

Document Number: N4001
Date: 2014-05-23
Authors: Michael Wong
Project: Programming Language C++, EWG, SG5 Transactional Memory
Reply to: Michael Wong <michaelw@ca.ibm.com>

SG5: Transactional Memory (TM) Meeting Minutes 2014/02/03-2014/05/19

Contents

Minutes for 2014/02/03 SG5 Conference Call	2
Minutes for 2014/02/17 SG5 Conference Call	6
Minutes for 2014/03/10 SG5 Conference Call	9
Minutes for 2014/03/31 SG5 Conference Call	13
Minutes for 2014/04/28 SG5 Conference Call	18
Minutes for 2014/05/12 SG5 Conference Call	21
Minutes for 2014/05/19 SG5 Conference Call	27

Minutes for 2014/02/03 SG5 Conference Call

Minutes by Victor

> The current secretary rota list is:

Torvald, Jens Maurer, Mike Spear, Tatiana, Hans,
Maged, Justin, Michael Scott, Mark, Michael Wong, Victo

> Reminder: We use the Secretary Rota to determine who is responsible for
> minutes at any given meeting. The first name on the list that is present
> at the meeting will be responsible for them. Upon completing the
> minutes, they should move their name to the end of the rota. In
> face-to-face meetings, minutes duties will be assigned for a morning
> session or an afternoon session or an evening session (if applicable) so
> as to distribute the load fairly (but not too fine grained; consider it
> a transaction).

>

> Agenda:

>

> 1. Opening and introductions

>

> 1.1 Roll call of participants

Michael Wong, Michael Scott, Maged, Mike Spear, Victor, Justin, Paul McKenney

> 1.2 Adopt agenda

ADOPTED.

> 1.3 Approve minutes from previous meeting, and approve publishing

> previously approved minutes to ISOCPP.org

APPROVED.

> 1.4 Review action items from previous meeting (5 min)

> 1.4.1. Michael Wong: AI: comments on changes and spelling issues from Mike Spear and
Justin

DONE.

> 1.4.2. Victor :

> Add to Section 5 one sentence para to say that allocation and dealloc functions are necessarily

- unsafe and user-replaced new/delete are also tx-unsafe
- > Then add to Section Section 7 the full explanation
 - > Change Section 7 title to Discussion of Feedback and other Issues,
 - > Make heading and sub-heading uniform

DONE.

- > Change Revision to 2

DIDN'T DO. (Sorry.)

- > 1.4.3. Michael, start on examples

DONE: sent out new version of N3859 with examples at the end.

- > 1.4.4 check on EWG and LEWG answer on timeslot.

DONE: both acknowledged and confirmed that we will have a slot, hopefully Tuesday.

- > 1.4.5 Considering going to Issaquah C++ Standard meeting, Hotel Booking
- > deadline Jan 20, 2014:
- > <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3707.pdf>

Confirmed: Michael W, Jens, Hans, Maged, Torvald, Justin, Tatiana, Victor

Michael W: Tuesday to Thursday are most important days.
Justin flying in on Tuesday morning.

- > 2. Main issues (50 min)
- >
- > 2.1 Discuss C++ Std meeting logistics: be prepared to be there Tuesday to Friday.
- >
- > 2.2 Feedback from EWG and LEWG chairs

(Elided: see Michael's agenda mail if you want to see this.)

- > 2.3 Examples

Michael W: Use of "(atomic) transactions"?

Victor: I mostly eliminated use of "transaction", but put in sentence saying that atomic blocks are sometimes called transactions, in part so that use of "transaction-safe" makes sense.

Michael Scott: We should use "atomic blocks", particularly for the syntactic construct.

"Transaction" might still be used for execution of an atomic block.

=> Agree to above usage.

Michael W: Section 9.8 says we can't have synchronized blocks within atomic blocks, but didn't we change that?

Victor: Yes. We should note explicitly that this is different from previous proposal.
=> Agree to update Section 9.8.

Michael W: Do we need Section 9.14, with very short lambda example?

Michael Scott: What's the point?

=> Agree to eliminate Section 9.14

Michael W: I have enough to edit the section.

AI: Michael W to update Examples section of N3859.

Victor: Macros for `transaction_safe` keyword, etc. have changed: no longer include space afterwards, so `.tex` file needs to be edited to insert space explicitly.

> 2.4 Slide prep

EWG: Victor to present N3859

LEWG: Michael W to present N3862

Try to have them in that order (but we can't guarantee this). Make sure that they don't overlap!

Michael W: Will try to arrange for Mike Spear to call into LEWG presentation.

AI: Michael W, Mike Spear and Justin to figure out how to arrange call into LEWG meeting.

Discussion of LEWG presentation:

Michael W: Should be easy: might just walk through paper (N3862), which is common for this group.

Mike Spear: Found bug in gcc: they use reference counting for strings (not allowed in C++11).

Michael W: Torvald said gcc has an alternate kind of strings (with different name) that isn't ref counted. We could use that instead.

Mike Spear: Any other objects that are reference counted?

MW: Not really, except for shared pointers, which are new in C++11.

Justin: The problem is that ref counting does an atomic increment, which isn't allowed in a transaction. Couldn't we fix that by using a transaction to do the atomic increment?

Mike Spear: Yes, but then it won't work with legacy code (i.e., without transactional support).

Maged: Will this be a problem in the future, with respect to shared pointers?

Michael W: Perhaps, but shared pointers are new in C++ so no/minimal legacy code.

Discussion of EWG presentation:

Victor: How much familiarity with previous proposal should I assume?

Michael W: Include brief recap—there may be new people—but no need to go into detail. Most people will have seen/heard much of it before.

Victor: Two possible approaches:

1. Describe new proposal de novo, then say how it's changed from old based on feedback at/since last meeting.

2. Refer to previous proposal first, give brief high-level description of it, summarize feedback, describe how we've adapted proposal to incorporate feedback, and then describe new proposal.

Justin: Prefer second approach: Otherwise, people may raise concerns about new proposal without realizing that it reflects feedback we've received in the past. With second approach, if they push us back towards old proposal, they'll do so knowingly.

Michael W: Agree with Justin.

=> Agree to second approach.

AI: Victor to write and send out draft slides.

Justin: It would be good if everyone attending Issaquah would refamiliarize themselves with N3859 to facilitate discussion at meeting, especially in post-presentation hallway discussions, so we can get better address any issues people might raise, and get more helpful feedback.

Michael W: Agree: most feedback comes from hallway discussions.

AI: Everyone (going to Issaquah) to reread N3859.

> 3. Any other business

Next meeting is next week at Issaquah (no phone meeting).

Next phone meeting is Feb 17.

> 4. Review

> 4.1 Review and approve resolutions and issues [e.g., changes to SG's

> working draft]

> 4.2 Review action items (5 min)

AI: Michael W to update Examples section of N3859.

AI: Michael W, Mike Spear and Justin to figure out how to arrange call into LEWG meeting.

AI: Victor to write and send out draft slides.

AI: Everyone (going to Issaquah) to reread N3859.

Meeting adjourned early!

Minutes for 2014/02/17 SG5 Conference Call

Minutes by Jens

Attendees: Mark Moir, Michael Spear, Michael Scott, Michael Wong, Maged Michael, Victor Luchangco, Jens Maurer (minutes), Hans Boehm, Justin Gottschlich, Torvald Riegel

1.2 Agenda

unanimously approved

1.3 Minutes from previous meeting

unanimously approved

1.4 Action items

all considered done

2.1 Discuss C++ Std results

Michael Wong: Report from WG21 meeting in Issaquah. Creation of new work item was approved by WG21; now up for ballot in SC22.

We also had presentations to EWG (Evolution Working Group) and LEWG (Library Evolution Working Group). First two from EWG were easy to address (already fixed in N3919).

LEWG had more comments, not as easy to address.

General plan forward: Have wording available for the Rapperswil meeting (June 2014), and approve a Working Draft for the TS (Technical Specification) there. CWG (Core Working Group) to review drafting in a teleconference before Rapperswil.

Three ballots; need 5 NB to be interested; need project editor and backup project editor. Most, but not Switzerland, are on board.

2.2 Feedback from LEWG and EWG

Feedback from EWG:

- Changed keywords for atomic blocks to `atomic_noexcept/atomic_cancel/atomic_commit`.

- Don't use `std::abort` to define actions of `atomic_noexcept`, and `atomic_cancel`, aim for undefined behavior.

Jens: The claim that a `noexcept` function does not guarantee a call to `std::terminate()` is factually wrong. Suggest to leave as-is and send e-mail to EWG mail reflector.

- We need to add wording to our spec that we do conversion from transaction-safe lambda to a function pointer that it remains transaction-safe.

Action Item for Jens to address this in the wording.

Victor: Two open issues in N3919:

- Issues when calling `std::terminate()`. Discuss this again.
- Concurrent initialization of function-local statics

Feedback from LEWG:

- extern template issue (declaration of explicit instantiation allows out-of-line definition)

Michael Spear: Disagree that safe-by-default solves this issue.

- swap does not need to be `transaction_safe`

Jens: No problem apparent.

- In what namespace should we place TM-enabled containers?

Michael Wong: Usually, TS uses `std::experimental` for new components, but we'll modify "std".

Torvald: This is not simple, since changes to existing components may cause changes to the ABI. In the TM case, this is not as critical as with e.g. the changes to `std::future`.

Michael Spear: Turn to safe-by-default. We wanted to specify that `std::list<>` is transaction-safe if the T you're using it with is also transaction-safe. Thus, we're not putting any annotations on `std::list`. Suppose
extern template `std::list<Foo>`;
is in a header file, with a definition in the separate .cc file.
What about shipping an object code library?

Jens: Yes, this is an issue, if we don't have safe-by-default. If we do, we need to evaluate the object code bloat caused by safe-by-default in the real world.

Michael Spear: How to specify that the allocator is transaction-safe? Change the type of the `allocate/deallocate` member functions?

Jens: Type of enclosing container changes when allocator changes type depending on transaction-safe or -unsafe. This might derail existing helper functions in large codebases.

Action Item: Post to the mailing list and explain "extern template".

Torvald: Evening session: Transactions are like fences, or like atomic accesses?

Michael Spear: We should say that accesses in the same thread are ordered with respect to transactions in the same thread.

Next meeting on March 10.
Meetings on March 3 and March 17 are cancelled.

Adjourned.

Minutes for 2014/03/10 SG5 Conference Call

Minutes by Maged

- > The current secretary rota list is (I moved people who took notes at
- > the meeting to the end)
- >
- > * Torvald, Mike Spear, Tatiana, Hans, Michael Scott, Mark, Michael
- > Wong, Victor, Justin, Jens Maurer, Maged, Justin
- > *
- > Reminder: We use the Secretary Rota to determine who is responsible for
- > minutes at any given meeting. The first name on the list that is present
- > at the meeting will be responsible for them. Upon completing the
- > minutes, they should move their name to the end of the rota. In
- > face-to-face meetings, minutes duties will be assigned for a morning
- > session or an afternoon session or an evening session (if applicable) so
- > as to distribute the load fairly (but not too fine grained; consider it
- > a transaction).
- >
- > Agenda:
- >
- >
- > 1. Opening and introductions
- >
- > 1.1 Roll call of participants
- >
- > * Mike Spear, Tatiana, Hans, Michael Scott, Mark, Michael Wong, Victor,
- > Jens Maurer, Maged
- > *
- >
- > 1.2 Adopt agenda
- >
- > 1.3 Approve minutes from previous meeting, and approve publishing
- > previously approved minutes to ISOCPP.org
- >
- > 1.4 Review action items from previous meeting (5 min)
- >
- > 1.4.1: Post to the mailing list and explain "extern template" issue. (who?)
- >
- > *DONE*
- >
- > 1.4.2. Next C++ Std meeting: Raperswil, Switzerland, June 16-21 (aim for
- > June 17-19)

- > <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3805.pdf>
- > Many hotels are already rapidly closing:
- > <http://wiki.ifs.hsr.ch/WG21Meeting/wiki.cgi?WG21Meeting>
- >
- > *ONGOING*
- >
- > *Michael Wong spoke about the next milestones for the proposal to be approved as a TS. See Agenda below under 2.1*
- > *The meeting in Raperswil and all subsequent meetings are important.*
- > *Jens: Knowing standard wording is important.*
- > *Victor: We should aim to get feedback*
- > *Jens: We should fewer changes*
- show quoted text -
- > *
- > *
- > *Jens: CWG conference call can be as early as April for presenting the updated wording*
- > *Action item: All review Jens updated wording (blue marking) and open issues*
- > *Justin: Question about STL list size specification*
- > *Jens: Recommends not changing library specification*
- > *Mark/Michael Scott: Not eliminate size but add a variant with weaker consistency requirements*

Just to be clear, I was not saying what I thought should be done, only suggesting an alternative to consider, namely augmenting an existing data structure with new operations, such that it would continue to have the same semantics if used only with the existing interface, but other (hopefully still useful) semantics if the new operations were used.

Of course there are a number of issues to consider, but I wanted to point out that there could be options other than largely duplicating an existing interface due to a small required difference.

I didn't mean to divert the conversation and it made sense to nip it in the bud.

Cheers

Mark

- > *Michael Spear / Michael Wong: Let's not spend too much time on this. We
- > should focus on scalable classes.*
- > *
- > *
- > 2.1.2 Resolve Extern template issue and possibly report back to LEWG.
- >
- > New action item: Get more information about the templates issue
- >
- > 2.1.3 Transactionalize many more containers (get GCC's Jonathan Wakely,
- > Clang's Howard Hinnant and Dinkumware's PJ Plauger involved)
- >
- > *Michael Spear; Many containers are not safe due to atomic operations
- > for referenced counted strings.*
- > *Michael Wong / Jens: We should work around these problems*
- > *Michael Wong: We need more volunteers for container transactionalization*
- > *Jens: This is not essential for the critical path to a TS*
- > *Michael Spear / Michael Wong / Victor: Agrees with Jens. The
- > implementation is a sanity check*
- > *More discussion on the reflector*
- >
- > 2.1.4 have LWG review wording (from 2.1.3) possibly have Alistair (LWG
- > chair and other LWG members) callin to our meeting
- >
- > 2.2 Discuss Evening Session topics
- >
- > We can start on this first by reviewing the issues, but I am going to
- > invite Chandler to the following meeting March 31 as I think we have
- > more then enough to discuss this time and prepare ourselves.
- > <https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/1RorkphKvsc>
- > <<http://?hl=en&fromgroups#%21topic/tm/1RorkphKvsc>>
- >
- >
- > 3. Any other business
- >
- > 4. Review
- > 4.1 Review and approve resolutions and issues [e.g., changes to SG's
- > working draft]
- > 4.2 Review action items (5 min)
- >
- > *New action item: All review Jens updated wording (blue marking) and
- > open issues*
- > *New action item: Get more information about the templates issue*
- > *New action item: All think about and discuss the empty transaction
- > ordering issue*
- >
- > 5. Closing process

- > 5.1 Establish next agenda
- > 5.2 Future meetings: Mar 31 teleconference
- > Mar 3: (Michael Wong away at LSU), many others at Transact.
- > Congratulation Mike Spear and Tatiana. Cancelled
- > Mar 10: Plan forward schedule
- > Mar 17: Euro TM school (Michael Wong, Victor, Torvald away) cancelled
- > Mar 31: Review with Chandler
- > Apr 14: (Michael Wong away at llvm dev con, then ACCU, then OpenMP
> Cambridge UK)
- > Apr 28: (Michael Wong away at ADC++ Munich)
- > May 12; (Michael Wong away at C++Now)
- > May 26:
- > Jun 9:
- > Jun 16: C++ Standard meeting Rapperswil
- >
- >
- > 5.3 Adjourn

Minutes for 2014/03/31 SG5 Conference Call

Minutes by Mike Spear

When you hear a single beep, someone has joined the call. When you hear a double beep, someone has dropped from the call. With large numbers of participants, audio interference can be a problem. Please try to keep your phone muted whenever possible. If your phone does not have a mute button, the bridge will mute or un-mute your line if you dial *6.

The current secretary rota list is (I moved people who took notes at the meeting to the end)

Torvald, Tatiana, Hans, Michael Scott, Mark, Michael Wong, Victor, Justin, Jens Maurer, Maged, Mike Spear

Reminder: We use the Secretary Rota to determine who is responsible for minutes at any given meeting. The first name on the list that is present at the meeting will be responsible for them. Upon completing the minutes, they should move their name to the end of the rota. In face-to-face meetings, minutes duties will be assigned for a morning session or an afternoon session or an evening session (if applicable) so as to distribute the load fairly (but not too fine grained; consider it a transaction).

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Present: Mike Spear, Michael Scott, Victor, Michael Wong, Tatiana, Justin, Torvald, Jens Maurer,

Not Present: Hans, Mark, Maged

1.2 Adopt agenda

Adopted.

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

Unanimously approved.

1.4 Review action items from previous meeting (5 min)

1.4.1: All review Jens updated wording (blue marking) and open issues (All)

(There was discussion on the email list)

1.4.2: Get more information about the templates issue (Michael)

Michael Wong had a discussion with Jeffrey and Diedmar?, awaiting reply (keep as action item)

1.4.3: All think about and discuss the empty transaction ordering issue (All)

This will be our main discussion today.

1.4.4. Next C++ Std meeting: Raperswil, Switzerland, June 16-21 (aim for June 17-19)

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3805.pdf>

Many hotels are already rapidly closing:

<http://wiki.ifs.hsr.ch/WG21Meeting/wiki.cgi?WG21Meeting>

If you think you might go, at all, then book hotels now and cancel later.

Victor could go, but would rather not if it can be avoided. Michael Wong thinks it would be OK for Victor to skip this one.

2. Main issues (50 min)

2.1 Review Core wording issues

A few issues from Michael Scott:

In Jens's wording, there were a few typos that Michael sent to Jens already. The main open issue is the ordering semantics. Michael Scott likes the current wording, but there seems to be some push for something more relaxed (i.e., happens before edges between loads and stores, vs between transaction boundaries).

(General discussion of how this relates to the empty transaction issue).

Victor: the interesting question is unintentional empty transactions (i.e., due to compiler optimizations or partial template instantiations), which becomes a matter of memory ordering.

Tatiana: who is in favor of this relaxation? Victor supports "thinking about it".

Victor: It would be useful to have Torvald on the call, since Torvald and Chandler spoke about this.

Mike: we need to be sure that read-only transactions have the right semantics. For example, in GCC today, read-only software transactions do not have the same quiescence as writers. Proxy privatization could be an issue. (Discussion of proxy privatization, with a three-transaction example, where third transaction is read-only).

Discussion between Victor, Michael, Tatiana, and Torvald about privatization being a catch-all phrase for describing the ways we transition the state of a variable.

General discussion of the symmetry between undo and redo for privatization problems. It does appear that GCC today is vulnerable to proxy privatization problems.

The key point is that we don't want to argue about whether this is a bug in the implementation, but rather whether we should specify this behavior as legal or not.

Summary: this makes clear that the "strong semantics" of ordering at transaction boundaries has a cost that we can understand today. Total ordering will raise costs because we'd have to reorder writer quiescence with writers dropping locks, and/or quiescence for readers. So even in the absence of eliding empty transactions, we can say that SGLA would add to the overhead of GCC and similar STM implementations. We need to choose whether we want the easy explanation or the high-performance implementation.

Victor: this is an example that requires one to understand the implementation... it is hard to even explain this to the programmer.

Discussion between Tatiana and Victor about explanations of the phenomenon and the semantics (and discussion of whether it's lock-based semantics or not).

Mike: propose a strawman: writes can't be hoisted out of a transaction, and all writers are ordered. This allows Torvald's high performance, but also allows an easy fix (as in write-skew anomaly for snapshot isolation... just add a write and you get ordering).

Michael Scott and Justin: Generally don't like this strawman.

Tatiana: but what about $x = x...$ is that a write? Is it intentional?

Victor: there are questions about privatization and questions about memory order, and whether they are intertwined. Consider the following weakening: we currently augment synchronizes-with to include transactions, and right now begin and end of transactions participate in the total order of synchronizes-with. Victor proposes that instead we say they only order if the transactions conflict with each other... so empty transactions don't conflict. Note that Mike's proxy privatization problem isn't a problem here, because there is a conflict ordering.

Michael Scott: but this means we need a formal definition of what it means to conflict. Does dead code conflict? What about variables that stay in registers? Not-live-out stuff, or stuff that gets hoisted in or out of a conditional that is not taken, but the hoisted access is still performed?

Torvald: Note that sequentially consistent fences and sequentially consistent accesses have different meanings today in C++11. Since there are differences in the memory model already, we might be able to do something similar for transactions.

Michael: Could we do this in revision #2?

Victor: Weakening semantics is hard.

Michael: We can't weaken the default behavior, but we can add special syntax.

Torvald: our semantics right now might be stronger than sequentially consistent atomics.

Michael Wong: we also forbid some hoisting right now in threaded mode, so hoisting from transactions may be incompatible with regular threaded code.

(Request to explain Torvald's claim).

Torvald: if we can use an empty transaction like a fence, but we can't get the same effect with a sequentially consistent load or store to a dummy variable. You need a sequentially consistent fence or it's not ordered.

(Repeated discussion about how empty transactions can arise due to optimization)

One last note about core wording:

Victor would like to discuss use of "transaction" vs. "atomic block" in the spec.

(Seems to be consensus on this... we will still say "transaction-safe", and Jens will update the document accordingly).

2.2 Discuss Evening Session topics

<https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/1RorkphKvsc>

3. Any other business

4. Review

4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]

4.2 Review action items (5 min)

Continue updating the wording changes.

Mike will send 3-transaction proxy privatization example, so that Michael Wong can share with Chandler.

Michael Wong awaiting reply from Diedmar?.

5. Closing process

5.1 Establish next agenda

We are likely to return to the memory ordering issues next meeting.

Meeting adjourned.

5.2 Future meetings: Mar 31 teleconference

Mar 3: (Michael Wong away at LSU), many others at Transact. Congratulation Mike Spear and Tatiana. Cancelled

Mar 10: Plan forward schedule (Done)

Mar 17: Euro TM school (Michael Wong, Victor, Torvald away) cancelled

Mar 31: Review with Core Wording and Evening Issues

Apr 14: (Michael Wong away at llvm dev con, then ACCU, then OpenMP Cambridge UK)

Apr 28: (Michael Wong away at ADC++ Munich)

May 12: (Michael Wong away at C++Now)

May 26:

Jun 9:

Jun 16: C++ Standard meeting Rapperswil

Minutes for 2014/04/28 SG5 Conference Call

Minutes by Torvald

Minutes Apr 28/29 SG5 Teleconference Call



Participants:

Torvald, Michael Scott, Michael Wong, Victor, Justin, Jens, Maged, Mike Spear, Chandler Carruth, Paul McKenney, Hans, Tatiana

On Wed, 2014-04-23 at 11:31 -0700, Michael Wong wrote:

- > 1.4 Review action items from previous meeting (5 min)
- >
- >
- > 1.4.1: All review Jens updated wording (blue marking) and open issues
- > (All)
- >
- > 1.4.2: Get more information about the templates issue (Michael)
- >
- > Sent to Dietmar April 10.

Dietmar hasn't replied yet.

- > 2.1 Discuss Evening Session topics, specifically empty TX ordering
- > (with Chandler present after I shared our position with Chandler at
- > the llvm dev conf):

Mike Spear: This is about the default semantics (weakening when special syntax is used is fine). [Explains his proxy privatization example.]

[Torvald and Michael Scott discuss privatization vs. whether txns are seq-cst fences.]

Hans: Empty txns weaker than seq-cst fences. [Explains details (eg, interaction with other acq/rel fences or accesses, then drops off.)

Chandler: My initial concerns unrelated to proxy privatization. Actual concerns about "empty" txns (ie, txns that contain accesses to memory not accessible to any other thread). Similar to difference between seq-cst fence and seq-cst RMW memory access. Txn is called atomic, so model for seq-cst atomics should be fine.

Michael Scott: How can programmers reason about which memory a thread accesses?

Chandler: Txn participates in total order if it accesses any memory operation visible to other threads. Programmer hasn't written a fence, so doesn't get one.

Hans: There might be some fencing behavior [even with what chandler said].

Chandler: [Gives analogy of seq-cst atomic access (not fence).]
Compiler can make this nonatomic if proving the targeted location isn't accessible to any other thread.

Victor: [???

Hans: Will look for description of seq-cst fence vs. seq-cst atomics, potentially send out.

Chandler: His compiler removes seq-cst atomics to locations not accessible to other threads. Open question, believes it to be correct.

Hans: Question is whether a txns always implicitly accesses some global object so that it participates in global order.

Chandler: All I want is that atomic txns don't sync with other atomic txns if code written only accesses local memory.

Tatiana: [Concerns over composability.]

[???

Mike Spear: Spec addresses value speculation?

Hans: Might have implications for it.

Mike Spear: [Gives example.]

Chandler: [Believes Mike described something different, and compiler transformation wouldn't be possible.]

Chandler: Make clear what the intent is: fence or (RMW) access. IMO, intent should be RMW.

Tatiana: Though about txns having semantics independently of what they access. Intuitive semantics matter.

Chandler: Txns should be like using global lock if there's any access to shared memory.

Michael Scott: Don't want programmers to have to reason about txns based on what's inside of them. Reluctant about special casing for empty txns.

Chandler: [Explains motivating example.]

Chandler: Removing ops on a thread-local lock is allowed.

Victor: This is where single-global lock analogy fails.

AIs

Hans: Send write-up of seq-cst fences vs. seq-cst accesses.

> May 12; (Michael Wong away at C++Now)

More discussion of this issue.

Minutes for 2014/05/12 SG5 Conference Call

Minutes by Justin Gottschlich

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Mike Spear, Victor, Michael Scott, Maged Michael, Tatiana, Torvald, Hans, Justin

1.2 Adopt agenda

Approved.

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

Approved.

1.4 Review action items from previous meeting (5 min)

1.4.1: All review Jens updated wording (blue marking) and open issues (All)

Done, discussion pending.

1.4.2: Get more information about the templates issue (Michael)

Sent to Dietmar April 10.

Ongoing.

1.4.3: All think about and discuss the empty transaction ordering issue (All)

Done, discussion below.

1.4.4: Send write-up of seq-cst fences vs. seq-cst accesses to SG5 Google group (Hans)

Done.

1.4.5. Next C++ Std meeting: Rapperswil, Switzerland, June 16-21 (aim for June 17-19)

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3805.pdf>

Many hotels are already rapidly closing:

<http://wiki.ifs.hsr.ch/WG21Meeting/wiki.cgi?WG21Meeting>

2. Main issues (50 min)

2.1 Continue discussion about empty TX ordering:

<https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/eNub3grW6Ko>

<https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/1RorkphKvSC>

Michael Scott: A point of clarification: Chandler is not really talking about empty transactions per se. He's talking about transactions that are not accessing shared memory, which can have their memory accesses moved outside of the transaction, thereby making it empty, which then means it does not need to participate in the total global ordering of transactions.

Everyone agrees.

Tatiana: What I would like to know is what are the practical technical issues of eliminating transactions?

Victor: Let's take a step back and consider the cases where this arises. In Chandler's example, one has a data structure that's intended for concurrency, but then is not actually used for it in some special cases.

Tatiana: Okay. And what is the synchronization cost for these empty transactions?

Torvald: There are at least two costs. First, you can't reorder instructions above/below it, unless you prove that it doesn't make a difference. The second point, you have an ordering that must follow synchronizes-with.

Michael Scott: Imagine I have a write of a shared var and a read of a shared var after. Must put in seq con fence between instructions.

Tatiana: I reviewed our DLA SPAA 2008 paper: two txes are ordered only if they access the same memory location. At some point, for seq const case, DLA is equiv to SGL as long as transactions don't contain other forms of synchronization. In such cases, empty transactions can be eliminated.

Victor: But we already have that restriction?

Tatiana: Not entirely; synchronized blocks can access locks, atomics, etc.

Victor: Okay, that's a good point. If I only use simple atomics (no relaxed or acquire/release ordering), is there any behavior that's not equiv to transactionally seq const? In general, I'm on Chandler's side.

Tatiana: I can write an example (publication via empty transaction), imagine you are writing some data, then you have a flag that ... in short, empty transaction acts as a fence:

<i>Thread 1</i>	<i>Thread 2</i>
<i>write data</i>	
<i>atomic {}</i>	
<i>flag = true</i>	
	<i>synchronized { if (flag) read data; }</i>

flag must be an atomic C++ variable.

Torvald: Dekker's example using an empty atomic transaction is a case that doesn't require relaxed transactions / synchronized blocks.

Victor: Okay, let's suppose all differences in execution behavior involve relaxed memory order. Is anyone opposed in principle to Chandler's optimization in this case?

Hans: My concern with relaxed memory examples is we need to define what these things mean. And we don't yet know if this is easy or hard.

Tatiana: You need to be able to say that your program follows this simple property, then you can expect this behavior.

Victor: I agree, I'd just like to limit the scope of these concrete examples. In the cases, where empty transactions are meaningful in the absence of relaxed or acquire/release memory accesses (rather, only the cases where memory accesses are sequentially consistent).

Tatiana: But the point is that atomic transactions may be extended to support atomic variables.

(General consensus, is that we may only require empty transactions when relaxed or acquire/release memory accesses are made.)

Tatiana: It might be what we are saying is we are using DLA. As long as your program doesn't use relaxed transactions / synchronization blocks ...

Justin: I have two issues with this optimization. First, I think it complicates the implementation space when we have non-shared memory accesses in atomic transactions and still need to ensure failure atomicity in the event of exceptions. Second, although I agree with that the optimization is a good thing to have in general, there seems to be no concrete evidence that the optimization will be highly applicable. This is because, to the best of my knowledge, identifying shared memory accesses in C++ programs is an open problem for static analysis.

Torvald: Yes, however, I'm not so pessimistic about optimization opportunities. If people only use transactions when they are needed for concurrency, it's true we won't have many optimization opportunities. However, I suspect that in many cases people will use them in cases just to be safe and there will be many cases where they are not necessary for shared memory synchronization.

Victor: It's an interesting point about the exceptions as it makes me pause about my preferred behavior for atomic transactions (that is, those that cancel on exceptions) as they could present an optimization bottleneck.

Victor: In any case, we just need to ask how our transactions fit into the existing C++ memory model.

*Tatiana: Here is a function, return $A.i * k$. $A.i$ is global. Can I assume that $A.i$ is going to be accessed?*

*Mike Spear: What if the parameter is k , $k / A.i$. What if we have a multiplication: $A \rightarrow i * k$, but k is zero?*

Tatiana: I'm scared about the case where an empty transaction is accessed as a fence.

Victor: If you're using transactions in this way, then you are using this the wrong way.

Mike Spear: Can I move non-shared memory accesses out of a transaction that does not make it an empty transaction?

Justin: Continue discussion next week in impromptu meeting? Who can make a meeting at that time?

Victor, Tatiana, Mike Spear, Torvald, Justin, Michael Scott.

AI: Michael Scott –send note out for clarification on Chandler's optimization.

AI: Justin - schedule impromptu SG5 meeting for May 19 to continue this week's discussion.

Justin,

Thanks for taking notes during very fast and intense discussion. I'd like to make two clarifications.

I haven't spelled out the publication via empty transaction example correctly in the beginning of the meeting. It should be:

```
// data and flag are atomic variables  
  
Thread 1           Thread 2  
  
write data  
  
atomic {}  
  
flag = true  
  
  
synchronized { t = data; if (flag) use t; }
```

The empty transaction in Thread 1 acts as a fence, guaranteeing that Thread 2 saw the write to data, if it sees `flag == true`.

As Michael and/or Victor pointed out, this is important if data or flag are weakly ordered atomic variables. If both data and flag are SC atomics, they are ordered with or without the empty transaction in between so they can be eliminated.

> Tatiana: It might be what we are saying is we are using DLA. As long as your program doesn't used relaxed transactions / synchronization blocks ...

My point was that we might be converging on a coherent story: Atomic transactions have DLA (or some other weaker than single global lock (SGL)) semantics. But, for the programs that contain only transactions, locks and sequentially consistent atomics, this semantics is indistinguishable from SGL. Of course, for this idea to work, we still need the concrete semantics for atomic transactions and the proof that for sequentially consistent executions it's equivalent to total global ordering.

Thanks,

Tatiana

2.2 Final review of core wording issues (needed for June 2 meeting)

3. Any other business

4. Review

4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]

4.2 Review action items (5 min)

AI: Michael Scott –send note out for clarification on Chandler's optimization.

AI: Justin - schedule impromptu SG5 meeting for May 19 to continue this week's discussion.

5. Closing process

5.1 Establish next agenda

5.2 Future meetings:

June 2 telecon

Mar 3: (Michael Wong away at LSU), many others at Transact. Congratulation Mike Spear and Tatiana. Cancelled

Mar 10: Plan forward schedule (Done)

Mar 17: Euro TM school (Michael Wong, Victor, Torvald away) cancelled

Mar 31: Review with Core Wording and Evening Issues

Apr 14: Many people away. Cancelled

Apr 28: (Michael Wong away at ADC++ Munich) Chandler to call in to discuss his feedback to our collective position.

May 12: (Michael Wong away at C++Now) Note: May 23 is deadline for C++ Std Mailing.

May 26: Canceled due to Memorial day

June 2: CWG review

Jun 9: Possibly discuss extern template issue with Dietmar

Jun 16: C++ Standard meeting Rapperswil

Minutes for 2014/05/19 SG5 Conference Call

Minutes by Tatiana

1. Opening and introductions

1.1 Roll call of participants

Victor, Jusitn, Tatiana, Maged, Torvald, Michael Scott, Michael Wong

1.2 Adopt agenda

Adopted

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

Approved

1.4 Review action items from previous meeting (5 min)

1.4.1: All review Jens's updated wording (blue marking) and open issues (All)

Still open. Please post any comments.

1.4.2: Get more information about the templates issue (Michael)

Still open. The issue is from Eric niebler will join us on June 9th.

1.4.3: All think about and discuss the empty transaction ordering issue (All)

Ongoing.

1.4.4: Post to SG5 reflector a clarification of Chandler's optimization (Michael Scott)

Done

1.4.5: Next C++ Std meeting: Rapperswil, Switzerland, June 16-21 (aim for June 17-19)

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3805.pdf>

Many hotels are already rapidly closing:

<http://wiki.ifs.hsr.ch/WG21Meeting/wiki.cgi?WG21Meeting>

Michael Wong: Paper mailing deadline is this Friday. We should prepare couple of papers to keep it on the radar. He wrote 2 papers – on library and core wording, just to keep them on

people radar. Michael will send the papers so that everybody could look at them. The core paper has been updated with all the issues discussed at the last meeting.

Action items:

- Michael Wong: Sent the papers
- Everybody: send the comments before May 22.

2. Main issues (50 min)

2.1 Continue discussion about empty TX ordering:

Victor: We should limit the scope of discussion to atomic blocks and consider synchronized block once we decided what to do about atomic.

Tatiana: Let's consider four options for atomic transactions:

1. Keep semantics as it is
2. Total order on atomic transactions excluding transactions that do not access shared memory locations
3. Atomic transactions behave as if all their reads and writes are executed atomically.
4. Add new kind of transactions that allow eliminating empty transactions

Michael Scott: There is also a proposal that Maged sent.

Victor: But it's about synchronized blocks.

Michael Wong: I've been discussing this change with Herb and Clark to see if it needs to be revisited by the evolution group. Their opinion that it does not require re-evaluation with the evolution group.

Discussion on whether we need to revisit this issue with evolution group.

Victor argues that we should not.

Michael Wong: We should describe the change and our resolution to the change in the paper.

Tatiana asks to clarify the procedural issue.

Michael Wong: If you change the proposal in a significant way one needs to go back to re-ask for evolution approval again. The question is whether we should do this or not. Michael suggests that we should not.

Victor: We should make a note that says that we are considering ways to allow this optimization. We should not bring it up till we have a solution.

Michael Wong: Just because it is approved by evolution it does not mean we cannot change it. After we get full approved for new item with the draft wording paper, we switch to the process where we are writing changes against draft paper. It's fine to come with the changes.

Victor: I don't want to bring it to evolution before we are ready.

Back to technical discussions.

Group agrees that it's OK not to consider Maged suggestion today because it's primarily about synchronized transactions.

Victor: Is it true that one cannot observe if atomic transaction with current semantics acts as sequentially consistent fence or not in a program that uses only SC atomics?

Michael Scott: I believe the answer is no.

Victor: In this case, having weaker atomic semantics (option 3) for atomic transactions is OK.

Michael Scott: I prefer option 2, as it preserves simplicity of atomic transactions.

Victor: There are two things – what we want to guarantee and what we are going to put in the specification. Specification should tie into synchronization guarantees that we already have about transactions.

...

Michael Scott: I am not advocating option 2. But, if we want to relax it at all, I'd prefer option 2 to option 3.

Victor: What in our specification prevents us today from eliminating empty atomic transactions? May be, we don't have problems at all.

Michael Scott: Our wording is pretty clear. Transaction start is acquire and transaction end is release.

Michael Scott, Victor and Tatiana discuss if empty transactions can be eliminated when program does not use non-SC atomics.

Torvald: Han's example of starting a new thread in a relaxed transaction is a counter-example.

Michael Scott: Proposal – while a compiler is free to elide transactions that do not touch shared memory, a programmer can ignore that optimization if he does not use sequentially consistent atomics.

Tatiana: could we express option 2 in a formal language?

Michael S: I am defending an option that we do not particularly like. We could 2 by saying that transactions that access at least one shared memory location induce happens-before edges ..

Justin: I don't think everybody chimed in on what they prefer. Anybody else would like to give their thoughts?

Torvald: I like 3 best. We cannot solve this before the paper submission deadline. I think we could summarize discussions that we have and counter examples and put that in a paper that we send either before or after deadline. I think we need time before Rapersvil to come up with the good summary of all pros and cons.

Maged: I like 3...

Victor: I agree with Torvald. We should understand the issues but not necessarily write a paper.

Torvald defends why writing a paper is a good idea.

Justin: Should we take this discussion to the reflector?

Victor: An action item should be for Michael Wong to write a paragraph or two about this issue

Tatiana: I support the suggestion to write the paper.

Justin gives summary of action items from this meeting.

Action Items:

- Michael W will send two papers – on library and core words
- Everybody would send the feedback on these papers
- Michael W to mention the empty transaction issue in the core paper and everybody to add to it before Friday paper deadline.

2.2 Final review of core wording issues (needed for June 2 meeting)

3. Any other business

4. Review

4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]

4.2 Review action items (5 min)

5. Closing process

5.1 Establish next agenda

5.2 Future meetings: June 2: CWG review

May 26: Canceled due to Memorial day

June 2: CWG review. 2 hours, 3pm-5pm. Michael Wong and Jens are required to attend. Everybody else is optional but welcome.

Jun 9: Regular meeting. Discuss extern template issue with Eric.

Jun 16: C++ Standard meeting Rapperswil

Adjourned.