



# LinkedIn ATS9 Migration

Xin Li  
Lokesh Jindal  
Sudheer Vinukonda

# LinkedIn - ATS9 Migration

- ATS use cases in LI
  - Edge PoP (TLS Offloading/H2 etc)
  - Data Center (Primary Reverse Proxy for LI)
    - ❖ Multiple Tiers
    - ❖ Custom Plugins
  
- ATS9 Migration
  - 6.2+ to 9.0+
  - Data Centers Completed
  - No plans to update PoP

# ATS9 Migration - Two Stages

- ATS8 Migration
  - Almost 1 year project
  - Migration for both I0 and I1
  - Get to canary testing in staging environment
  
- ATS9 Migration
  - Nov. 2019 - Jul. 2020
  - Migration for I1
  - Roll out to PROD

## ATS9 Migration - Effort Summary

- Significant changes
  - New ATS APIs
  - Config Changes
  - Compiler update (gcc 7.3.0, c++17)
  - Yaml code change
  - ...
  
- A large development effort
  - Migrated 200+ local patches
  - Rebuild 30+ 3rd party libs
  - Migrated 60+ ats plugins/libraries
  - Submitted 420+ internal code changes
  - Contributed 80+ PRs over the last 2 years

# ATS9 Migration - Process & Strategy

- Branch based development  
Feature branch for ats9 and master for ats6
- Migration by order  
traffic\_server  
3rd party libraries  
ATS libraries  
ATS plugins
- Testing and verification  
Ats regression, AuTest  
Functional test  
Integration test  
Test in Staging environment  
Test in PROD
- Custom scripts and tools  
Git  
Version update  
Email Reminder/Follow ups

## Issues during ATS 9 upgrade

- 80+ PRs during last 2 years
- Presenting top 3 issues highlighting symptoms, debug methodology and fix
  - ◆ Dynamic Plugin Reload
  - ◆ Connection failure with origins
  - ◆ Buffer overflow and dynamic\_cast crashes
- Debug tools/methodology:
  - ◆ Gdb
  - ◆ Debug tags
  - ◆ Core file analysis
  - ◆ ASAN
  - ◆ tcpdump
  - ◆ Deep dive into code

# Dynamic Plugin Reload

- Symptoms [dev]: “Plugin level” test failed. Plugin used as global as well remap plugin with shared state initialized during loading.
- Debug:
  - ◆ Gdb showed two breakpoints
  - ◆ Expected only one copy of DSO to be loaded, getting two
  - ◆ Deep dive into code
- Fix: Built support for a new config to disable the feature

Ref: [PR 6421](#)



## Connection failure with origins

→ Symptoms [E]: Increased 502 for many python-based origins

→ Debug:

◆ **Enabled debug tag** "ssl"

```
DEBUG: <SSLUtils.cc:1788 (SSLConnect)> (ssl.error.connect) SSL connect returned -1, ssl_error=1, ERR_get_error=336151568  
(error:14094410:SSL routines:ssl3_read_bytes:sslv3 alert handshake failure)
```

```
DEBUG: <SSLNetVConnection.cc:1441 (sslClientHandShakeEvent)> (ssl-diag) SSL::47498867529472:error:14094410:SSL  
routines:ssl3_read_bytes:sslv3 alert handshake failure:ssl/record/rec_layer_s3.c:1528:SSL alert number 40: peer address is 127.0.0.1
```

◆ Reproduced ssl handshake failure in dev

◆ Compared **tcpdump** for handshake for ATS 6 and ATS 9

- Cipher suite mismatch. [TLS\\_ECDHE\\_RSA\\_WITH\\_AES\\_256\\_CBC\\_SHA \(0xc014\)](#) not present in ATS 9

→ Fix: Enabled the cipher suite in our configs.

→ Note: Another similar issue where origins were using TLS 1.0, which is deprecated by default in ATS9. We enabled TLS 1.0 support in configs, and are moving those origins to newer TLS versions.

◆ Used command "openssl s\_client -connect" to check different TLS versions





# Buffer overflow and dynamic\_cast crashes

→ Symptoms: Crash every few minutes in prod canary

→ Debug:

## ◆ Core file analysis

## ◆ Code deep dive

- Dynamic\_cast for “all” requests seemed unnecessary

## ◆ Ran with ASAN and...

- Bingo! Buffer overflow resulting in corrupting global variables when the unmapped URL is pointing to the global INVALID\_STR (default initial value). A new commit in ats9 did not play well with all use cases.

→ Fix:

- ◆ Fixed the code causing overflow in “logfilter” (PR 6950)
- ◆ Avoid dynamic\_cast when not needed.
- ◆ A bunch of other corruption related fixes in parallel.

→ Note:

- ◆ ASAN in a prod-box survives only for half an hour due to CPU usage
- ◆ ASAN exposed another use-after-free issue (string::c\_str() related) at multiple places in our plugins

```
(gdb) bt
#0 0x00002b8ea38987e in __cxxabi1::_dynamic_cast (src_ptr=0x2b89451cb5d0, src_type=0x81dd48 <typeinfo for
NetVConnection>, dst_type=dst_type@entry=0x7c49d8 <typeinfo for PluginIdentity>, src2dst=src2dst@entry=-2)
    at ../.././libstdc++-v3/libsupc++/dyncast.cc:71
#1 0x00000000071d0e8 in ProxyTransaction::new_transaction (this=this@entry=0x2b894a65e240,
from_early_data=<optimized out>) at ProxyTransaction.cc:46
#2 0x000000000523c11 in new_transaction (this=0x2b894a65ded0) at HttpClientSession.cc:473
#3 HttpClientSession::state_keep_alive (this=0x2b894a65ded0, event=100, data=<optimized out>) at
HttpClientSession.cc:388
#4 0x000000000770393 in handleEvent (data=0x2b89451cb7b0, event=100, this=0x2b894a65ded0) at
/home/syinkon/Traffic/ATS/ats9/ats-core_trunk/ats9/src/locore/eventsystem/I_Continuation.h:190
#5 read_signal_and_update (vc=0x2b89451cb5d0, event=100) at UnixNetVConnection.cc:83
#6 UnixNetVConnection::readSignalAndUpdate (this=this@entry=0x2b89451cb5d0, event=event@entry=100) at
UnixNetVConnection.cc:1016
#7 0x000000000741d53 in SSLNetVConnection::net_read_io (this=0x2b89451cb5d0, nh=0x2b88ef623d80,
lthread=<optimized out>) at SSLNetVConnection.cc:671
```

Ref: [Issue-6946](#), [PR-6950](#), [Issue-6853](#), [PR-6868](#), [Issue-6850](#), [PR-6869](#)



# ATS9 Migration - SSL Corruption

→ [Issue 7096](#), [Issue 7284](#)

```
(gdb) bt
#0 0x00002b74100ed31e in asni_string_embed_free () from /lib/libcrypto.so.1.1
#1 0x00002b74100f8eaf in asni_primitive_free () from /lib/libcrypto.so.1.1
#2 0x00002b74100f8e89 in asni_primitive_free () from /lib/libcrypto.so.1.1
#3 0x00002b74100f9240 in asni_template_free () from /lib/libcrypto.so.1.1
#4 0x00002b74100f8fde in asni_item_embed_free () from /lib/libcrypto.so.1.1
#5 0x00002b74100f9240 in asni_template_free () from /lib/libcrypto.so.1.1
#6 0x00002b74100f8fde in asni_item_embed_free () from /lib/libcrypto.so.1.1
#7 0x00002b74100f9240 in asni_template_free () from /lib/libcrypto.so.1.1
#8 0x00002b74100f8fde in asni_item_embed_free () from /lib/libcrypto.so.1.1
#9 0x00002b74100f9240 in asni_template_free () from /lib/libcrypto.so.1.1
#10 0x00002b74100f8fde in asni_item_embed_free () from /lib/libcrypto.so.1.1
#11 0x00002b74100f9185 in ASN1_item_free () from /lib/libcrypto.so.1.1
#12 0x00002b7410246f11 in OPENSSL_sk_pop_free () from /lib/libcrypto.so.1.1
#13 0x00002b740fe03eca in SSL_SESSION_free () from /lib/libssl.so.1.1
#14 0x00002b740ffdfbba in SSL_free () from /lib/libssl.so.1.1
#15 0x000000000736c78 in SSLNetVConnection::clear (this=0x2b7592c85aa0) at SSLNetVConnection.cc:928
#16 0x000000000737374 in SSLNetVConnection::free (this=0x2b7592c85aa0, t=0x2b7413c00000) at SSLNetVConnection.cc:928
#17 0x00000000075d2af in NetHandler::free_netevent (this=0x2b7413c0a00, ne=neentry=0x2b7592c85c50) at UnixNet.cc:104
#18 0x00000000076f27e in read_signal_and_update (vc=0x2b7592c85aa0, event=100) at UnixNetVConnection.cc:104
#19 UnixNetVConnection::readSignalAndUpdate (this=this@entry=0x2b7592c85aa0, event=event@entry=100) at UnixNetVConnection.cc:104
#20 0x00000000073d6f3 in SSLNetVConnection::net_read_io (this=0x2b7592c85aa0, nh=0x2b7413c0a00, lthread=optin) at UnixNetVConnection.cc:104
#21 0x00000000075d5e8 in NetHandler::process_ready_list (this=this@entry=0x2b7413c0a00) at UnixNet.cc:412
#22 0x00000000075d8dd in NetHandler::waitForActivity (this=0x2b7413c0a00, timeout=optimized out) at UnixNet.cc:412
#23 0x0000000007b060a in EThread::execute_regular (this=this@entry=0x2b7413c06000) at UnixEThread.cc:266
#24 0x0000000007ba332 in EThread::execute (this=0x2b7413c06000) at UnixEThread.cc:327
#25 0x0000000007b06d9 in spawn_thread_internal (a=0x2b7412d69600) at Thread.cc:192
#26 0x00002b7410a93dd5 in start_thread () from /lib64/libpthread.so.0
#27 0x00002b7411844ead in clone () from /lib64/libc.so.6
```

→ Transform Plugins (ESI/ComboHandler)

◆ [PR 7096](#), [PR 7284](#)

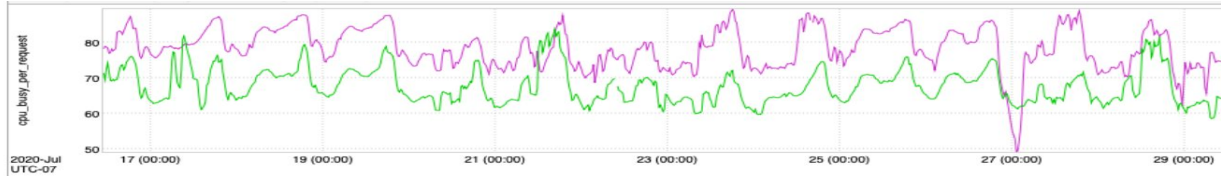
→ VC Event History

→ SSL->VC Bookkeeping

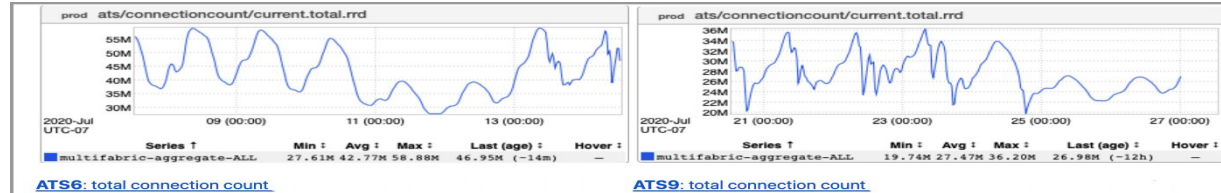
```
...
location = {
  file = 0x02826b6 "read_signal_and_update",
  func = 0x7e8750 "ServerSessionPool::eventHandler",
  line = -1056816000
},
event = 104,
reentrancy = 1
}, {
  location = {
    file = 0x7e8420 "HttpSessionManager.cc",
    func = 0x7e8820 "ServerSessionPool::eventHandler(int, void):_FUNCTION__eventHandler",
    line = 285
  },
  event = 0,
  reentrancy = 0
}, {
  location = {
    file = 0x7e8158 "Http1ServerSession.cc",
    func = 0x7e8300 "Http1ServerSession::do_io_close(int):_FUNCTION__do_io_close",
    line = 120
  },
  event = 0,
  reentrancy = 0
}, {
  location = {
    file = 0x81c03b "do_io_close",
    func = 0x7e8750 "ServerSessionPool::eventHandler",
    line = 8292176
  }
}
```

# ATS9 Migration - Wins

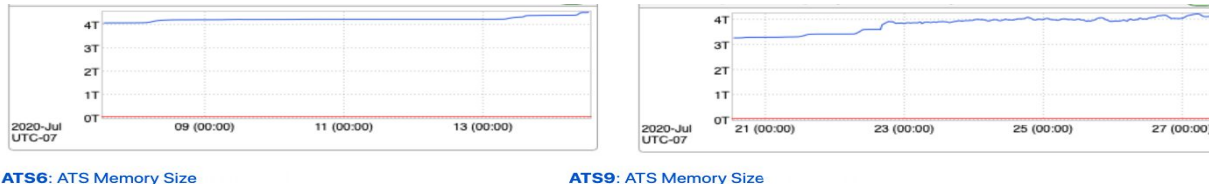
- New features : Proxy Protocol, Event loop Visibility etc
- Decrease in cpu/req Usage 10-15%



- Reduce Origin Connections 35-50%



- Decrease in Memory Usage about 20%



Back up slides

# Scheduling continuation and transform

- Symptoms [dev]: Segmentation fault or assert
- Debug:
  - Core dump file analysis, code deep dive
  - Assert during mutex operations since continuation was scheduled from “pthread”, not “TS thread”
  - Another assert  
`_ink_assert (expression=expression@entry=0x828118 "!\"BUG: It must have acquired the NetHandler's lock before doing anything on keep_alive_queue.\"", ... at ink_assert.cc:37`
- Fix:
  - Assert in TSContSchedule\* APIs to check that “scheduling” thread is a “TS Thread”
  - Schedule Transform on the same thread as the continuation
  - When using TSContSchedule() and TSContScheduleAPI() set the calling thread as the thread affinity when not already set

## Connection failure with origins (Part 1 of 2)

→ Symptoms [dev]:

- ◆ SSL connection with origin failed due to unsupported protocol

```
[ET_NET 15] DEBUG: <SSLNetVConnection.cc:1487 (sslClientHandShakeEvent)> (ssl-diag)
SSL::47981136992000:error:1425F102:SSL routines:ssl_choose_client_version:unsupported
protocol:ssl/statem/statem_lib.c:1940: peer address is 10.190.199.148
```

→ Debug:

- ◆ Simple - used command “openssl s\_client -connect” to check with different TLS versions

→ Fix:

- ◆ TLS 1.0 and 1.1 disabled by default in ATS9. Re-enabled in our configs.
- ◆ Origins would be moving to new TLS versions.

## Connection failure with origins (Part 2 of 2)

→ Symptoms [EI]: Increased 502 for many python-based origins

→ Debug:

- ◆ Enabled “ssl debug tag”

```
DEBUG: <SSLUtils.cc:1788 (SSLConnect)> (ssl.error.connect) SSL connect returned -1, ssl_error=1,
ERR_get_error=336151568 (error:14094410:SSL routines:ssl3_read_bytes:sslv3 alert handshake failure)
```

```
DEBUG: <SSLNetVConnection.cc:1441 (sslClientHandShakeEvent)> (ssl-diag)
SSL::47498867529472:error:14094410:SSL routines:ssl3_read_bytes:sslv3 alert handshake
failure:ssl/record/rec_layer_s3.c:1528:SSL alert number 40: peer address is 127.0.0.1
```

- ◆ Reproduced ssl handshake failure in dev
- ◆ Compared tcpdump for handshake for ATS 6 and ATS 9
  - Cipher suite mismatch ( [TLS\\_ECDHE\\_RSA\\_WITH\\_AES\\_256\\_CBC\\_SHA \(0xc014\)](#) not present in ATS 9

→ Fix: Enabled the cipher suite in our configs.



# Buffer overflow and dynamic\_cast crashes

→ Symptoms: Crash every few minutes in a box out of 700 in prod

→ Debug:

## ◆ Core dump file analysis

(gdb) bt

#0 0x00002b88ea38987e in \_\_cxxabiv1::\_\_dynamic\_cast (src\_ptr=0x2b89451cb5d0, src\_type=0x81dd48 <typeinfo for NetVConnection>, dst\_type=dst\_type@entry=0x7c49d8 <typeinfo for PluginIdentity>, src2dst=src2dst@entry=-2)

at ../../../../libstdc++-v3/libsupc++/dyncast.cc:71

#1 0x000000000071d0e8 in ProxyTransaction::new\_transaction (this=this@entry=0x2b894a65e240, from\_early\_data=<optimized out>) at ProxyTransaction.cc:46

#2 0x0000000000523c11 in new\_transaction (this=0x2b894a65ded0) at Http1ClientSession.cc:473

#3 Http1ClientSession::state\_keep\_alive (this=0x2b894a65ded0, event=100, data=<optimized out>) at Http1ClientSession.cc:388

#4 0x0000000000770393 in handleEvent (data=0x2b89451cb7b0, event=100, this=0x2b894a65ded0) at /home/svinukon/Traffic/ATS/ats9/ats-core\_trunk/ats9/src/iocore/eventsystem/I\_Continuation.h:190

#5 read\_signal\_and\_update (vc=0x2b89451cb5d0, event=100) at UnixNetVConnection.cc:83

#6 UnixNetVConnection::readSignalAndUpdate (this=this@entry=0x2b89451cb5d0, event=event@entry=100) at UnixNetVConnection.cc:1016

#7 0x0000000000741d53 in SSLNetVConnection::net\_read\_io (this=0x2b89451cb5d0, nh=0x2b88ef623d80, lthread=<optimized out>) at SSLNetVConnection.cc:671





## Buffer overflow and dynamic\_cast crashes (contd ...)

### → Debug:

- ◆ code deep dive
  - Dynamic\_cast for “all” requests seemed unnecessary
- ◆ Ran ASAN in EI and...
  - Bingo! Buffer overflow resulting in corrupting global variables when the unmapped URL is pointing to the global INVALID\_STR (default initial value)
  - A new commit in ats9 did not play well

### → Fix:

- ◆ Fixed the code causing overflow in “logfilter”
- ◆ Avoid dynamic\_cast when not needed.
- ◆ A bunch of other corruption related fixes in parallel.

### → Note:

- ◆ ASAN in a prod-box survives only for half an hour due to CPU usage
- ◆ ASAN exposed another use-after-free issue (string::c\_str() related) at multiple places in code