OpenAFS Out-of-Band TCP

Andrew Deason
Sine Nomine Associates

European AFS and Kerberos Conference 2012

Agenda

SINE NOMINE ASSOCIATES

- Why is AFS so slow?
- Project Background
- OOB Design
- Current Status (numbers!)
- Future Directions

Why is AFS so slow?

SINE NOMINE ASSOCIATES

- Define “performance” / “slow”
- AFS-specific factors (cache, CBs, etc)
- Inherent UDP restrictions
  - Firewalls, checksum offloading, etc

Why is AFS so slow?

SINE NOMINE ASSOCIATES

- Rx implementation and protocol
  - See Simon's talk(s)
- Rx window size
  - (32 * 1400) / RTT
  - 1ms RTT: ~43 MiB/s
  - 10ms RTT: ~4 MiB/s

Project Background

SINE NOMINE ASSOCIATES

- AFS too slow for customer
  - Need fix quickly
- Declined approaches:
  - RxOSD vicep-access
  - RxOSD non-vicep-access
  - RxTCP
  - RxUDP improvements

Project Background

SINE NOMINE ASSOCIATES

- Compromise on TCP OOB
  - Rx handles args, aborts, auth, etc
  - No long-lived TCP conns
  - Tie TCP conn to Rx call
- Rapid development
- First pass not public

Project Background

SINE NOMINE ASSOCIATES

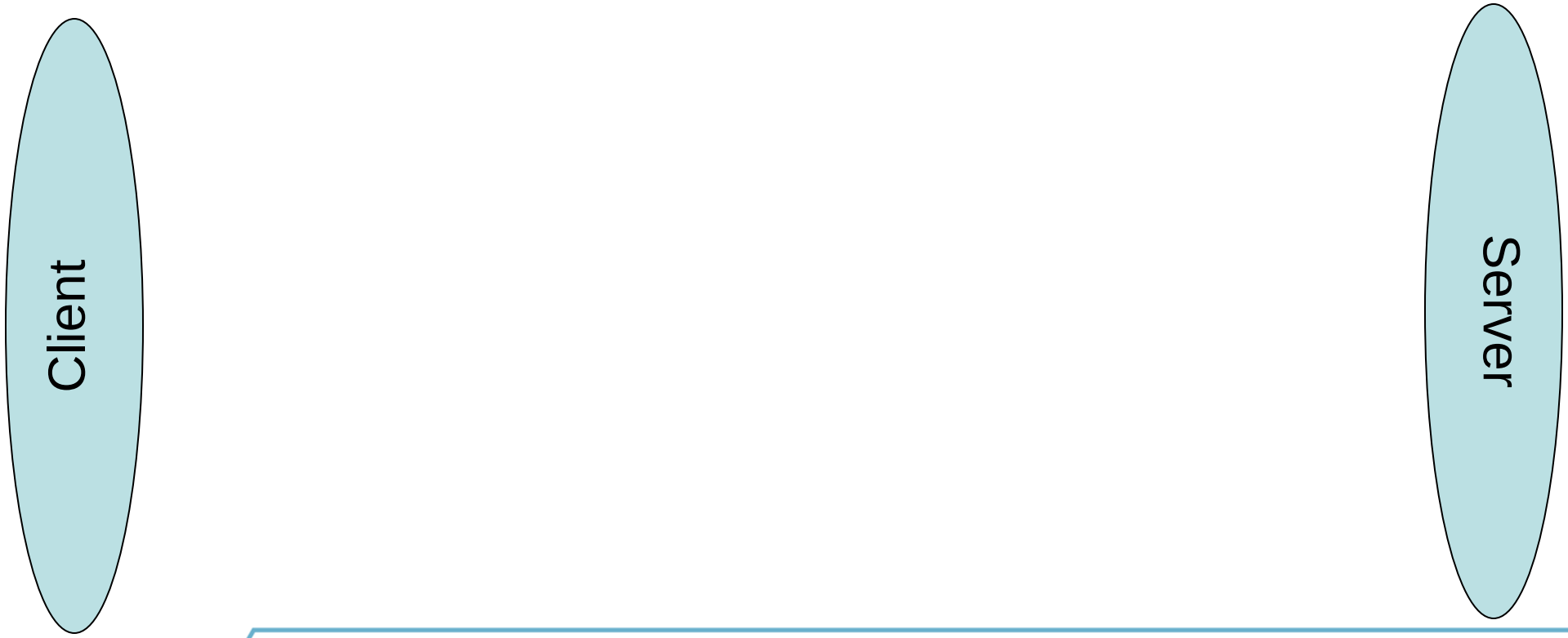
- Started in August/September 2011
- 1.4 client/server delivered in October
- 1.6 client in February
- Production deployment in March/April
- 1.6 server in May

OOB Design

- Designed for rapid dev
- FTP-like control/data channels
- Very similar to existing FetchData64
- Not just for TCP

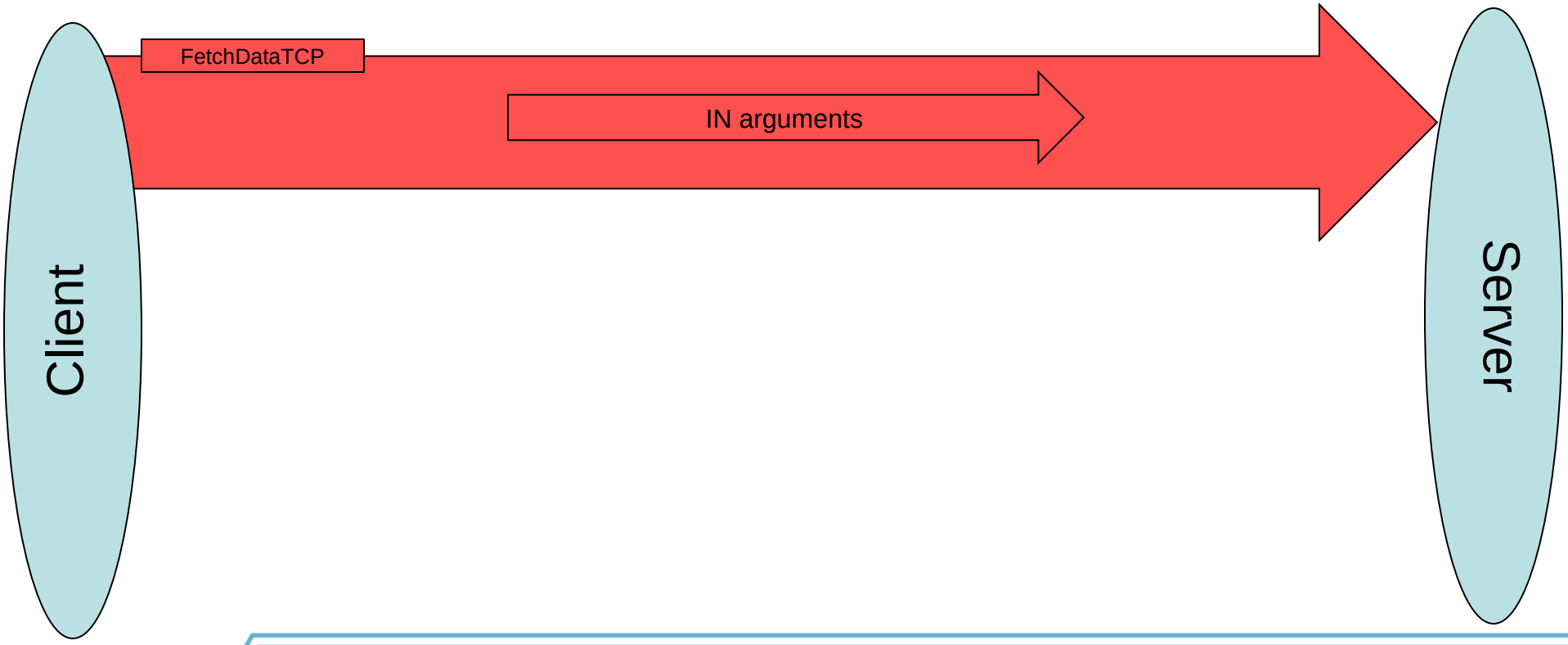
OOB Design (protocol)

Say a client wants to fetch a file…



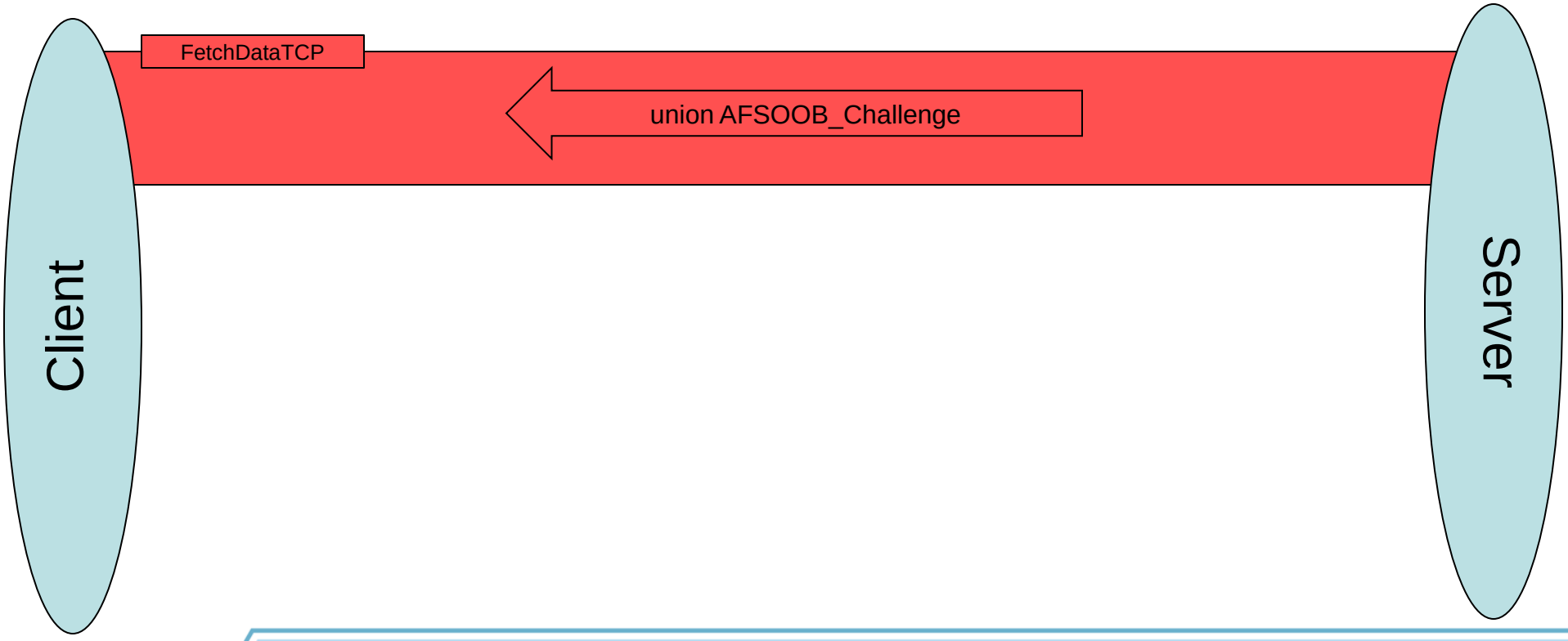
OOB Design (protocol)

Client starts split FetchDataTCP call



OOB Design (protocol)

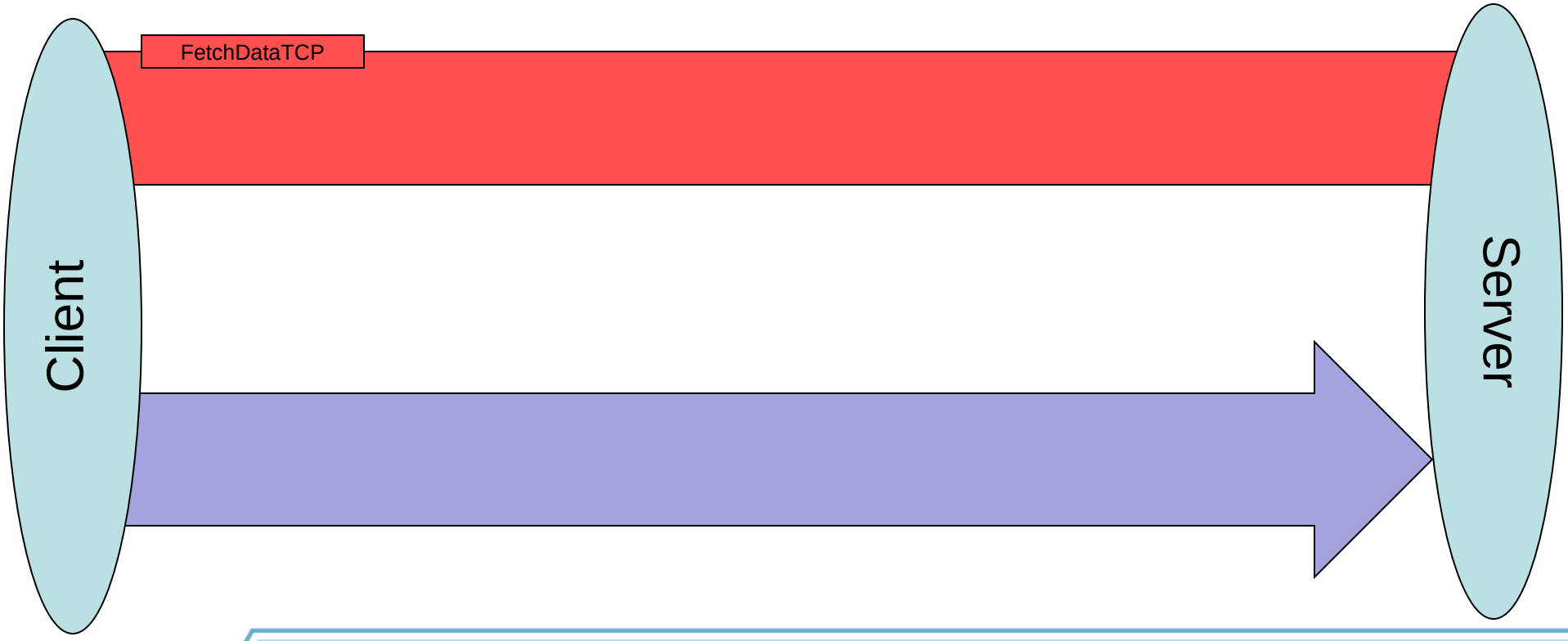
Server sends TCP information



OOB Design (protocol)

SINE NOMINE ASSOCIATES

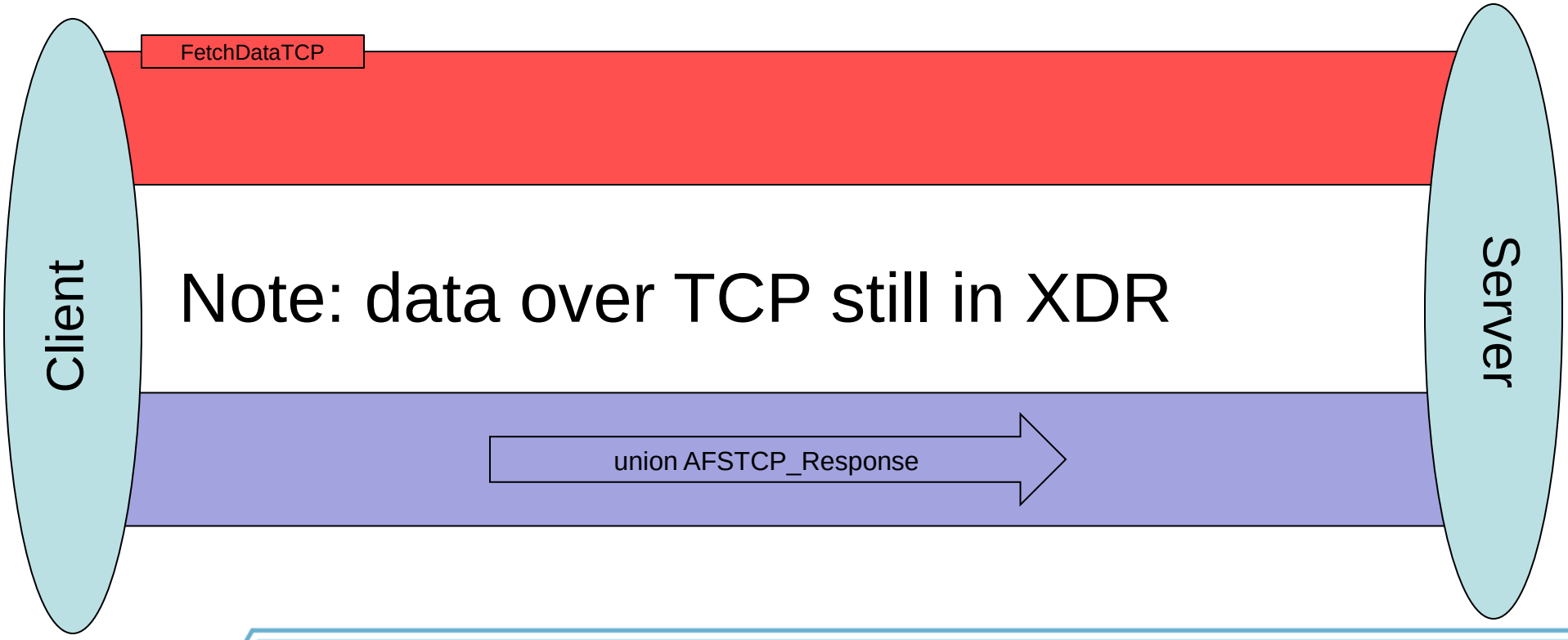
Client creates TCP connection



OOB Design (protocol)

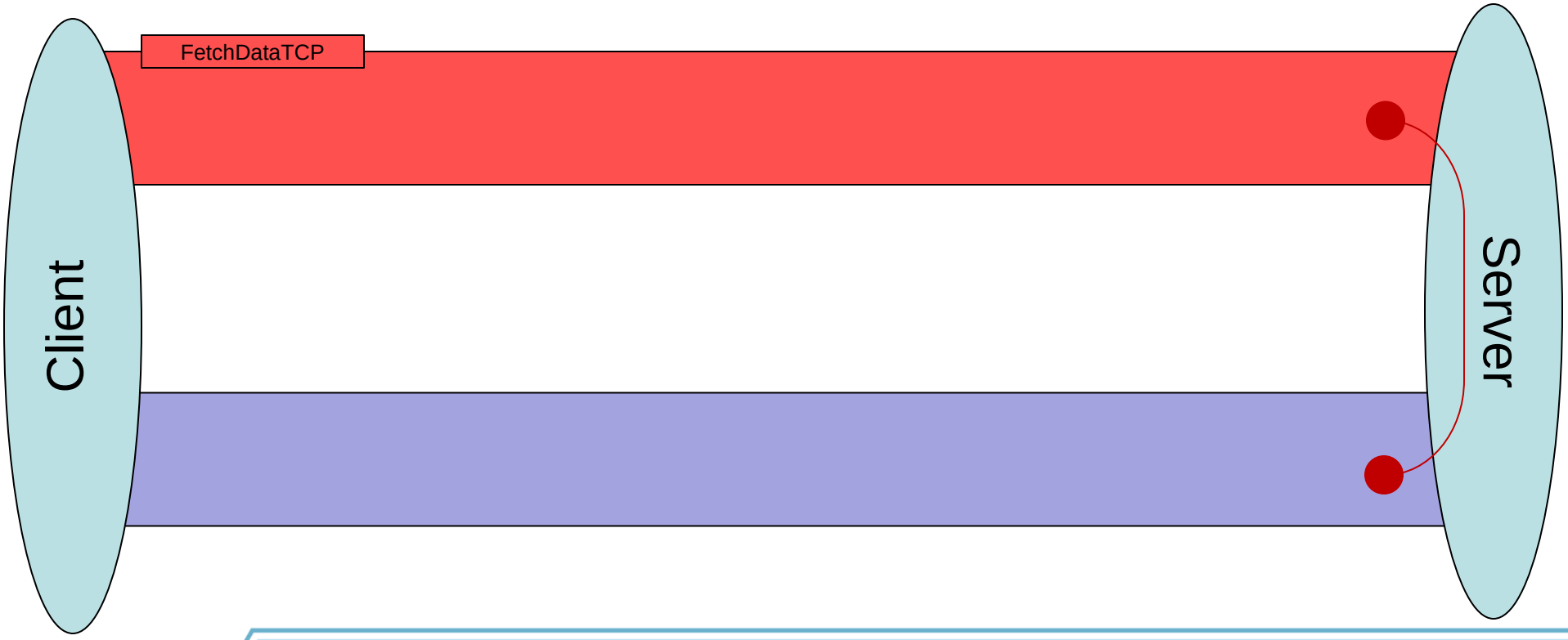
SINE NOMINE ASSOCIATES

Client send conn metadata (IDs Rx call)



OOB Design (protocol)

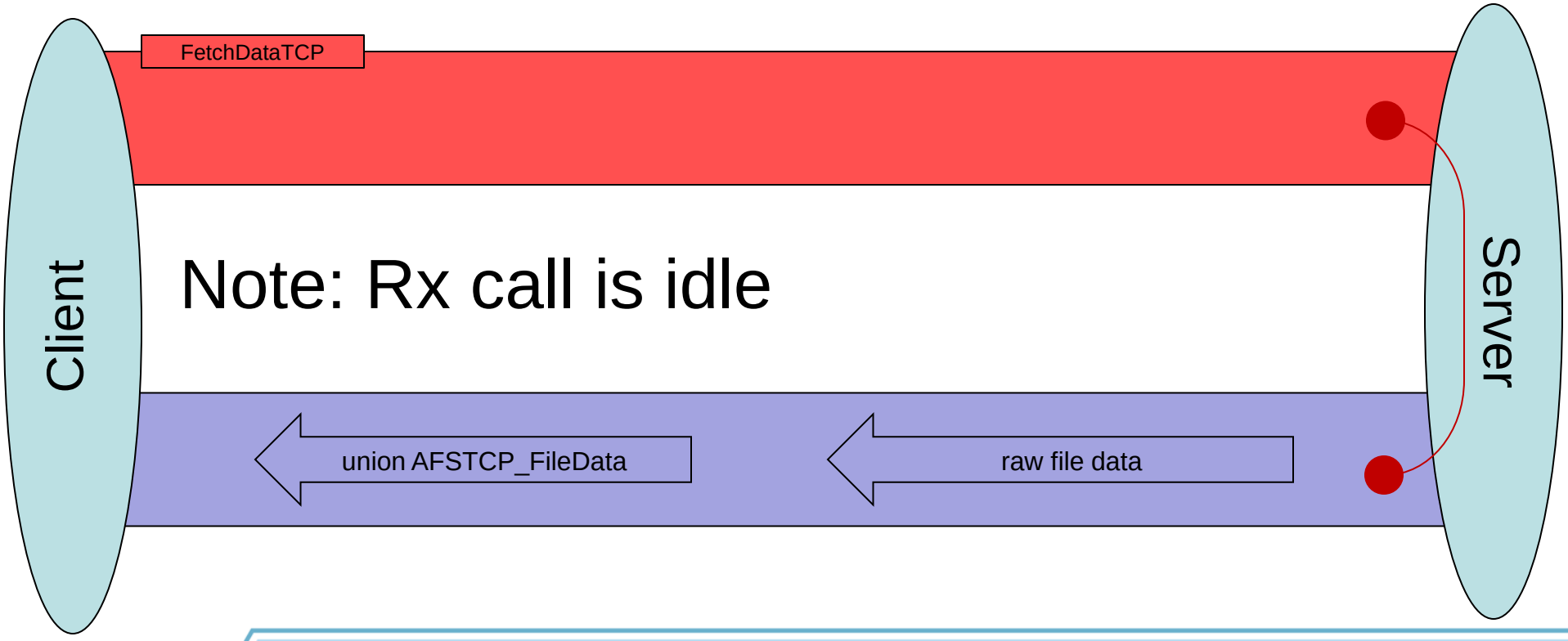
Server associates connection



OOB Design (protocol)

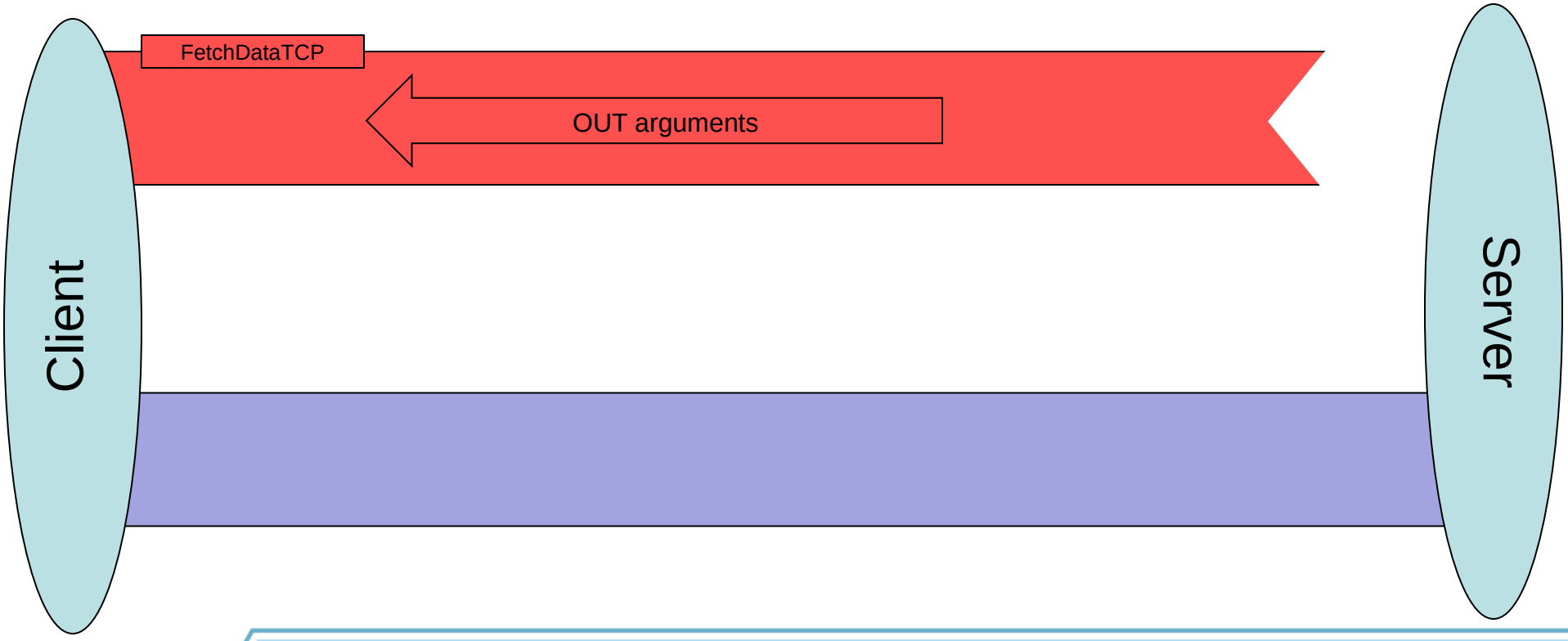
SINE NOMINE ASSOCIATES

Server sends file data



OOB Design (protocol)

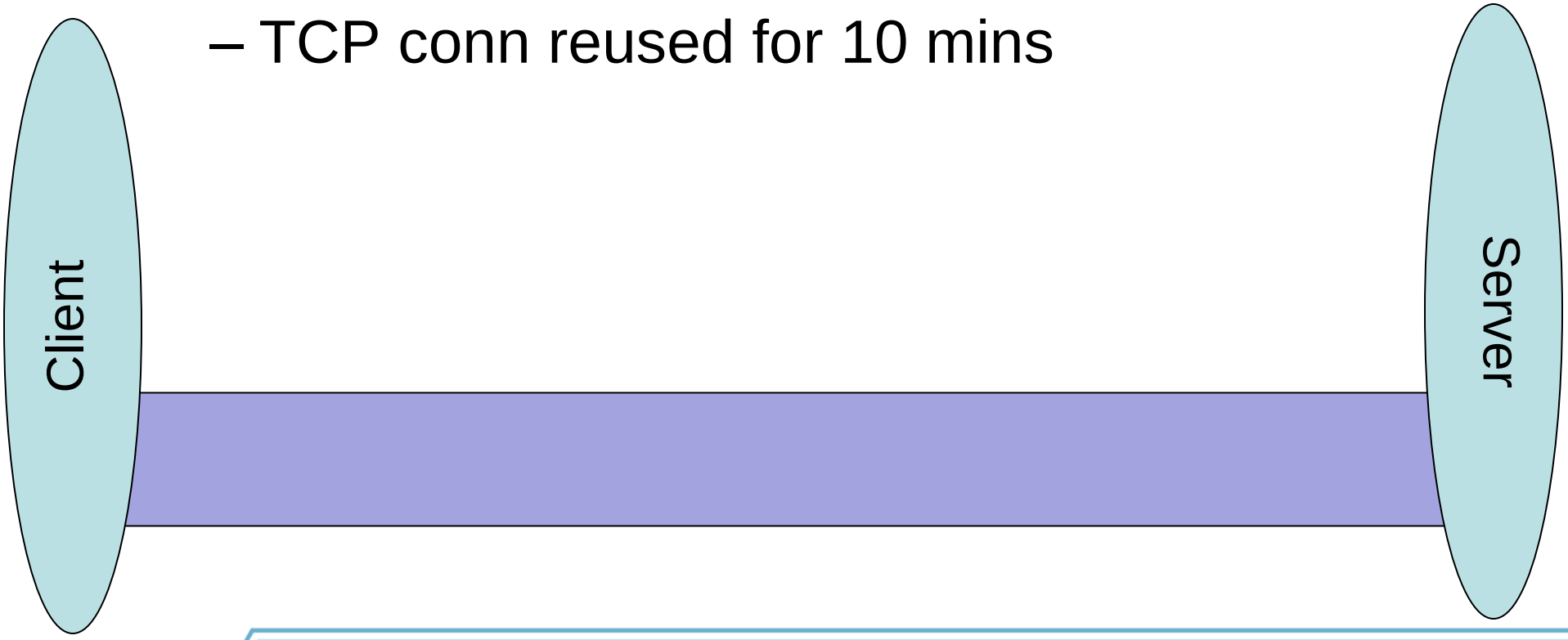
Server ends FetchDataTCP call



OOB Design (protocol)

Transfer is complete

- TCP conn reused for 10 mins



OOB Design (client)

SINE NOMINE ASSOCIATES

- Cache-bypass-like threshold
  - sysctl afs.oob_tcp_thresh
- RXGEN_OPCODE server detection
- Parameters tweakable via sysctl

OOB Design (server)

- libevent
- Async connection receipt
- TCP conns handed to Rx thread
- TCP conn always after Rx call

OOB Design (limitations)

SINE NOMINE ASSOCIATES

- Extra round-trip
- Not extensible well
- Client code organization

Implementation Details

SINE NOMINE ASSOCIATES

- rx_FlushWriteNotLast
- rxkad_Encrypt
- rx_SetErrorProc
- Linux configurable readahead
- osi_BlockSignals

Current Status

SINE NOMINE ASSOCIATES

- 1.4, 1.6, client, server
- Linux now, but portable
- Cache bypass
- Zero-copy fetch
- Non-standard protocol published
- Source available on request

Performance

Performance

SINE NOMINE ASSOCIATES

- 10GigE
  - Write: 300s MiB/s (~3gbps) memcache
  - Read: 500s MiB/s (~4-5gbps) memcache
  - Read: 700s MiB/s (~6gbps) bypass
- afscp even higher with high chunksize
- Same results when capped at 7gbps

Performance

SINE NOMINE ASSOCIATES

- Major limiting factors:
  - Chunksize (or bypass readahead size)
  - Cache overhead
  - Extra RTT
- Benchmarks are “remote”

Future Directions

SINE NOMINE ASSOCIATES

- Standardize new OOB protocol
  - Extra RTT, UUIDs, cap bit, ext-union
- Platform support
- Volserver OOB

SINE NOMINE
ASSOCIATES

Questions?

SINE NOMINE ASSOCIATES

Thanks!

Andrew Deason
Sine Nomine Associates
adeason@sinenomine.net