be in a transmit stall state.

ifErrRxNoBuffs .1.3.6.1.4.1.5951.4.1.1.54.1.30 Number of times the specified interface receive hardware had no packet buffer in which to place a received packet.

ifErrTxNoNSB .1.3.6.1.4.1.5951.4.1.1.54.1.31 Number of times the specified interface receive software was unable to obtain a packet buffer from the system.

Future work

What our team is working on over the next 12 months

- IPFIX and AppFlow monitoring.
- Add NetScaler interfaces to increase buffer pool.
- Custom tcpProfiles for specific applications like ad serving.
- Move to route based, active/active architecture.
- Start to roll out Jumbo Frames & WebSockets
- Investigating OpenFlow/SDN
- Replace current tools with a completely new tool set based on Trigger (to be opensourced).

46

AOL is Hiring!

As you have seen, we are NOT just your parent's dialup anymore.

http://corp.aol.com/careers

47

We value your feedback!

Take a survey of this session now in the mobile app

- Click 'Sessions' button
- Click on today's tab
- Find this session
- Click 'Surveys'

#CitrixSynergy

48

Appendix: AOL custom tcpProfile configs

```
add ns tcpProfile aol_vip_std_tcpprofile -WS ENABLED -SACK ENABLED -WSVal 0 -nagle DISABLED -ackOnPush
ENABLED -maxBurst 10 -initialCwnd 10 -delayedAck 100 -oooQSize 64 -maxPktPerMss 0 –pktPerRetx 2 -minRTO 400 -
slowStartIncr 4 -bufferSize 8190

add ns tcpProfile aol_vip_dialup-client_tcpprofile -WS ENABLED -SACK ENABLED -WSVal 0 -nagle Enabled -
ackOnPush ENABLED -maxBurst 4 -initialCwnd 4 -delayedAck 50 -oooQSize 100 -maxPktPerMss 0 -pktPerRetx 2 -
minRTO 500 -slowStartIncr 2 -bufferSize 8190

add ns tcpProfile aol_vip_mobile-client_tcpprofile -WS ENABLED -SACK ENABLED -WSVal 0 -nagle Enabled -
ackOnPush ENABLED -maxBurst 10 -initialCwnd 6 -delayedAck 50 -oooQSize 100 -maxPktPerMss 0 -pktPerRetx 2 -
minRTO 1000 -slowStartIncr 4 -bufferSize 8190

add ns tcpProfile aol_vip_server-2-vserver_tcpprofile -WS ENABLED -SACK ENABLED -WSVal 5 -nagle Enabled -
ackOnPush ENABLED -maxBurst 10 -initialCwnd 10 -delayedAck 100 -oooQSize 64 -maxPktPerMss 0 -pktPerRetx 2 -
minRTO 100 -slowStartIncr 4 -bufferSize 8190

add ns tcpProfile aol_svc_std_tcpprofile -WS ENABLED -SACK ENABLED -WSVal 3 -nagle DISABLED -ackOnPush
ENABLED -maxBurst 10 -initialCwnd 10 -delayedAck 100 -oooQSize 64 -maxPktPerMss 0 -pktPerRetx 2 -minRTO 100 -
slowStartIncr 4 -bufferSize 8190
```

Appendix: AOL custom tcpParam Config

```
set ns tcpParam -WS Enabled -WSVal 2 -SACK Enabled -maxBurst 6
-initialCwnd 4 -delayedAck 100 -downStateRST DISABLED -nagle DISABLED
-limitedPersist ENABLED -oooQSize 64 -ackOnPush ENABLED -maxPktPerMss 0
-pktPerRetx 2 -minRTO 100 -slowStartIncr 2
```

Appendix: NSCLI Show Commands

tcpParam NSCLI Commands

Show all current values

- `sh ns tcpparam -format old -level verbose`
- `sh ns tcpparam -level verbose`

tcpProfile NSCLI Commands

Show all current values

- `sh ns tcpprofile -format old -level verbose`
- `sh ns tcpprofile <profile_name> -level verbose`

Appendix: tcpParam NSCLI Commands

Set AOL custom tcpParam

```
set ns tcpParam -WS Enabled -WSVal 2 -SACK Enabled -maxBurst 6 -initialCwnd 6
-delayedAck 100 -downStateRST DISABLED -nagle DISABLED -limitedPersist ENABLED
-oooQSize 64 -ackOnPush ENABLED -maxPktPerMss 0 -pktPerRetx 2 -minRTO 100
-slowStartIncr 2
```

Set a specific value

```
set ns tcpParam -WS enabled
```

Changing tcpparam modifies the same values in the default tcpprofile.

Appendix: tcpProfile NSCLI Commands

Creating new profiles

```
add ns tcpProfile aol_vip_std_tcpprofile -WS ENABLED -SACK ENABLED -WSVal 0 -nagle DISABLED
-ackOnPush ENABLED -maxBurst 10 -initialCwnd 10 -delayedAck 100 -oooQSize 64 -maxPktPerMss 0
-pktPerRetx 2 -minRTO 400 -slowStartIncr 4 -bufferSize 8190 –mss 0
```

Changing a value

```
Set ns tcpProfile aol_vip_std_tcpprofile -WSVal 3
```

Applying Profiles

```
set cs vserver <vservername:port> -tcpprofile <profilename>
set cs vserver www_aol_com:80 -tcpprofile aol_vip_std_tcpprofile
set lb vserver <vservername:port> -tcpprofile <profilename>
set service <servicename:port> -tcpprofile <profilename>
```

Resetting to Default

```
unset cs vserver <vservername:port> -tcpprofile
unset lb vserver <vservername:port> -tcpprofile
unset service <servicename:port> -tcpprofile
```

Explanation of TCPProfile variables. (V9.2 & 9.3)

Variable	Default	Min	Max	Description
-name	----	----	----	The name for a TCP profile. A TCP profile name can be from 1 to 127 characters and must begin with a letter, a number, or the underscore symbol (_). Other characters allowed after the first character in a name are the hyphen, period, pound sign, space, at sign, and equals.
-WS	Disabled	---	---	Enable or disable window scaling. If Disabled, Window Scaling is disabled for both sides of the conversation.
-WSVal	4	0	8	The factor used to calculate the new window size.
-maxburst	6	1	255	The maximum number of TCP segments allowed in a burst. The higher this value, the more frames are able to be sent at one time.
-initialCwnd	4	2	44	The initial maximum upper limit on the number of TCP packets that can be outstanding on the TCP link to the server. As of 9.2.50.1, this number was upped from 6 to 44
-delayedAck	100	10	300	The time-out for TCP delayed ACK, in milliseconds.
-oooQSize	64	0	65535	The maximum size of out-of-order packets queue. A value of 0 means infinite. The name is a misnomer, this buffer contains sent frames that are awaiting acks or received frames that are not sequential, meaning some packets are missing due to SACK.
-maxPktPerMss	0	0	512	The maximum number of TCP packets allowed per maximum segment size (MSS). A value of 0 means that no maximum is set.
-pktPerRetx	1	1	512	The maximum limit on the number of packets that should be retransmitted on receiving a "partial ACK". Partial ACK are ACKs indicating not all outstanding frames were acked.
-minRTO	100	10	64000	The minimum Receive Time Out (RTO) value, in milliseconds. The NetScale supports New Reno and conforms to RFC 2001 and RFC 5827 (?). Since the Netscaler does not use TCP Timestamping, these values do not correspond to actual propagation delays.
-slowStartIncr	2	1	100	Multiplier determines the rate which slow start increases the size of the TCP transmission window after each ack of successful transmission.
-SACK	Disabled	---	---	Enable or disable selective acknowledgement (SACK). There is NO reason this should be off
-nagle	Disabled	---	---	Enable or disable the Nagle algorithm on TCP connections. When enabled, reduces the number of small segments by combining them into one packet. Primary use is on slow or congested links such as mobile or dial.
-ackOnPush	Enabled	---	---	Send immediate acknowledgement (ACK) on receipt of TCP packets with the PUSH bit set.
-mss (v 9.3)	0	0	1460	Maximum segment size, default value 0 uses global setting in “set tcpparam”, maximum value 1460
-bufferSize (v9.3)	8190	---	4194304	The value that you set is the minimum value that is advertised by the NetScaler appliance, and this buffer size is reserved when a client initiates a connection that is associated with an endpoint-application function, such as compression or SSL. The managed application can request a larger buffer, but if it requests a smaller buffer, the request is not honored, and the specified buffer size is used. If the TCP buffer size is set both at the global level and at the entity level (virtual server or service level), the buffer specified at the entity level takes precedence. If the buffer size that you specify for a service is not the same as the buffer size that you specify for the virtual server to which the service is bound, the NetScaler appliance uses the buffer size specified for the virtual server for the client-side connection and the buffer size specified for the service for the server-side connection. However, for optimum results, make sure that the values specified for a virtual server and the services bound to it have the same value. The buffer size that you specify is used only when the connection is associated with endpoint-application functions, such as SSL and compression. Note: A high TCP buffer value could limit the number of connections that can be made to the NetScaler appliance.

54