ATS & Yahoo Homepage

PRESENTED BY **Pushkar Sachdeva (psachdev@yahoo-inc.com)** | October 27, 2016

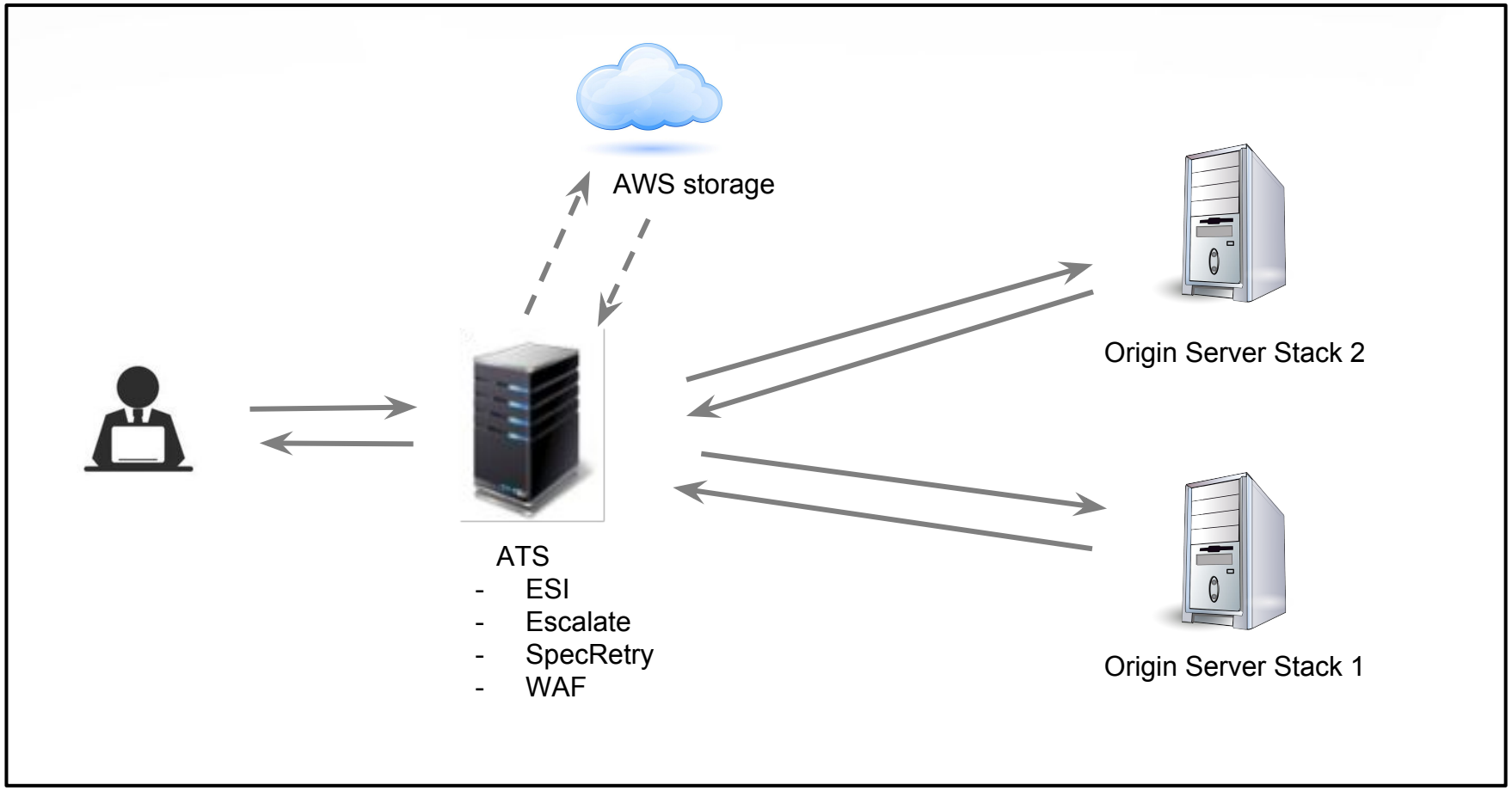
Agenda

- Architecture Overview
- Plugins
  - Speculative Retry
  - Escalate & Improvements
  - Brotli
  - Unit Testing for CPP plugins

YAHOO!

Architecture Overview

Architecture Overview



YAHOO!

Architecture Overview - Why ATS ?

- Proxy
- Cache & ESI
- Resiliency
  - failsafe
  - speculative retry
  - WAF
- Other Features
  - e.g. location based service

YAHOO!

Plugins

Speculative Retry Plugin - Motivation

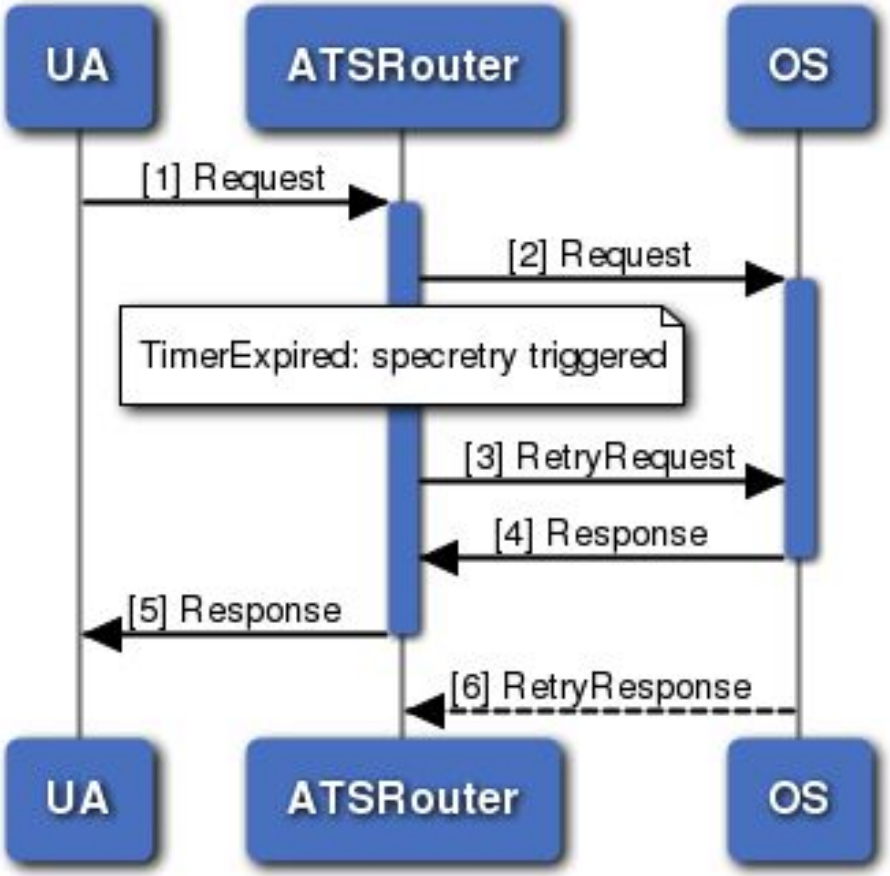
- Yahoo! uses Node.js for serving
- Not very reliable
- Results in ~1.5% fallbacks
- Multiple points of failure
- A simple request retry could improve things

Speculative Retry Plugin - When to retry ?

- Timer Based
- Negative Responses

YAHOO!

Speculative Retry Plugin - Timer Based



YAHOO!

Speculative Retry Plugin - Implementation

- It's an 'intercept' plugin
- Uses atscppapi

```
void SpeculativeRetryPlugin::handleInputComplete() {
    SendOrigRequest(orig_url);
    Async::execute<AsyncTimer>(this, new
AsyncTimer(timer_value));
}
```

YAHOO!

Speculative Retry Plugin - Implementation

- Sends retry request on timer expiry if needed

```
void SpeculativeRetryPlugin::handleAsyncComplete(AsyncTimer &a)
{
    if(active_fetch_index_ == -1) {
       //original response not yet received
        sendRetryRequest(orig_url);
        retry_type_ = TIMER_EXPIRY;
    }
}
```

YAHOO!

Speculative Retry Plugin - Implementation

- Response that comes first is streamed back
- The other is discarded

```
void SpeculativeRetryPlugin::handleAsyncComplete(AsyncHttpFetch &async_http_fetch) {
    if( active_fetch_index_ != -1) {
        //already received a response before, if not same then discard
        if(async_req_list_[active_fetch_index_]  != &async_http_fetch) {
            async_http_fetch.cancel();
            return ;
        }
    } else {
        // first response received ; skipped negative response handling for simplicity
        if(async_http_fetch.getResult() == AsyncHttpFetch::RESULT_HEADER_COMPLETE) {
            // mark the response being picked by setting active_fetch_index
            for(unsigned int i = 0; i < async_req_list_.size(); ++i) {
               if(async_req_list_[i] == &async_http_fetch)
                       active_fetch_index_ = i;
               else
                    async_req_list_[i]->cancel();
            }
        }
    }
    streamResponseBack(async_http_fetch)
}
```

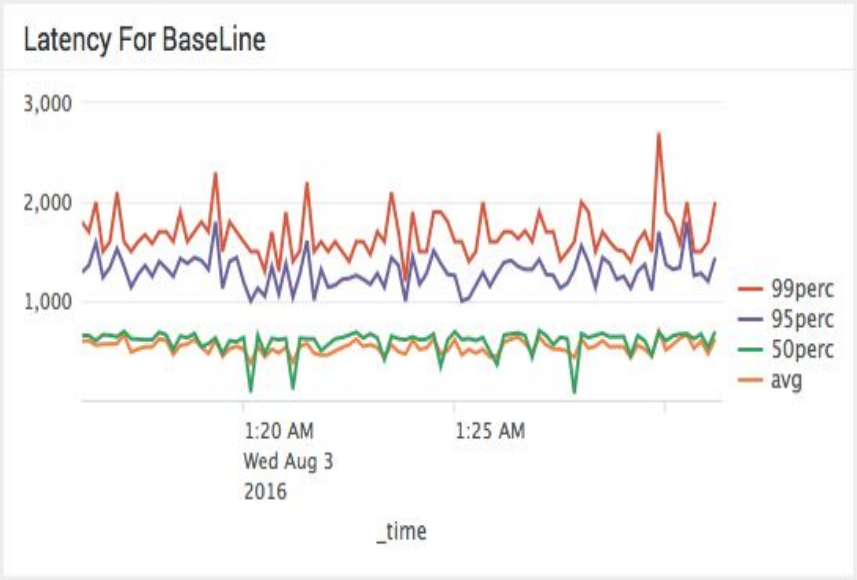
YAHOO!

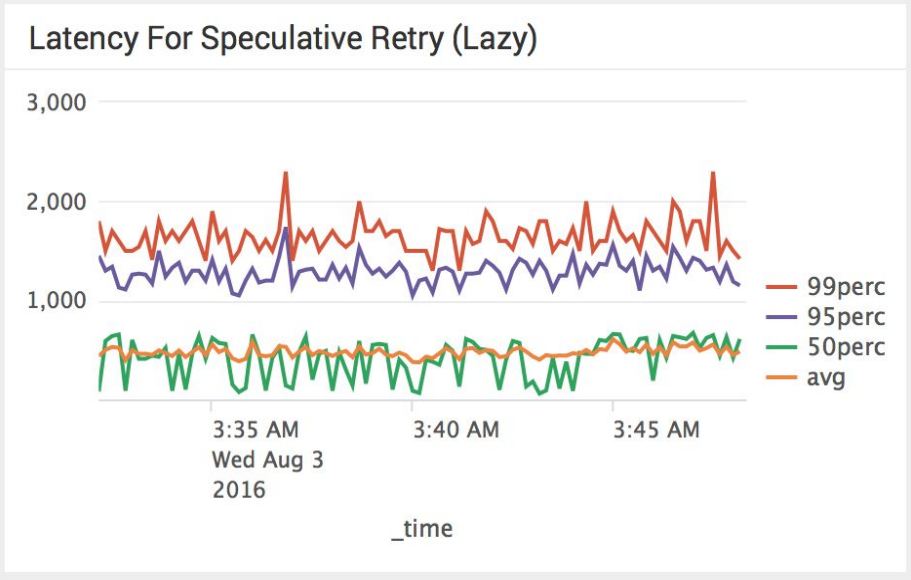
Speculative Retry Plugin - Additional Features

- Retries can be rate limited
- Bypass plugin by passing '`X-Bypass-Speculative`' header
- Ability to configure number of retries needed
- Ability to retry for specific endpoints

Speculative Retry Plugin - Results

- Reduced latency





YAHOO!

Speculative Retry Plugin - Results

- Reduced number of fallbacks

ESI Batch Status for Baseline

s	count	percent
Success	13206	99.622812
Fallback (Stream)	43	0.324381
Fallback (App)	7	0.052806

ESI Batch Status for Speculative Retry (Aggresive)

s	count	percent
Success	15585	99.782316
Fallback (Stream)	32	0.204879
Fallback (App)	2	0.012805

YAHOO!

Escalate Plugin

```
map cdn.example.com origin.example.com
  @plugin=escalate.so @pparam=401,404,410,502:second-origin.example.com @pparam=--pristine
```

- Remap plugin
- Handle bad response code from origin server
- Use TSHttpTxnRedirectUrlSet() to try out a different host or URL
- Won’t work if origin server is not responding

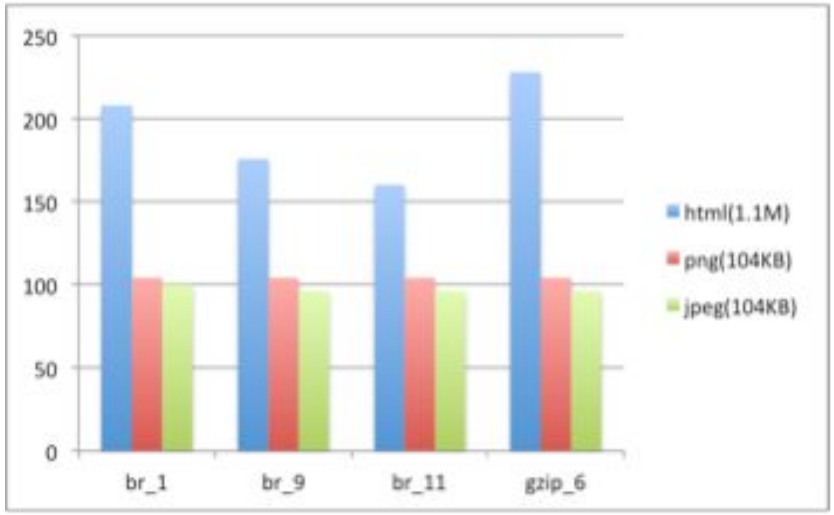
Escalate Plugin Improvements

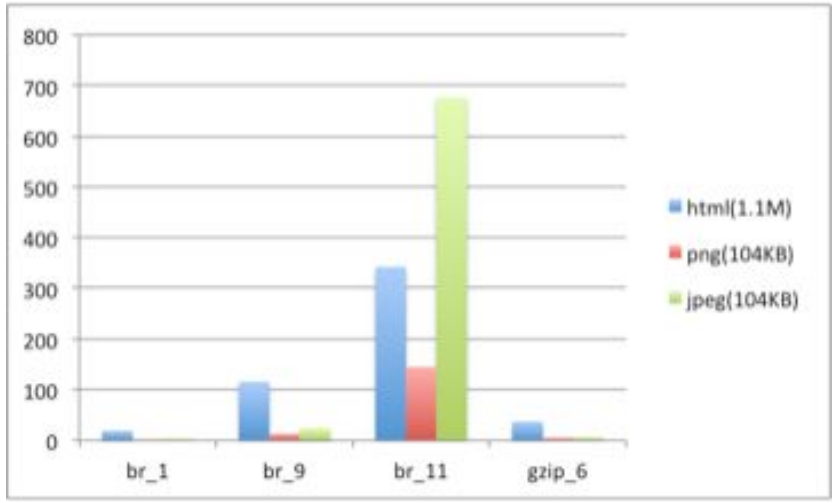
- Implement TSRemapOSResponse() for remap plugin
  - called when there is a origin server response (good or bad or no connection)
  - “TSServerState” passed as argument
  - If not TS_SRVSTATE_CONNECTION_ALIVE, take action!
- Quirks
  - If you have multiple remap plugins for the same remap.config line, only the first one’s TSRemapOSResponse() will be called.
  - It can be called multiple times due to connect attempt retries
  - It is called after READ_RESPONSE hook
- TS-4587

Brotli Plugin - About Brotli

- It is a compression algorithm developed by Google
- Better compression ratio than gzip
- But compression speed is slow
- Could be useful for compressing cacheable static responses (like JS/CSS files)
- Supported by chrome (v>49) and firefox (v>44)
- Brotli is ‘HTTPS’ only

Brotli Plugin - Brotli Vs Gzip





YAHOO!

Brotli Plugin

- Compresses response using brotli algorithm if brotli supported by client
- If response is gzipped then it decompresses it and compresses it again using brotli
- Supports streaming
- PR - https://github.com/apache/trafficserver/pull/776
- Jira - TS-4553

YAHOO!

Brotli Plugin - Config Options

- Brotli compression quality
- List of blacklisted Content-Types
- Disable ‘proxy.config.http.normalize_ae_gzip’ setting in records config

Unit Testing for CPP Plugins

- Mock ‘atscppapi’ using googlemocks
  - PR - https://github.com/apache/trafficserver/pull/408
  - Jira - TS-4103
- Has same namespace and directory structure

Unit Testing for CPP Plugins

```
class Transaction: noncopyable {
public:

  ~Transaction() { }

  //shared_ptr<ContextValue> getContextValue(const std::string &key);
  MOCK_METHOD1(getContextValue, shared_ptr<ContextValue> (const std::string &key));

  //void setContextValue(const std::string &key, shared_ptr<ContextValue> value);
  MOCK_METHOD2(setContextValue,  void (const std::string&, shared_ptr<ContextValue>));

  //void resume();
  MOCK_METHOD0(resume, void ());

  //void error();
  MOCK_METHOD0(error, void ());
  .......
  ....
};
```

YAHOO!

Unit Testing for CPP Plugins

- Compile and link libatscppapi_mock

```
+AM_CPPFLAGS = -I$(top_builddir)/lib/atscppapi/mocks/ -I../ -I.
+bin_PROGRAMS = webp_test
+
+webp_test_SOURCES = Magick++.cc ImageTransform.cc unit_test.cc
+webp_test_LDADD = -L$(top_builddir)/lib/atscppapi/mocks/ -latscppapi_mock -lgtest
-lgmock -lpthread
```

YAHOO!

Unit Testing for CPP Plugins

- Sample Code

```
void GlobalHookPlugin::handleReadResponseHeaders(Transaction &transaction)
{
   string ctype = transaction.getServerResponse().getHeaders().values("Content-Type");
   if (ctype.find("jpeg") != string::npos) {
        TS_DEBUG(TAG, "Content type is either jpeg. Converting to webp");
        transaction.addPlugin(new ImageTransform(transaction));
    }
   transaction.resume();
}
```

YAHOO!

Unit Testing for CPP Plugins

- Write testcase

```
TEST_F(GlobalHookPluginTest, NotSupportedContentType)
{
 void *arg1 = NULL;
 Headers hdrs;
 Response response;
 Transaction transaction(arg1);
 EXPECT_CALL(transaction, getServerResponse()).WillOnce(ReturnRef(response));
 EXPECT_CALL(response, getHeaders()).WillOnce(ReturnRef(hdrs));
 EXPECT_CALL(hdrs, values(“Content-Type”)).WillOnce(Return("image/png"));
 EXPECT_CALL(transaction, addPlugin(_)).Times(0);
 EXPECT_CALL(transaction, resume()).Times(1);
 GlobalHookPlugin globalHookPlugin;
 globalHookPlugin.handleReadResponseHeaders(transaction);
}
```

YAHOO!

Thanks!