
whamcloud

whamcloud

Improvements for Traversing Large Directory

- Fan Yong
  yong.fan@whamcloud.com

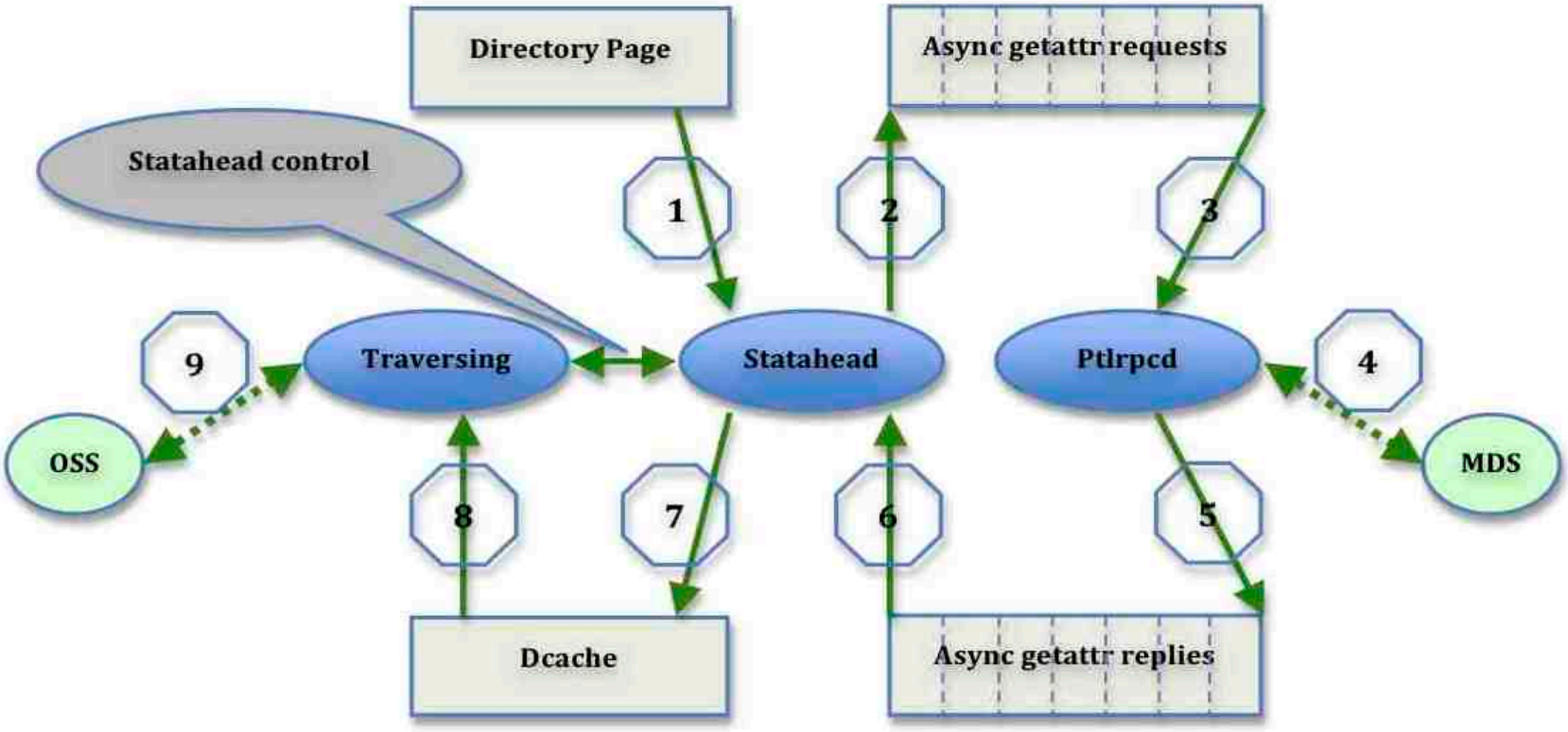
Overview

Focus on sequential traversing large directory on single client: ls/du/find

- Current statahead
- Improvements
  - Asynchronous glimpse lock
  - Statahead load balance
  - Aggregate attributes to save RPCs
  - Efficient ldlm/object hash
- Test results
- Further works

Existing statahead

- Only pre-fetch attributes from MDS
  - Single pipeline for async getattr



Asynchronous glimpse lock

- Prefetch glimpse lock from OST
  - Async RPC without waiting
  - Cache file size/blocks if AGL granted
  - No glimpse callback if OST can't grant AGL
  - Re-enqueued by traversing thread if AGL non-granted/cancelled

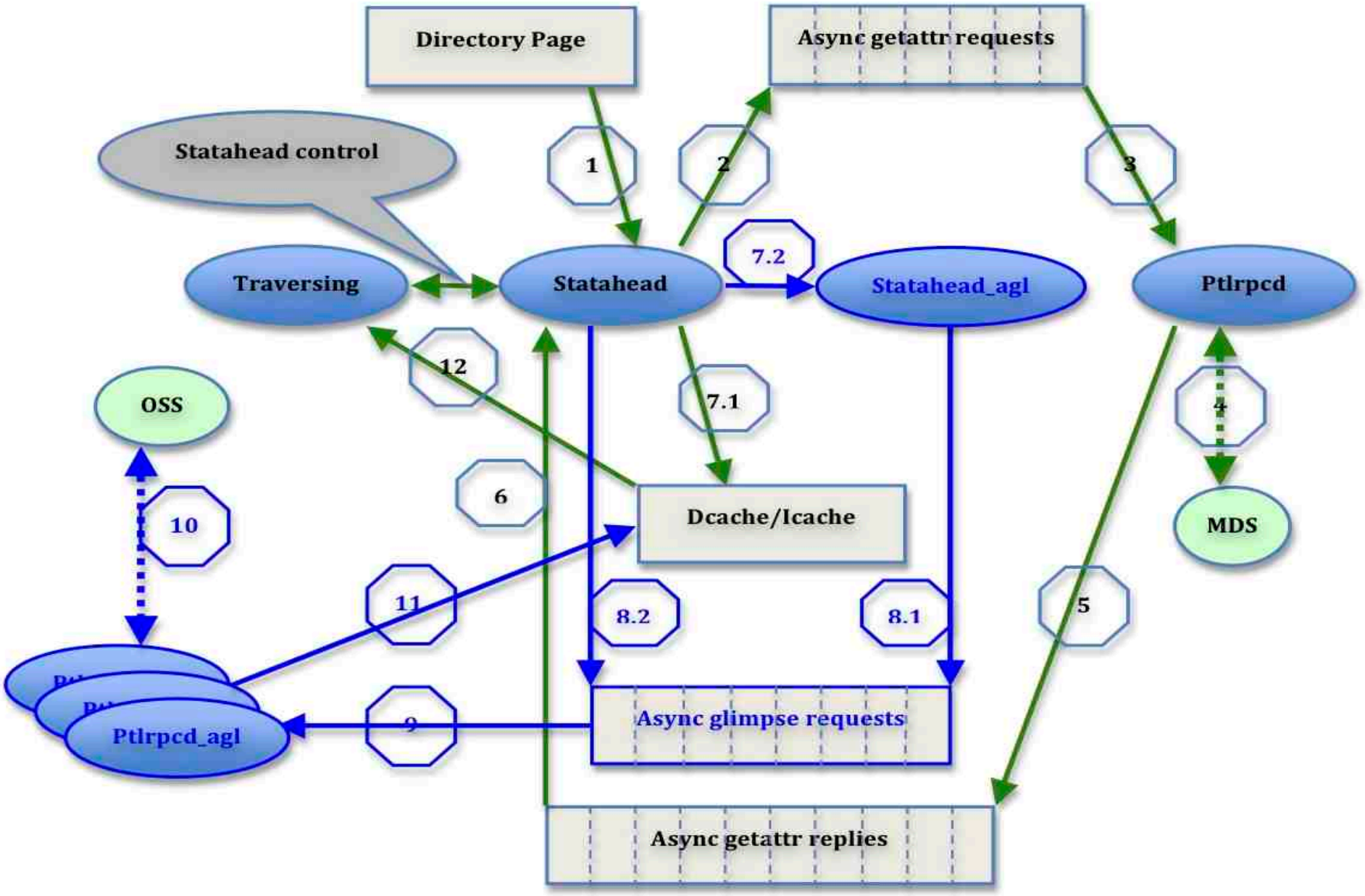
May cause additional RPCs if someone held conflicting locks on some items, but it works well for large directory, since stat/read cases are more common than writes.

Statahead load balance (1/2)

- Independent portal RPC services (ptlrpcd_agl)
  - Dedicated to processing AGL RPC
    - Single ptlrpcd on client may be overloaded to process all non-I/O async RPCs when traversing large directory
    - Unbalanced load, other CPUs maybe idle
  - Separate pipelines for async getattr and AGL, decrease unnecessary implied dependence between MDS-side RPC and OSS-side RPC
    - Async RPCs for getattr and glimpse can be processed in parallel (for different files), no contention between ptlrpcd and ptlrpcd_agl
  - Multiple ptlrpcd_agl services can scatter AGL loads among more CPUs, and more efficient for multiple-striped cases.

Statahead load balance (2/2)

- Unbalanced load between statahead and traversing
  - Statahead thread
    - Async getattr pipeline Input & Output
    - AGL pipeline Input
  - Traversing thread
    - AGL pipeline Output
  - AGL is more time-consuming than async getattr for multiple-striped cases. Maybe cause statahead slow down as to RPC processing time can't be hidden.
- Independent LLITE level statahead_agl thread
  - Statahead_agl thread
    - AGL pipeline Input
  - Statahead thread
    - Async getattr pipeline Input & Output
    - Help to process AGL pipeline Input when idle
  - Traversing thread
    - AGL pipeline Output


Directory Page
Async getattr requests
Statahead control
1
2
3
Traversing
Statahead
7.2
Statahead_agl
Ptlrpcd
12
7.1
4
OSS
6
Dcache/Icache
MDS
10
11
8.2
8.1
5
Ptlrpcd_agl
9
Async glimpse requests
Async getattr replies

Aggregate attributes to save RPCs

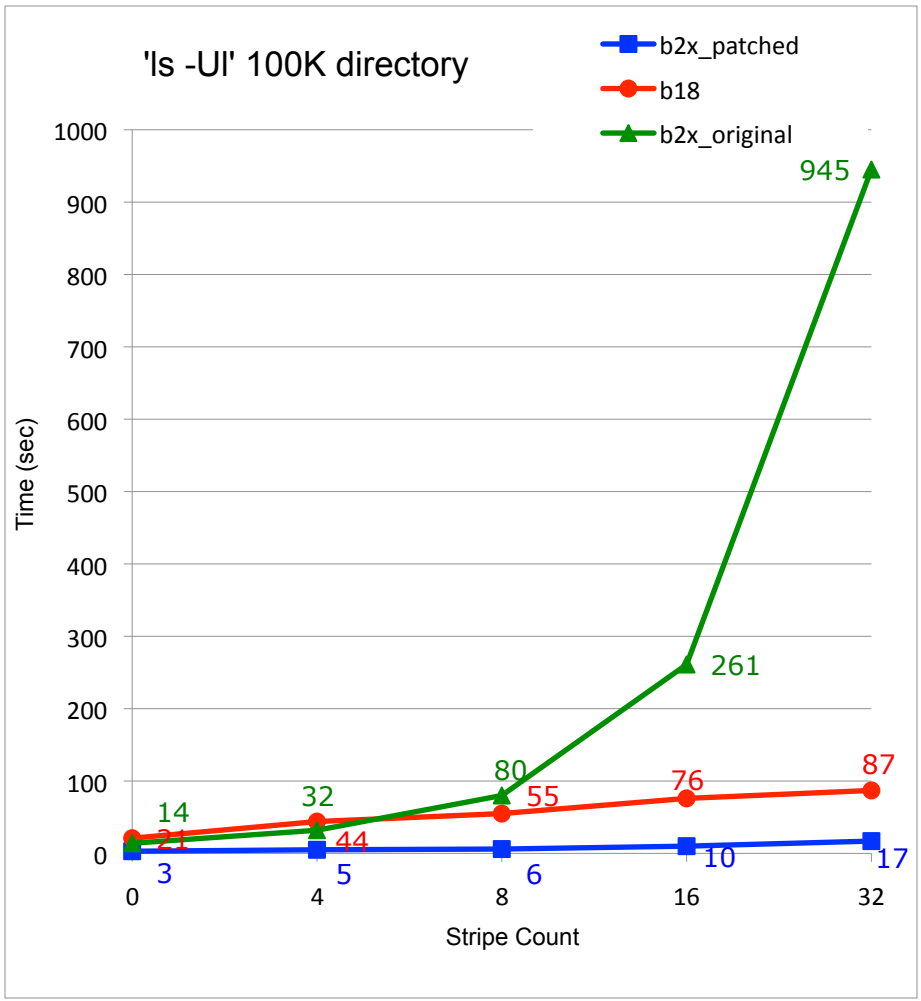
- MDS-side attributes for traversing directory:
  1) Basic attributes, like mode/owner/flags
  2) Stripe information
  3) Access Control List
  4) Default ACL for directory

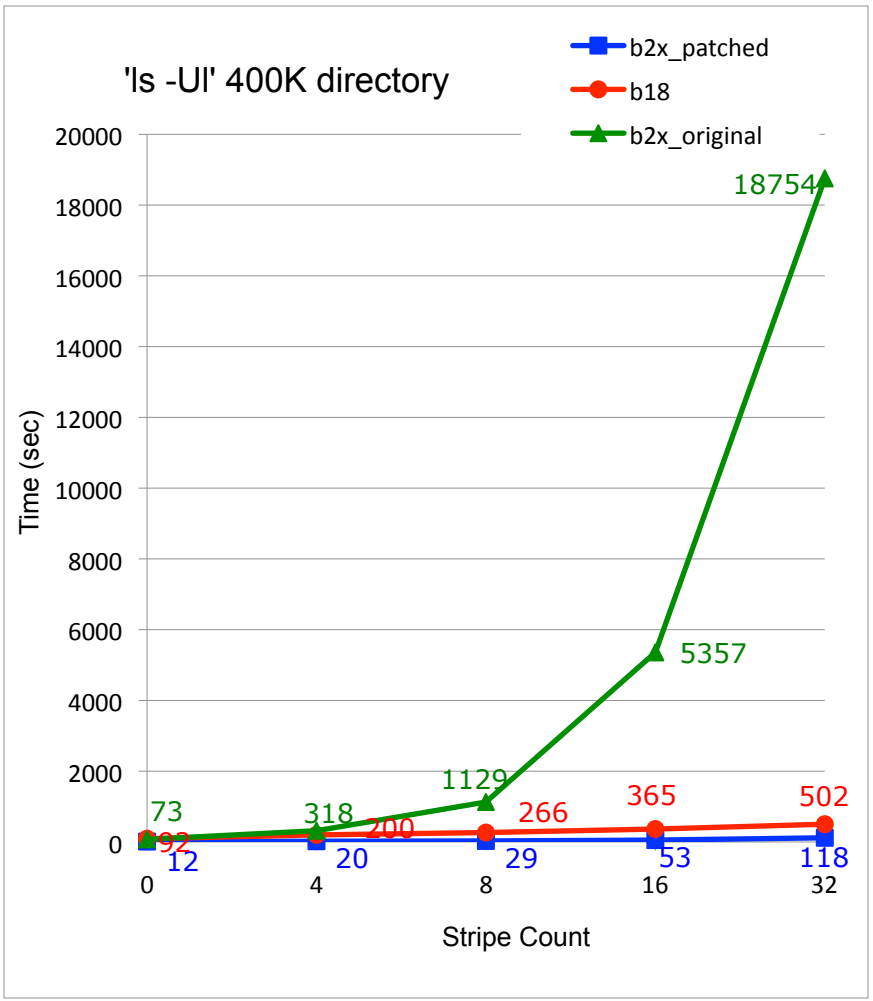
  Currently, 1) & 2) & 3) can be obtained through single RPC; and further combining 4) can save about 50% RPC for the best cases.
  - Maybe more in future

Efficient ldlm/object hash

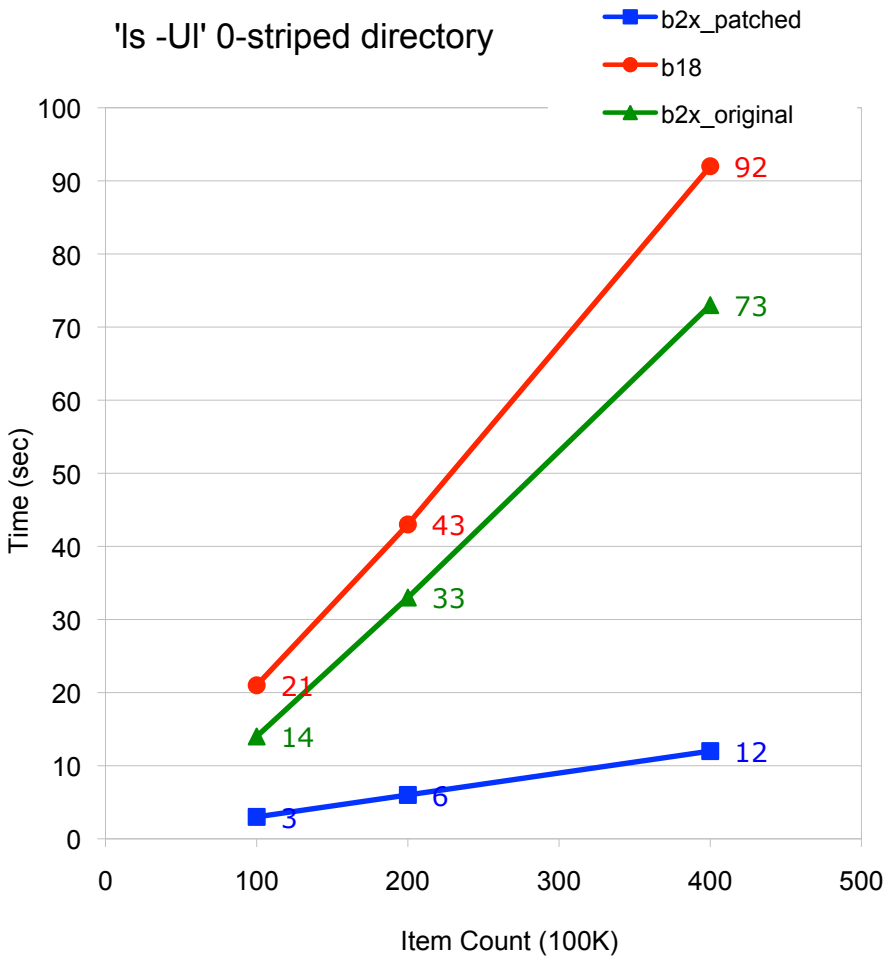
- Hash greatly affects traversing performance
  - MDS: object hash & ldlm hash
  - OSS: ldlm hash
  - Client: object hash & ldlm hash
- Hash bucket depth is quite important
  - Basically depends on hash function
    - Scatter ldlms/objects throughout buckets as evenly as possible
  - RAM size and filesystem size are associated
  - Experience: Max depth for millions of ldlms/objects hash
    - lustre-2.0 shows nearly 3K
    - no more than 50 is expected

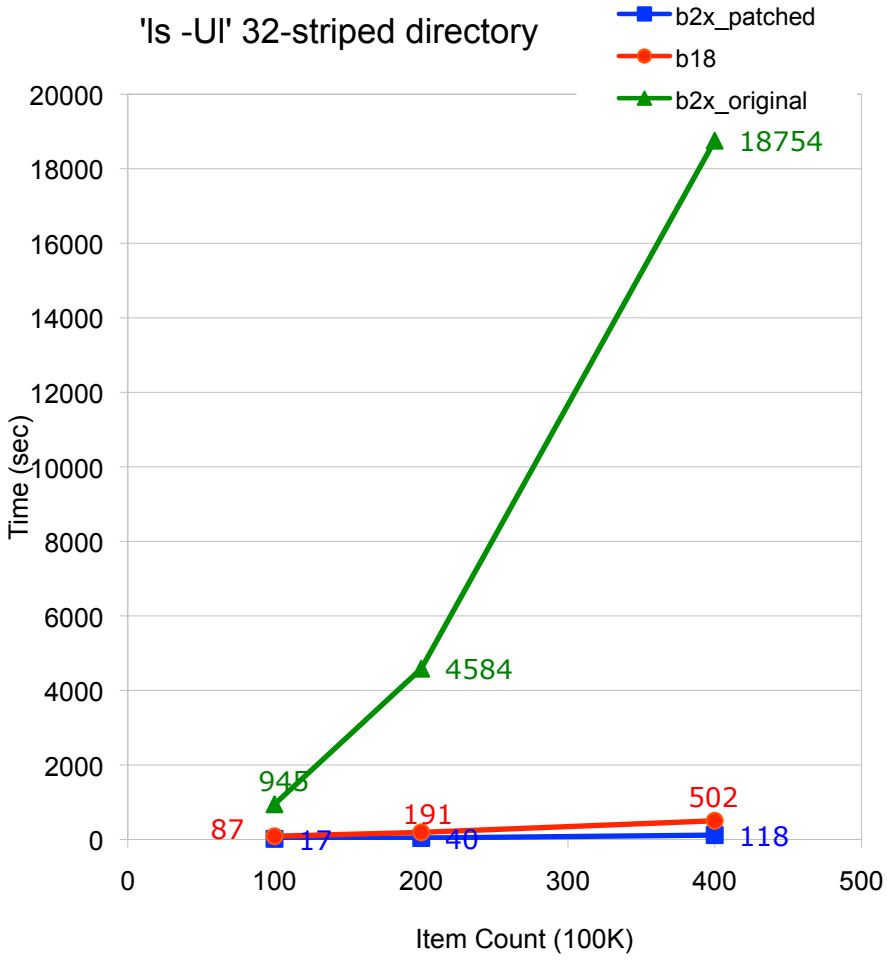
'ls –Ul' on single client (1/2)





‘ls –Ul’ on single client (2/2)





Further works

- Ptlrpcd threads pool
  - Common used for kinds of async operations: statahead, AGL, IO
- Aggregate RPC
  - Same type requests in single RPC: getattr/glimpse
  - Similar as readdir+ (mainly) for high latency network
  - Getattr-by-Fid & extent IBITS lock
- Size-on-MDS
  - Global eviction

Thank You

- Fan Yong

  yong.fan@whamcloud.com