



ApacheCon 2009 Traffic Server Meetup

November 3, 2009

Overview

- What is Traffic Server?
- Performance
- How it is used
- History
- Why did we open source?
- The open source process
- The architecture
- Future projects
- How to help



What is Traffic Server?

- HTTP and HTTPS proxy cache server
 - Reverse proxy
 - Forward proxy
- Multi-threaded, event driven asynchronous state machine
- Extensible plug-in architecture
 - Remap URLs;
 - State Machine hooks
 - Handle other protocols (FTP, SMTP, SOCKS, RTSP, etc)

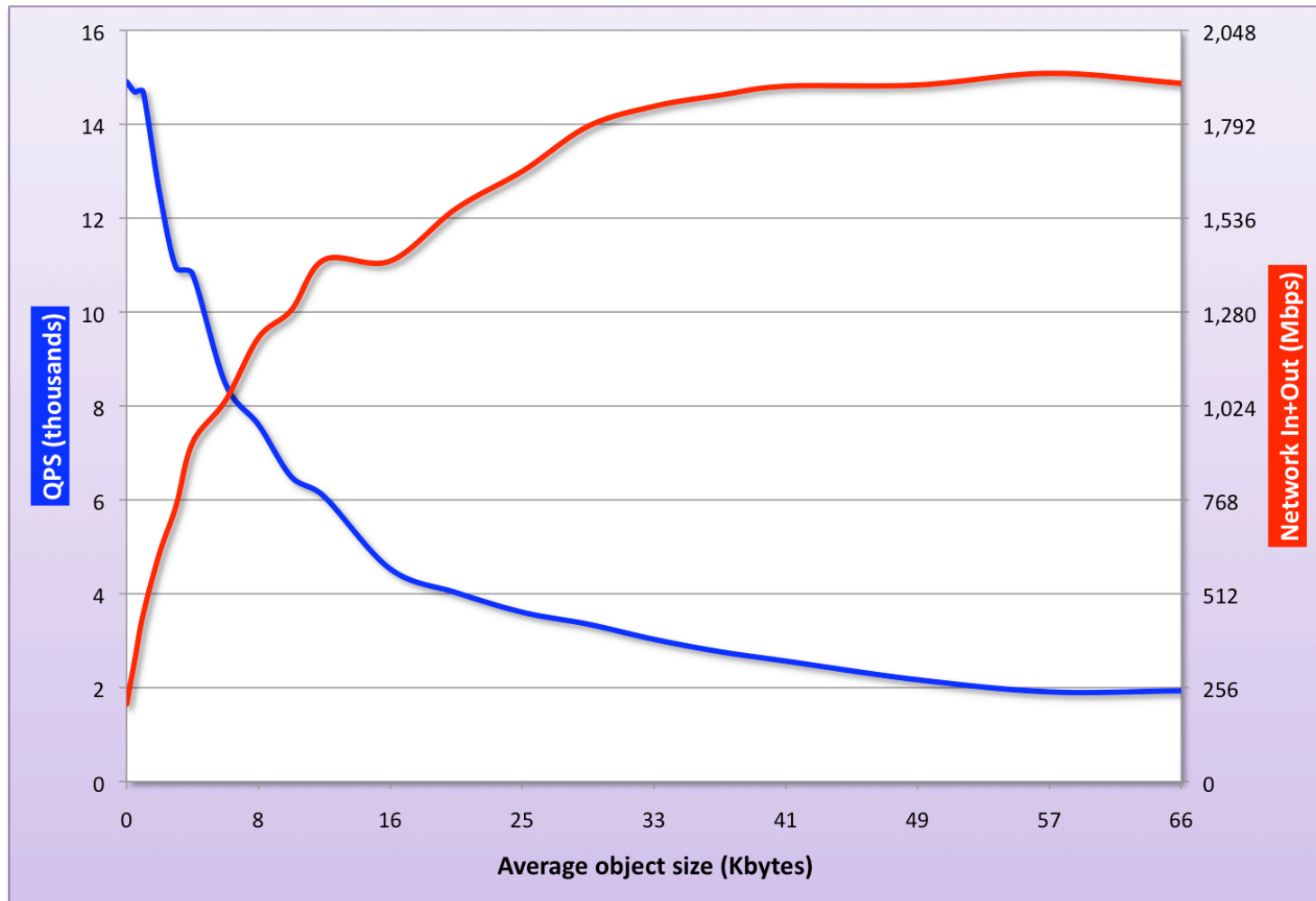


It's fast

- Quad core 1.86GHz processor: 35K rps
 - 95% cache hit rate
 - 1,000 client connections
 - 1KB response from the origin
 - 4 Keep-alive requests per connection
 - 10,000 unique objects
- Up to 3.6 Gbits/sec per server
- Seen 50+ Gbits/sec in production – 400 Terabytes a day
- 500K+ rps in production
- Tested with 100K connections, 40K active
 - Idle connections are cheap (CPU wise)



Throughput (single server)



Use Cases at Yahoo!

- Static content (CDN)
- Connection management and routing
- Layer 7 routing

History

- Code base came from Inktomi acquisition
- Commercial history
 - Launched in 1997
 - Sold access to source code + binaries
 - used by many companies including AOL, Microsoft
 - Retired ~2003
- Restarted development 2005
- Well documented SDK and Administration guide
 - Caveat: Lots of documentation clean up needed to remove outdated references
- Lots of features and code



Lots of Work to Get to Open Source Release

- Coverity scan and 2500+ issues resolved
- Internal tools for code scans
- grep for potential leaks of information
- Patent review and analyzes of what we might be giving up
- Outside company copyright scan
- Copyright and license issues
 - removing code and proper license notifications
- Removing features we can't or didn't want to open source
 - SNMP, authentication, streaming, NTTP, FTP, internal features
- The Apache process
- OSON 2009 BOF (Bryan Call and Leif Hedstrom)



Why did we open source?

- Great experience with Hadoop
- HTTP Server may have natural symbiotic relationship
- With past selling of source code, have heard of external interest in code donation
- Traffic Server used extensively internally
 - Continually finding new use cases
 - Have dedicated team working on improvements
 - Want to work with community to accelerate development



The Architecture

- Multi-threaded asynchronous state machine
 - Separate accept threads per listening port
 - Normally 2.5 worker threads per core
 - Additional helper threads for logs and stats
 - State machine per active request
- Plugins support
 - Able to hook plugins at different stages of the state machine
 - Ability to support other protocols
 - NNTP, streaming, FTP (but not open sourced)



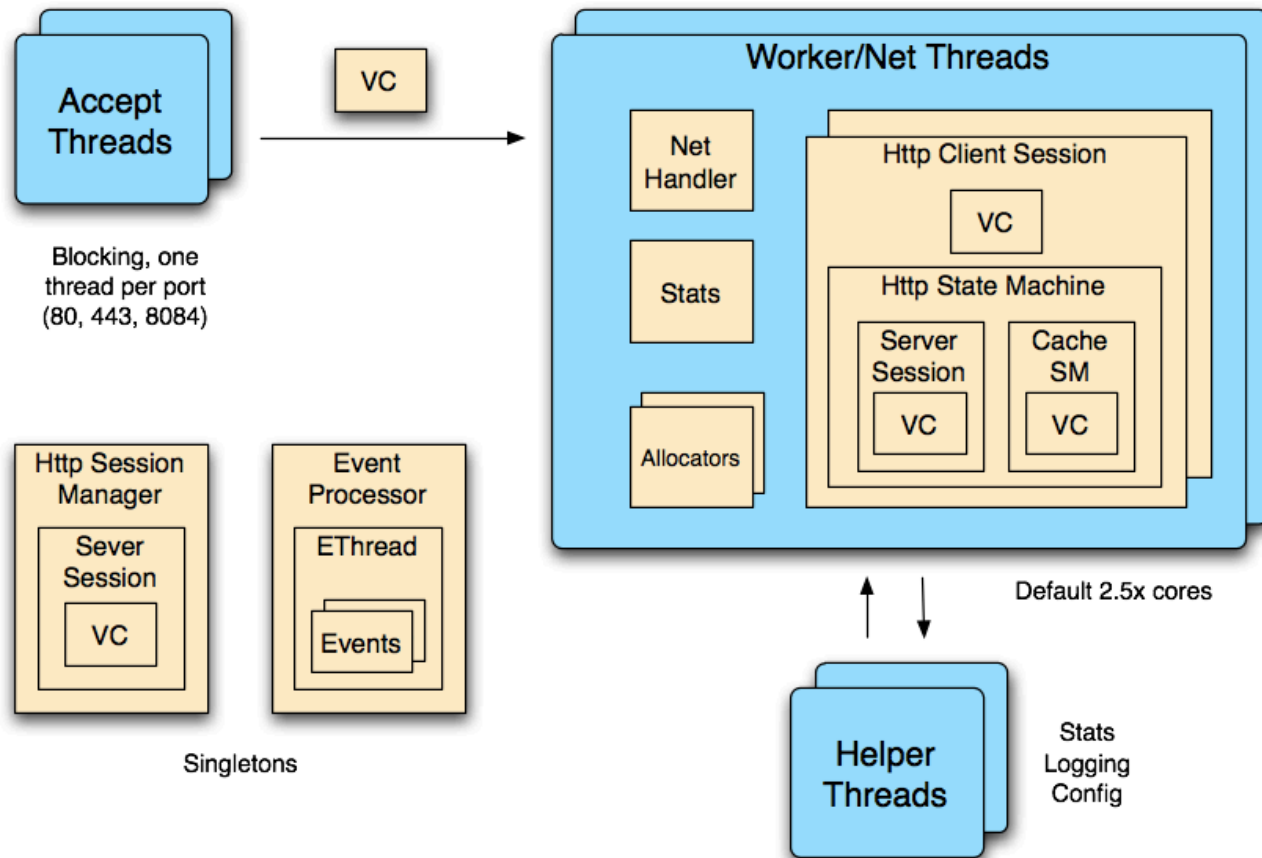
The Architecture - Definitions

- Continuation
 - Subclassed to create event-driven state machines. Continuations are heavily used in the code.
- Action
 - An operation on a Processor.
 - An Event is a subclassed Action that is used by the EventProcessor
- Processor
 - Used to schedule work.
- Virtual Connection (VC)
 - Uni or bi-directional communication
 - UnixNetVConnection represent a TCP network connection
- VIO
 - Description of a IO operation. Keeps track of how much work has been done. Used to reenale IO.
- IOBuffer

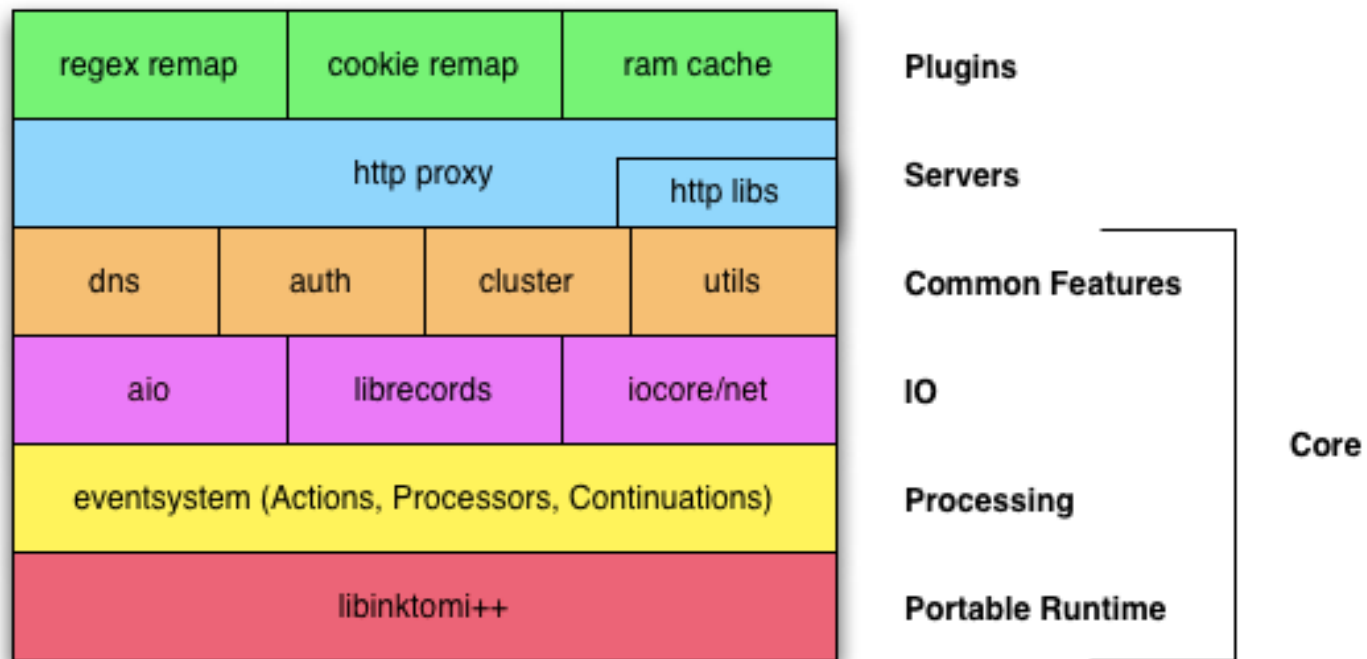


The Architecture

Traffic Server Overview



The Architecture



Future Plans

- IPv6 support
- 64-bit
- Testing harness for functional testing (in progress)
- Event system – run jobs on separate threads (for sync)
- Cache API redesign
 - Cache chaining
 - Disk cache in separate thread
- HTTP State Machine redesign
- ESI
- Upload Proxy
- Stale-while-revalidate
- COMET support (pushing data to the client)



How To Help

- Code
- Testing
- Feature and design ideas
- Contact information
 - trafficserver-user-subscribe@incubator.apache.org
 - trafficserver-dev-subscribe@incubator.apache.org
 - #traffic-server on irc.freenode.net
 - <http://cwiki.apache.org/confluence/display/TS/Traffic+Server>

