
whamcloud

whamcloud

LUG 2012
April 25th 2012

MDS-Survey

Simulating Large Clients Loads on MDS

- Oleg Drokin

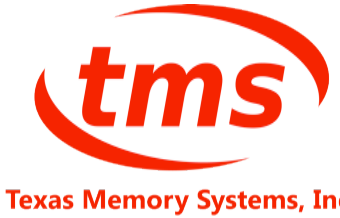
  Senior Software Engineer
  Whamcloud, Inc.

Agenda

- Acknowledgements
- How It All Started
- Analysis
- Limitations
- Conclusions

Acknowledgments

- This presentation was produced in collaboration with
  - Minh Diep, Richard Henwood, and Wang Di of Whamcloud
  - John Hammond of TACC
  - James Simmons of ORNL
  - Texas Memory Systems

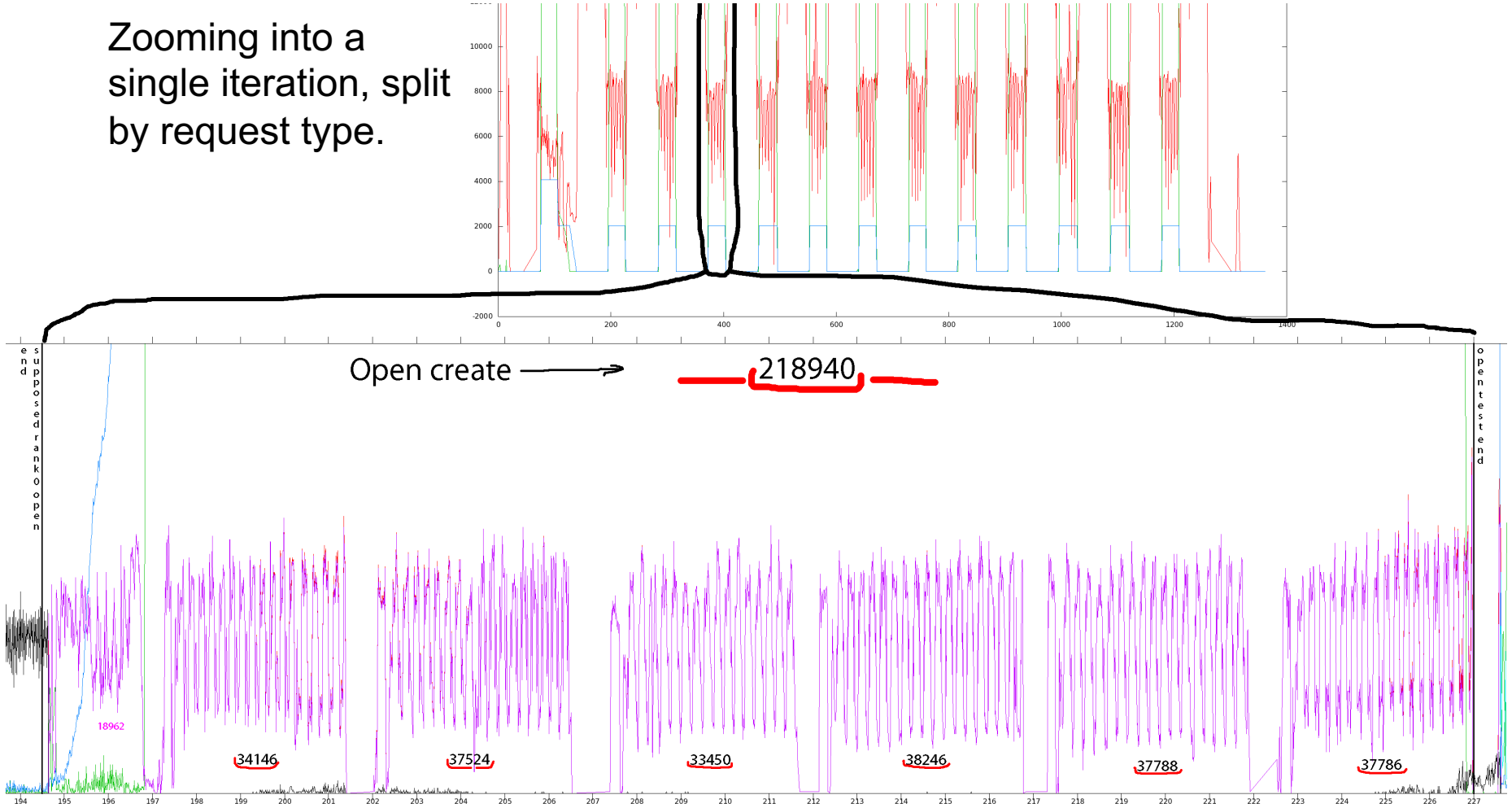


How it all started

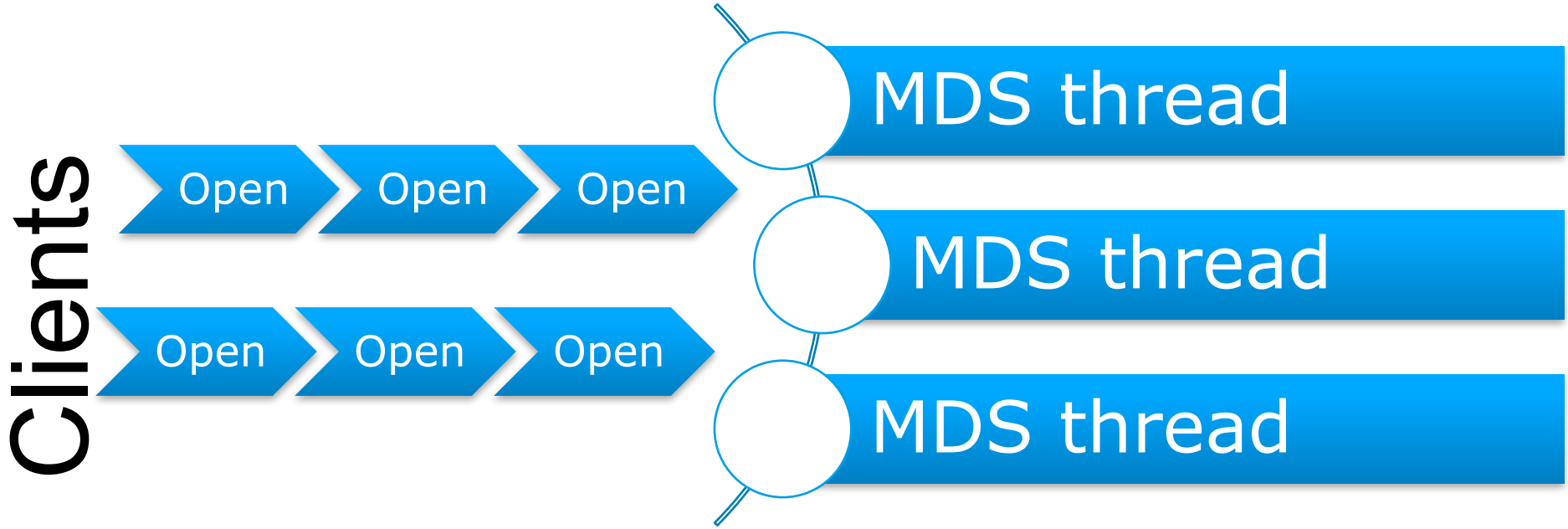
224k threads from 18k clients creating 1 file in a shared dir each. JaguarPF

How it all started (cont'd)

Zooming into a single iteration, split by request type.



What the MDS was really doing



Current implementation

- Executed threads are in context of process requesting ioctl
  - This allows for easier control on number of threads
- For simplification no actual requests are generated, ioctl plugs straight into mdd layer
  - Only mds-local testing possible as a result
- Same familiar echo_client interface as with obdfilter testing

Supported operations

- Each thread could perform one of the following operations in a loop (sequential numbered filenames):
  - Create a file or a directory
  - Lookup a filename
  - Unlink/delete a file/directory
- For creates you have a choice of striping, or no striping
- All operations happen in the Lustre namespace

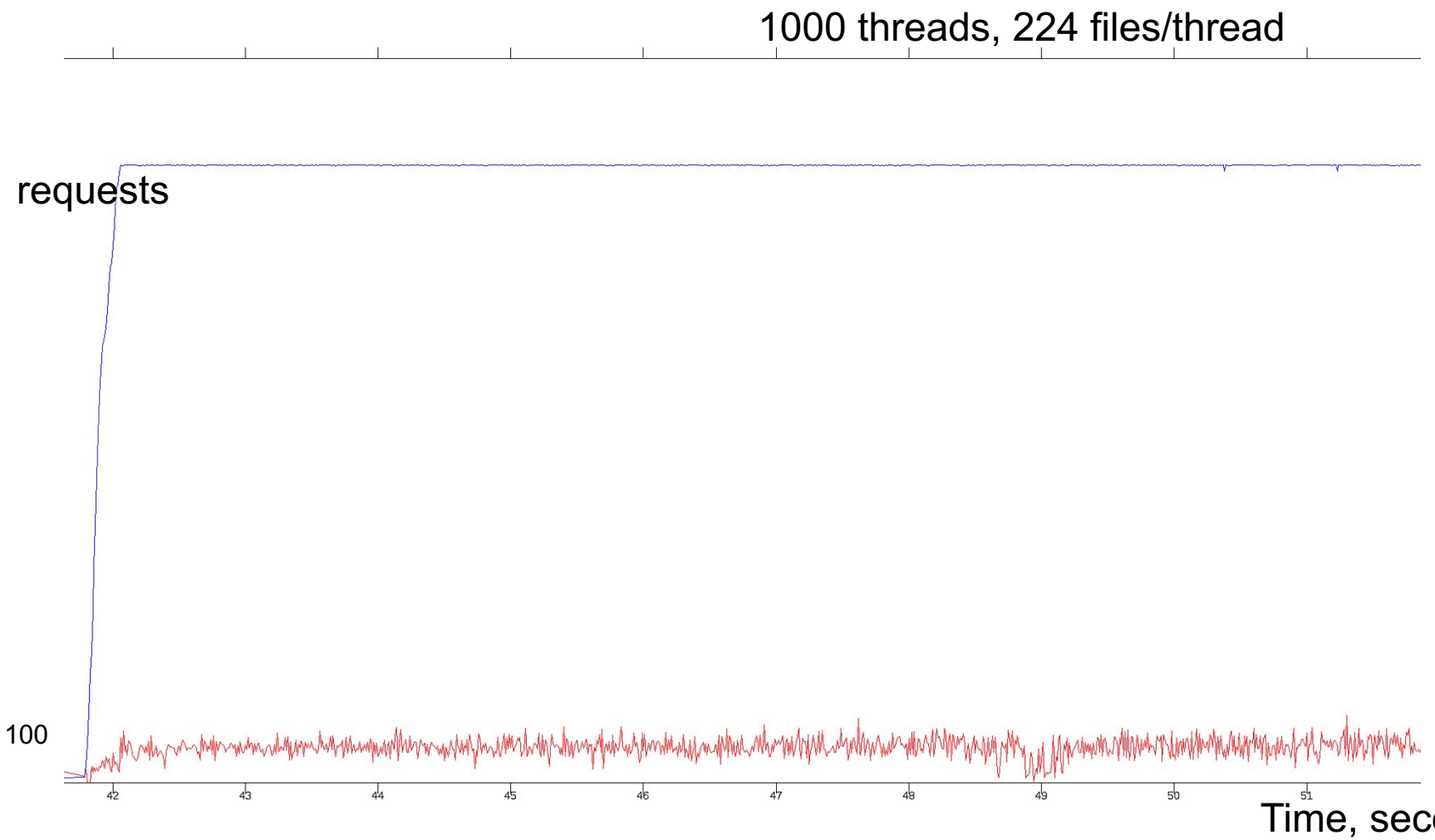
Example lctl script

```
#setup
attach echo_client ecc-MDT0000 ecc-MDT0000_UUID
setup lustre-MDT0000 mdd
attach echo_client ecc-MDT0001 ecc-MDT0001_UUID
setup lustre-MDT0000 mdd

cfg_device 7
test_mkdir /tests
test_mkdir /tests1
test_mkdir /tests2
test_mkdir /tests3
--threads 4 -1 7 test_create -d /tests -b 2 -n 1000
--threads 4 -1 7 test_create -d /tests -D 4 -b 20000 -n 1000
--threads 4 -1 7 test_destroy -d /tests -b 2 -n 1000
--threads 4 -1 7 test_destroy -d /tests -D 4 -b 20000 -n 1000
```

Internal journal test graph

External journal test graph



Mds-survey script for automation

Inputs

- `thrlo` threads to start testing
- `thrhi` maximum number of threads to test
- `targets` MDT instance
- `file_count` total number of files per thread to test
- `dir_count` total number of directories to test
- `stripe_count` number stripe on OST objects
- `tests_str` test operations. i.e. "create" or “destroy“

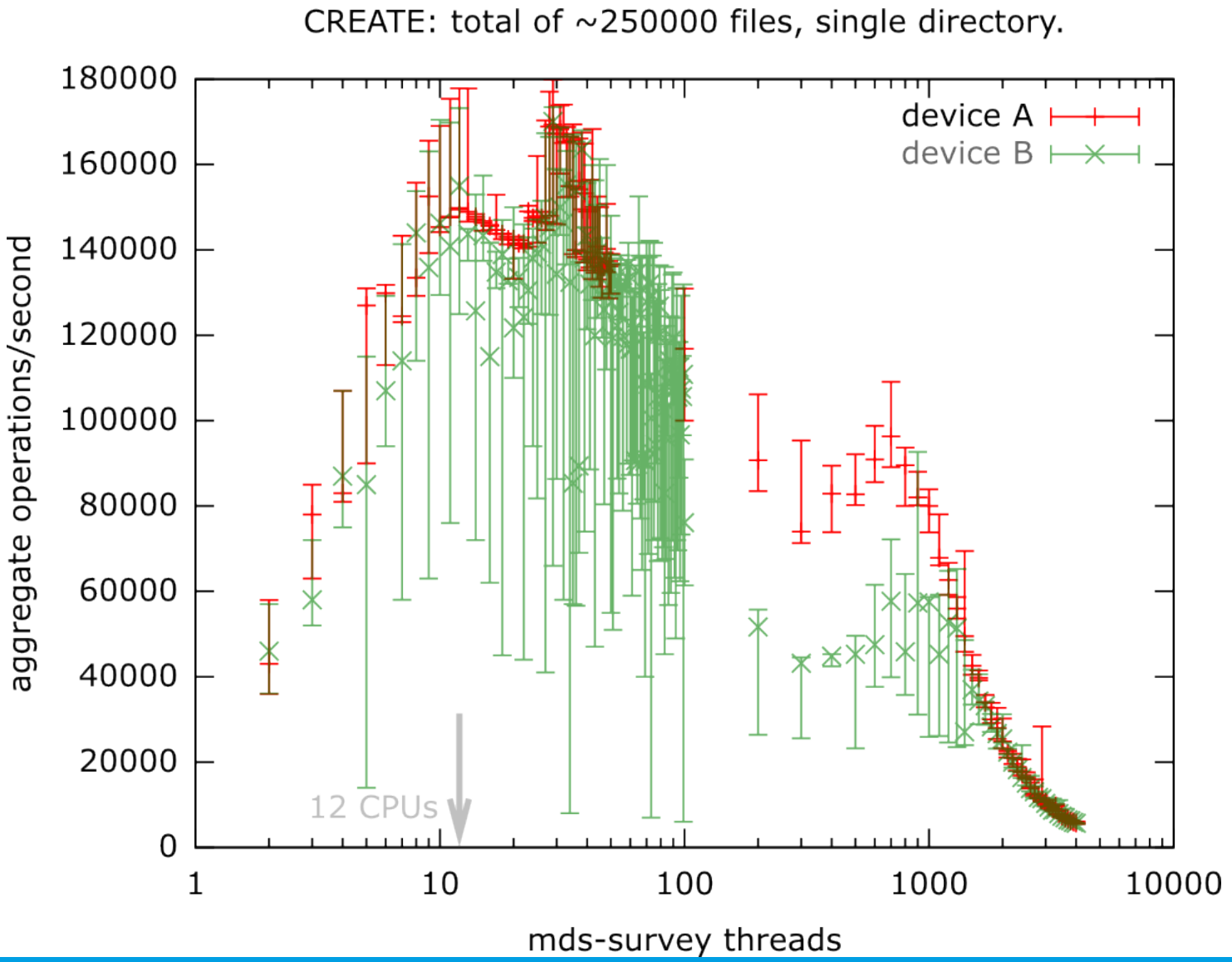
Outputs

- Time to create
- Time to lookup
- Time to md_getattr
- Time to setxattr
- Time to destroy

Eg. In a single directory, operate over twenty thousand files with two threads. Do not create objects on OSTs.

```
dir_count=1 thrlo=2 thrhi=2 file_count=10000 sh mds-survey
```

mds-survey example run: create



Limitations

- Only same-type operations could be performed in a loop
- In 2.2 release unlink type of operations does not destroy objects on OSTs
- Does not handle interruptions well, so don’t interrupt the test once started

Conclusions

- MDS-survey as landed in 2.2 adds another valuable tool to simulate several common large cluster workloads without employing the large cluster
- For smaller scale deployments supplied scripts allow for easier multi-run metadata-performance gathering
- Compare backend storage devices
- As always, a lot of room for further improvements ;)

Thank You

- Oleg Drokin

  Sr. SW Engineer

  Whamcloud, Inc.