fs - Feature #25131

mds: optimize the way how max export size is enforced

07/27/2018 01:55 AM - Zheng Yan

Status:	Resolved	**Start date:**	07/27/2018
Priority:	Normal	**Due date:**	
Assignee:	Zheng Yan	**% Done:**	0%
Category:	Performance/Resource Usage	**Estimated time:**	0.00 hour
Target version:	v14.0.0		
Source:	Development	**Affected Versions:**	
Tags:		**Component(FS):**	MDS
Backport:	mimic,luminous	**Labels (FS):**	multimds
Reviewed:		**Pull request ID:**	

Description

Current way of enforcing export size is checking export size after subtree gets frozen. It may freeze a large subtree, but only exports small portion of the subtree.

Related issues:

Related to fs - Bug #24881: unhealthy heartbeat map during subtree migration	**Duplicate**	**07/12/2018**
Related to fs - Bug #26858: mds: reset heartbeat map at potential time-consum...	**Resolved**	**08/06/2018**
Copied to fs - Backport #32098: luminous: mds: optimize the way how max expor...	**Resolved**	
Copied to fs - Backport #32100: mimic: mds: optimize the way how max export s...	**Resolved**	

History

#1 - 07/27/2018 01:57 AM - Zheng Yan

- Status changed from New to Need Review

https://github.com/ceph/ceph/pull/23088

#2 - 07/27/2018 03:26 AM - Patrick Donnelly

- Category set to Performance/Resource Usage

- Assignee set to Zheng Yan

- Target version set to v14.0.0

- Source set to Development

- Backport set to mimic,luminous

- Component(FS) MDS added

- Labels (FS) multimds added

#3 - 08/01/2018 09:02 AM - Zheng Yan

When importing a large subtree, mds can spends long time on sending cap import messages.

Importer

```
2018-08-01 16:59:13.988904 7ffff0d0e700  1 mds.1.migrator start decode_import_dir [dir 0x1000d7356c9 /testdir1
/ [2,head] rep@1,0.2 REP dir_auth=1,0 state=8196|frozentree f(v64 m2018-08-01 08:09:20.367198 255=0+255) n(v42
032 rc2018-08-01 08:09:20.367198 2560256=2560000+256) hs=71+0,ss=0+0 | child=1 frozen=1 subtree=1 importing=1
0x555560665500]
2018-08-01 16:59:22.630245 7ffff0d0e700  1 mds.1.migrator finish decode_import_dir 978929
2018-08-01 16:59:25.303462 7fffead02700  1 mds.1.migrator start finish_import_inode_caps  [dir 0x1000d7356c9 /
testdir1/ [2,head] auth{0=1,2=1,3=1,4=1,5=1} v=6017425340 cv=0/0 REP dir_auth=1,0 state=1073750020|frozentree
f(v64 m2018-08-01 08:09:20.367198 255=0+255) n(v42032 rc2018-08-01 08:09:20.367198 2560256=2560000+256) hs=167
+0,ss=0+0 | child=1 frozen=1 subtree=1 importing=1 replicated=1 0x555560665500]
2018-08-01 16:59:26.196152 7fffead02700  1 mds.1.migrator finish finish_import_inode_caps  978914
```

```
2018-08-01 16:59:26.810240 7ffff0d0e700  1 mds.1.migrator start sending cap imports [dir 0x1000d7356c9 /testdi
r1/ [2,head] auth{0=1,2=1,3=1,4=1,5=1} v=6017425340 cv=0/0 REP dir_auth=1,1 state=1073750020|frozentree f(v64
m2018-08-01 08:09:20.367198 255=0+255) n(v42032 rc2018-08-01 08:09:20.367198 2560256=2560000+256) hs=167+0,ss=
0+0 | child=1 frozen=1 subtree=1 importing=1 replicated=1 0x555560665500]
2018-08-01 16:59:38.133159 7fffedd08700  1 heartbeat_map is_healthy 'MDSRank' had timed out after 15
2018-08-01 16:59:38.133177 7fffedd08700  1 mds.beacon.b _send skipping beacon, heartbeat map not healthy
2018-08-01 16:59:40.331869 7ffff1d10700  1 heartbeat_map is_healthy 'MDSRank' had timed out after 15
2018-08-01 16:59:40.426150 7ffff0d0e700  1 mds.1.migrator finish sending cap imports 978914
2018-08-01 16:59:40.426155 7ffff0d0e700  1 heartbeat_map reset_timeout 'MDSRank' had timed out after 15
2018-08-01 16:59:40.429147 7ffff0d0e700  1 mds.1.migrator start eval [dir 0x1000d7356c9 /testdir1/ [2,head] au
th{0=1} v=6017425340 cv=0/0 REP dir_auth=1 state=1073741824 f(v64 m2018-08-01 08:09:20.367198 255=0+255) n(v42
032 rc2018-08-01 08:09:20.367198 2560256=2560000+256) hs=167+0,ss=0+0 | child=1 frozen=0 subtree=1 importing=0
 replicated=1 0x555560665500]
2018-08-01 16:59:40.933443 7ffff0d0e700  1 mds.1.migrator finish eval 978914
```

Exporter

```
2018-08-01 16:59:07.131565 7fffead02700  1 mds.0.migrator start encode_export_dir [dir 0x1000d7356c9 /testdir1
/ [2,head] auth{1=2,2=1,3=1,4=1,5=1} v=6017425340 cv=0/0 REP dir_auth=0,1 state=1074794500|frozentree|auxsubtr
ee f(v64 m2018-08-01 08:09:20.367198 255=0+255) n(v42032 rc2018-08-01 08:09:20.367198 2560256=2560000+256) hs=
167+0,ss=0+0 | child=1 frozen=1 subtree=1 replicated=1 waiter=0 authpin=0 0x55555ee4b100]
2018-08-01 16:59:11.967910 7fffead02700  1 mds.0.migrator finish encode_export_dir 978929
2018-08-01 16:59:26.810604 7ffff0d0e700  1 mds.0.migrator start finish_export_dir [dir 0x1000d7356c9 /testdir1
/ [2,head] auth{1=2,2=1,3=1,4=1,5=1} v=6017425340 cv=0/0 REP dir_auth=1,0 state=1074794500|frozentree|auxsubtr
ee f(v64 m2018-08-01 08:09:20.367198 255=0+255) n(v42032 rc2018-08-01 08:09:20.367198 2560256=2560000+256) hs=
167+0,ss=0+0 | child=1 frozen=1 subtree=1 replicated=1 waiter=0 authpin=0 tempexporting=1 0x55555ee4b100]
2018-08-01 16:59:34.920142 7ffff0d0e700  1 mds.0.migrator finish finish_export_dir 978929
```

patch that makes mds print debug message.

```
diff --git a/src/mds/Migrator.cc b/src/mds/Migrator.cc
index c765ff0158..4bb777dd03 100644
--- a/src/mds/Migrator.cc
+++ b/src/mds/Migrator.cc
@@ -1599,10 +1599,13 @@ void Migrator::export_go_synced(CDir *dir, uint64_t tid)
   // fill export message with cache data
   MExportDir *req = new MExportDir(dir->dirfrag(), it->second.tid);
   map<client_t,entity_inst_t> exported_client_map;
+
+  dout(1) << "start encode_export_dir " << *dir << dendl;
   uint64_t num_exported_inodes = encode_export_dir(req->export_data,
                                                    dir,   // recur start point
                                                    exported_client_map,
                                                    now);
+  dout(1) << "finish encode_export_dir " << num_exported_inodes << dendl;
   ::encode(exported_client_map, req->client_map,
            mds->mdsmap->get_up_features());

@@ -2200,8 +2203,10 @@ void Migrator::export_finish(CDir *dir)
   // finish export (adjust local cache state)
   int num_dentries = 0;
   list<MDSInternalContextBase*> finished;
+  dout(1) << "start finish_export_dir " << *dir << dendl;
   finish_export_dir(dir, ceph_clock_now(), it->second.peer,
                     it->second.peer_imported, finished, &num_dentries);
+  dout(1) << "finish finish_export_dir " << num_dentries << dendl;

   assert(!dir->is_auth());
   cache->adjust_subtree_auth(dir, it->second.peer);
@@ -2692,6 +2697,7 @@ void Migrator::handle_export_dir(MExportDir *m)

   bufferlist::iterator blp = m->export_data.begin();
   int num_imported_inodes = 0;
+  dout(1) << "start decode_import_dir " << *dir << dendl;
   while (!blp.end()) {
     num_imported_inodes +=
       decode_import_dir(blp,
```

```
@@ -2703,6 +2709,7 @@ void Migrator::handle_export_dir(MExportDir *m)
                        it->second.updated_scatterlocks,
                        now);
   }
+  dout(1) << "finish decode_import_dir " << num_imported_inodes << dendl;
   dout(10) << " " << m->bounds.size() << " imported bounds" << dendl;

   // include bounds in EImportStart
@@ -3021,6 +3028,7 @@ void Migrator::import_logged_start(dirfrag_t df, CDir *dir, mds_rank_t from,
   // force open client sessions and finish cap import
   mds->server->finish_force_open_sessions(imported_session_map, false);

+  dout(1) << "start finish_import_inode_caps  " << *dir << dendl;
   map<inodeno_t,map<client_t,Capability::Import> > imported_caps;
   for (map<CInode*, map<client_t,Capability::Export> >::iterator p = it->second.peer_exports.begin();
        p != it->second.peer_exports.end();
@@ -3029,6 +3037,7 @@ void Migrator::import_logged_start(dirfrag_t df, CDir *dir, mds_rank_t from,
     finish_import_inode_caps(p->first, MDS_RANK_NONE, true, imported_session_map,
                              p->second, imported_caps[p->first->ino()]);
   }
+  dout(1) << "finish finish_import_inode_caps  " << it->second.peer_exports.size() << dendl;

   it->second.session_map.swap(imported_session_map);

@@ -3080,6 +3089,7 @@ void Migrator::import_finish(CDir *dir, bool notify, bool last)
   assert(g_conf->mds_kill_import_at != 9);

   if (it->second.state == IMPORT_ACKING) {
+    dout(1) << "start sending cap imports " << *dir << dendl;
     for (map<CInode*, map<client_t,Capability::Export> >::iterator p = it->second.peer_exports.begin();
          p != it->second.peer_exports.end();
          ++p) {
@@ -3103,6 +3113,7 @@ void Migrator::import_finish(CDir *dir, bool notify, bool last)
       p->second.clear();
       in->replica_caps_wanted = 0;
     }
+    dout(1) << "finish sending cap imports " << it->second.peer_exports.size() << dendl;
     for (auto& p : it->second.session_map) {
       Session *session = p.second.first;
       session->dec_importing();
@@ -3148,6 +3159,7 @@ void Migrator::import_finish(CDir *dir, bool notify, bool last)
     mut->cleanup();
   }

+  dout(1) << "start eval " << *dir << dendl;
   // re-eval imported caps
   for (map<CInode*, map<client_t,Capability::Export> >::iterator p = peer_exports.begin();
        p != peer_exports.end();
@@ -3156,6 +3168,7 @@ void Migrator::import_finish(CDir *dir, bool notify, bool last)
       mds->locker->eval(p->first, CEPH_CAP_LOCKS, true);
     p->first->put(CInode::PIN_IMPORTINGCAPS);
   }
+  dout(1) << "finish eval " << peer_exports.size() << dendl;

   // send pending import_maps?
   mds->mdcache->maybe_send_pending_resolves();
```

#4 - 08/25/2018 08:14 PM - Patrick Donnelly

- Status changed from Need Review to Pending Backport

#5 - 08/28/2018 11:10 AM - Nathan Cutler

- Copied to Backport #32098: luminous: mds: optimize the way how max export size is enforced added

#6 - 08/28/2018 11:10 AM - Nathan Cutler

- Copied to Backport #32100: mimic: mds: optimize the way how max export size is enforced added

#7 - 09/25/2018 04:49 PM - Patrick Donnelly

- Related to Bug #24881: unhealthy heartbeat map during subtree migration added

#8 - 09/25/2018 04:55 PM - Patrick Donnelly

- Related to Bug #26858: mds: reset heartbeat map at potential time-consuming places added

#9 - 10/19/2018 10:38 PM - Nathan Cutler

- Status changed from Pending Backport to Resolved