Ceph - Subtask #5510

Feature # 4929 (Resolved): Erasure encoded placement group

Subtask # 5487 (Closed): Factor out ObjectContext / ReplicatedPG::object_contexts

ObjectContext : replace ref with shared_ptr

07/06/2013 05:27 AM - Loïc Dachary

Status:	Resolved	**% Done:**	100%
Priority:	Normal	**Spent time:**	48.00 hours
Assignee:	Loïc Dachary		
Category:	OSD		
Target version:			
Source:	Development	**Reviewed:**	
Tags:		**Affected Versions:**	
Backport:		**Pull request ID:**	

Description

work in progress
take 1

Associated revisions

Revision e1be37a3 - 08/22/2013 12:10 AM - Loïc Dachary

PG: remove unused PG::_cond

http://tracker.ceph.com/issues/5510 refs #5510

Signed-off-by: Loic Dachary <loic@dachary.org>

Revision 1688fb48 - 08/22/2013 12:10 AM - Loïc Dachary

ReplicatedPG: add Mutex to protect snapset_contexts

snapset_contexts_locks is added and locked in each function where snapset_contexts or the SnapSetContext::ref data member needs to be accessed or modified.

http://tracker.ceph.com/issues/5510 refs #5510

Signed-off-by: Loic Dachary <loic@dachary.org>

Revision 7e85c632 - 08/22/2013 12:10 AM - Loïc Dachary

ReplicatedPG: ObjectContext is made compatible with SharedPtrRegistry

When creating a new object SharedPtrRegistry::lookup_or_create uses the default ObjectContext constructor with no argument. The existing ObjectContext constructor is modified to have no argument and the initialization that was previously done within the constructor is done by the caller (that only happens three times).

The ObjectContext::get method is removed: its only purpose is to increment the ref.

The ObjectContext::registered data member is removed as well as all the associated assert()

The ObjectContext::destructor_callback data member Context is added and called by the destructor. It will allow the caller to perform additional cleanup, if necessary.

All ObjectContext * data members are replaced with shared_ptr.

http://tracker.ceph.com/issues/5510 refs #5510

Signed-off-by: Loic Dachary <loic@dachary.org>

Revision ff70e764 - 08/22/2013 12:10 AM - Loïc Dachary

ReplicatedPG: ObjectContext * becomes ObjectContextRef

The map of hobject_t to ObjectContext is made a SharedPtrRegistry owned by ReplicatedPG

```
-  map&lt;hobject_t, ObjectContext*&gt; object_contexts;
    +  SharedPtrRegistry&lt;hobject_t, ObjectContext&gt; object_contexts;
```

All ObjectContext pointers are changed into ObjectContextRef, i.e. shared_ptr.

In Watch.h std::tr1::shared_ptr<ObjectContext> is used instead of ObjectContextRef because Watch.h is included before it is defined.

http://tracker.ceph.com/issues/5510 refs #5510

Signed-off-by: Loic Dachary <loic@dachary.org>

Revision 13f6807e - 08/22/2013 12:10 AM - Loïc Dachary

ReplicatedPG: remove reference counting logic

ObjectContext manual reference counting and managing the object_contexts object involves calls to

- obc->ref++ and obc->get()

- put_object_context and put_object_contexts
- register_object_context
- assertions on obc->registered

They are all removed because SharedPtrRegistry provides the same service.

http://tracker.ceph.com/issues/5510 refs #5510

Signed-off-by: Loic Dachary <loic@dachary.org>

Revision 8c745944 - 08/22/2013 12:10 AM - Loïc Dachary

ReplicatedPG: remove lookup_object_context method

Both ReplicatedPG::lookup_object_context and ReplicatedPG::_lookup_object_context methods are provided by SharedPtrRegistry.

http://tracker.ceph.com/issues/5510 refs #5510

Signed-off-by: Loic Dachary <loic@dachary.org>

Revision 833a2250 - 08/22/2013 12:10 AM - Loïc Dachary

ReplicatedPG: replace map iterators with SharedPtrRegistry::get_next

SharedPtrRegistry does not provide an iterator equivalent to

```
map&lt;hobject_t, ObjectContext*&gt;::iterator i
```

It is replaced with a thread safe get_next method roughly used as follows:

```
pair&lt;hobject_t, ObjectContextRef&gt; i;
     while (object_contexts.get_next(i.first, &i))
```

All occurences of the iterator are replaced with get_next style traversal.

http://tracker.ceph.com/issues/5510 refs #5510

Signed-off-by: Loic Dachary <loic@dachary.org>

Revision 95349c02 - 08/22/2013 12:10 AM - Loïc Dachary

ReplicatedPG: add Context to cleanup the PG after an ObjectContext deletion

ReplicatedPG::C_PG_ObjectContext is added to encapsulate a
call to ReplicatedPG::object_context_destructor_callback method
which is reponsible for

- manually de-allocating the SnapSetContext of the ObjectContext if
  any. It will eventually be managed by a SharedPtrRegistry.

ReplicatedPG::C_PG_ObjectContext must be added to the destructor_callback
member of ObjectContext immediately after it is created.

http://tracker.ceph.com/issues/5510 refs #5510

Signed-off-by: Loic Dachary <loic@dachary.org>

Revision bd9f73d8 - 08/22/2013 12:10 AM - Loïc Dachary

ReplicatedPG: replace object_contexts.find with object_contexts.lookup

The std::map equivalent of find is SharedPtrRegistry::lookup

http://tracker.ceph.com/issues/5510 refs #5510

Signed-off-by: Loic Dachary <loic@dachary.org>

Revision d980f581 - 08/22/2013 12:10 AM - Loïc Dachary

ReplicatedPG: create ObjectContext with SharedPtrRegistry

All new ObjectContext are replaced with calls to
SharedPtrRegistry::lookup_or_create to ensure that they are all
registered. Because the constructor is invoked with no argument, care
is taken to always initialize the destructor_callback data member
immediately afterwards.

ReplicatedPG::get_object_context contains a redundant call to
get_snapset_context that is removed.

http://tracker.ceph.com/issues/5510 refs #5510

Signed-off-by: Loic Dachary <loic@dachary.org>

History

#1 - 07/07/2013 06:03 AM - Loïc Dachary

- Description updated

- Status changed from In Progress to Fix Under Review

#2 - 07/08/2013 11:09 PM - Loïc Dachary

- Description updated

#3 - 07/09/2013 12:47 AM - Loïc Dachary

- Status changed from Fix Under Review to In Progress

#4 - 07/09/2013 05:41 AM - Loïc Dachary

- Status changed from In Progress to Fix Under Review

#5 - 08/08/2013 01:34 AM - Loïc Dachary

rebased against master

#6 - 08/26/2013 03:17 AM - Loïc Dachary

- Status changed from Fix Under Review to Resolved

- % Done changed from 0 to 100

- translation missing: en.field_remaining_hours set to 0.00

merged and the rados tests results from the past few days does not exhibit problems that can be obviously traced back to this patch series

#7 - 03/07/2014 04:50 AM - Loïc Dachary

- Estimated time set to 0.00 h