Document Number: P0359R0
Date: 2016-05-30
Authors: Michael Wong
Project: Programming Language C++, SG5 Transactional Memory
Reply to: Michael Wong <michael@codeplay.com>

SG5: Transactional Memory (TM) Meeting Minutes 2016/02/22-2016/05/23

Contents

Minutes for 2016/02/22 SG5 Conference Call

Meeting minutes by Michael Scott

The current secretary rota list is (the person who took notes at the last meeting is moved to the end):
Torvald, Tatiana, Jens Maurer, Victor, Maged, Mike Spear,
Hans, Michael Wong, Michael Scott

Agenda:

1. Opening and introductions

1.1 Roll call of participants
Hans Boehm, Brett Hall, Victor Luchangco, Maged Michael,
Michael Scott, Mike Spear, Michael Wong

1.2 Adopt agenda
done

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

1.4 Review action items from previous meeting (5 min)
Michael Spear+Victor to work on getting something to paper on in_transaction. Michael Scott will help.

no progress

2. Main issues (50 min)

2.1 Review feedback on whether to propose TS for C++17
Group's feeling is no due to needing more user feedback, waiting for when GCC 6 starts to support it.
Unless others on the committee register a strong interest.
Look at any feedbacks.

Michael W. sent letter asking if there was sentiment in favor of including in C++'17. Did not mean to imply that we wanted to include it, but a few people interpreted it that way. Canadian group was ok with inclusion, though not pushing it (though actually opposing a variety of other proposals).

Subsequently issued doc explicitly indicating intent not to include
in C++'17.

2.2 Discuss in_transaction proposal from Dec 14 call
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/fn73ZddwmoA

Pending action item.

2.3 Review defects and fixes
Michael W. still plans to add defect tracking on GitHub.
So far we have
replace terminate with abort
add transaction cancellation to list of allowed copy elision
JM: We also need "fix type checking for functions
annotated with transaction_safe".

Background: Since the TM TS got issued, a lot of work was expended
for "Make exception specifications be part of the type system"
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/p0012r1.html .

This touches essentially the same spots as the transaction_safe
annotation for functions, but it had a lot more scrutiny.
We need to fix our TM TS accordingly (i.e. check we touch
the same places).
2.4 Retry discussion
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib__PFfc
Mike Spear: we have suggested that in_transaction() be useful for
txn_defer(). Might immediate_abort_and_retry() be similarly useful
for (true) retry()?

Do we need/want to embed speculation in the semantics?

Michael Scott: kind of prefer not to, in the abstract, but we may
have to in C++, since so much of the language is operational.
Hans: but not entirely so; the memory model is declarative...
Victor: escape actions also suggest we need
explicit speculation as well.
Mike Spear: do exceptions force explicit speculation?
Victor: don't think so.
Mike Spear: but then if we have in_transaction(), atomic_cancel, and
surrounding loops, retry pretty much syntactic sugar.
Michael Scott: that doesn't work super well with nesting. More
important, retry gives the compiler the option to track read
sets and do the right thing.

Hans: agree, but if you put retry in you encourage people to use it, and there may be usage traps that will lead people to write bad programs.
Michael Scott: but we need _something_ for condition synchronization. If not this, what?
Mike Spear: have to figure out what "synchronized { retry" means, or "synchronized { launch_missiles(); atomic { retry"...
Hans: agree we should work out the best semantics we can, then talk about whether we want it. Leans toward closed nesting.
Mike Spear: risk finding ourselves back with "may_cancel_outer" or its equivalent in the type system.

Somebody needs to take the lead on this...

3. Any other business

NB: commit action paper has published. Thanks, Victor.
Probably won't get vetted in Jacksonville.
Probably won't be any significant SG5 action, in fact.
Michael Wong will be there;

4. Review

4.1 Review and approve resolutions and issues (changes to SG's working draft)

N4513 is the official working draft (these links may not be active yet until ISO posts these documents)
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4513.pdf

N4514 is the published PDTS:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.pdf

N4515 is the Editor's report:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.html
The repository (post-Leneaxa) is on Github:
https://github.com/cplusplus/transactional-memory-ts

4.2 Review action items (5 min)
in_transaction paper
possible start on retry

5. Closing process

5.1 Establish next agenda

5.2 Future meeting

Next call : post Jacksonville meeting TBD
(second or third week of March)

Past and future agendas

Nov 2: Post Kona C++/C meeting reviews
Nov 16: Michael away at SC 15
Nov 30: Discuss std::terminate vs std::abort and on_commit proposal or retry
Dec 14: Michael away: continue with on_commit
Jan 11: Michael away
Jan 25: continue with on_commit, also in_transaction
Feb 8: Michael away at OpenMP meeting; review on_commit;
Feb 22: in_transaction; retry
Feb 29: C++ Std Meeting Jacksonville

Minutes for 2016/04/04 SG5 Conference Call

Meeting minutes by Victor
Secretary Rota: Torvald, Tatiana, Jens Maurer, Maged, Mike Spear, Hans, Michael Wong, Michael Scott, Victor

Attendees: Hans, Michael Wong, Mike Spear, Victor

Adopted agenda

REVIEW ACTION ITEMS

1. Michael Spear+Victor to work on getting something to paper on in_transaction. Michael Scott will help.

Some discussion about this:
transaction_wrap is in gcc, which has some relevance to this topic.
Mike has incomplete draft that he will send to Victor, with three alternatives.
They will work on it and send to the rest of the committee.

2. Michael W to file 2 bugzillas for TS
https://issues.isocpp.org/describecomponents.cgi

Still pending.

MAIN ISSUES

1. Report on C++17 progress

We didn’t propose to include TM in C++17, which was good because there was lots of controversy on other proposed additions.
Here are the status updates on some relevant outstanding proposals:

(Most of the) Parallelism TS is being incorporated into C++17
Concepts was proposed to be included but rejected as not yet ready.
Concurrency TS was not proposed for inclusion in C++17
Coroutine was approved for a TS (but not proposed for inclusion in C++17)

The inclusion of the parallelism TS into C++17 may require updates to our TS to specify that some new algorithms must be transaction-safe.

AI (Michael Wong): Add “defect” to bug database to check what changes are needed due to new parallel algorithms (and possibly other new library functions).

Mike Spear: If TM becomes part of C++, should we provide guidance about what is needed for new features to be compatible.
Michael Wong: Good idea

AI (group): Write up white paper with such guidance.

2. in_transaction proposal
(https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/fn73ZddwmoA)

Tabled pending paper on this (open AI for Mike Spear and Victor)

3. Review defects and fixes at https://issues.isocpp.org/describecomponents.cgi
  3.1. Replace terminate with abort
  3.2. Add transaction cancellation to list of allowed copy elision

Tabled pending the write-up of these defects (open AI for Michael Wong)

4. retry
(https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib__PFfc)

Mike Spear: We should have some experience to justify the inclusion of explicit retry mechanism. We can sort of simulate it with transaction cancelation, and that may be adequate for now.

(Some discussion about this followed, but with the same conclusion.)

Hans: It might be useful to flesh out a proposal, so that we can experiment it.
Mike S: We have an implementation in gcc TM that will be presented at Eurosys.

5. Other business

Michael Wong: We should consider pushing a Clang implementation.
Mike Spear: I may have a student who might work on this.

ACTION ITEMS

File defects and fixes at https://issues.isocpp.org/describecomponents.cgi (Michael Wong)
- Replace terminate with abort
- Add transaction cancellation to list of allowed copy elision
- Check for changes needed due to new parallel algorithms being added to C++17

Write up in_transaction proposal (Mike Spear and Victor, with help from Michael Scott)

Write up guidance for TM compatibility for when TM is included in C++ standard (SG5)

NEXT MEETING: April 18 (Victor to send out agenda and concall info)

Minutes for 2016/04/18 SG5 Conference Call

Minutes by Jens

2016-04-18 19:00 UTC - 20:02 UTC
SG5 Transactional Memory

Notes by Jens Maurer

1.1 Roll call of participants
Attendees: Victor Luchangco, Jens Maurer, Mike Spear, Michael Scott,
Maged Michael

1.2 Adopt agenda

No objections.

1.3 Approve minutes

No objections.

1.4 Review action items from previous meeting
Victor: Michael Wong has left IBM and joins Codeplay; see
https://www.codeplay.com/portal/michael-wong-joins-codeplay .

1.4.1 File defects and fixes at https://issues.isocpp.org/describecomponents.cgi (Michael Wong)
 - Replace terminate with abort
 - Add transaction cancellation to list of allowed copy elision
 - Check for changes needed due to new parallel algorithms being added to C++17
Jens: add "rules for function pointer conversions are not quite correct"

Michael is not present during this call, so postpone disucssion.

1.4.2 Write up in_transaction proposal (Mike Spear and Victor, with help from Michael Scott)
Victor sent it around a few minutes before this call.

1.4.3 Write up guidance for TM compatibility for when TM is included in C++ standard (SG5)
Victor: Create notes for people how to change their code to be compatible with TM.

2.1 Review in_transaction proposal

Victor: Four proposals as a basis for discussion. Review section 3.

Jens: See "constexpr if" paper; that might be related.

Victor: Option 2 does not seem viable.

Jens: My model is that two variants of a transaction_safe function are produced: One that is called from outside of a transaction and one that is called from inside of a transaction.

Victor: That matches my model.

Victor: For the second option, how do we satisfy the static type-checking of transaction-safe function?

Mike Spear: Runtime computation for in_transaction() might not be compile-time computable.

Victor: One feature is the ability to determine whether we're inside a transaction, and a different feature is the ability to have unsafe code appear in a transaction-safe function.

Michael Scott: std::in_transaction() must be the only part of the condition of the "if".

Mike Spear: There is a runtime overhead involved in creating lambdas, which puts a burden on the non-transactional codepath.

Victor: Another option is a version with the lambdas.
-> Action item for Mike Spear and Victor.

Mike Spear: With in_transaction mechanism, we might need three object code variants of a transaction-safe function: One that is non-transactional, one that is transactional (different algorithm), but not instrumented (for HTM), one that is transactional (different algorithm), but instrumented (for STM).

Mike Spear: I find transaction_wrap confusing and error-prone, but one nice feature is that you don't have to change the original code; you just add extra code for transaction support.

Jens: Second option is right out, because it uses existing runtime functionality and defies existing semantics of static checking of transaction_safe. First option uses a keyword, which seems over-the-top for a tiny implementation corner. Fourth option (overloading on transaction_safe) seems like the right mental model. C++ programmers are used to overloading and overload resolution depending on argument types; adding another attribute influencing overload resolution seems feasible.

Mike Spear: memcpy_safe annotate with transaction_only for overload resolution purposes.

Victor: We are allowing behavior that is different.

Jens: Yes. If you give your user the feature to write different code for in-transaction vs. outside-transaction, you cannot prevent your user to write code with totally different semantics for in-transaction vs. outside-transaction. Tough luck. That means, in the examples, the HTM case can never call memcpy_asm; it must call memcpy_safe, because the user specified this as the semantics of the in-transaction case.

Mike Spear: If gcc wishes to use an optimized memcpy for HTM, they can do that as an extension internally.

4.2 Action items:

- Show option with lamddas (Mike Spear and Victor)
- Show option with "real" overloading of memcpy instead of transaction_wrap (Victor)

5.2 Next meeting

May 2, 2016

Minutes for 2016/05/02 SG5 Conference Call

Minutes by Mike Spear

On Mon, May 2, 2016 at 12:07 PM, Victor Luchangco <victor.luchangco@oracle.com> wrote:
Hi folks,

Here's the info for the call this afternoon:

Start Time: Monday, 18 Apr 2016, 12:00 PM US Pacific Time (07:00 PM in GMT)
End Time: 1:00 PM US Pacific Time (duration: one hour)

NOTE: New phone numbers (for this meeting only)

Toll-free Dial-in number: +1(866)682-4770
Local US number: +1(408)774-4073
Local Germany number: +49 89 1430 2323
(Let me know if you need a local number for a different country.)

Conference code: 8918244
Security passcode: 1234

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 (the person who took notes at the last meeting is moved to the end)

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

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Victor, Hans, Brett, Michael Scott, Michael Wong, Mike Spear

1.2 Adopt agenda

No additions.

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

Approved.

1.4 Review action items from previous meeting (5 min)

1.4.1. File defects and fixes at https://issues.isocpp.org/describecomponents.cgi (Michael Wong)
- Replace terminate with abort
- Add transaction cancellation to list of allowed copy elision
- Check for changes needed due to new parallel algorithms being added to C++17

No action yet by Michael Wong.

1.4.2. Edit in_transaction proposal (Mike Spear and Victor)
- Show option with lambdas
- Show option with “real” overloading of memcpy (instead of transaction_wrap)

No action yet by Victor.

1.4.3. Write up guidance for TM compatibility for when TM is included in C++ standard (SG5)

Not assigned yet, and this perhaps shouldn't be an action item as much as a "TODO in the future".
--> If such guidance were written up, it could be submitted as an ISO paper.
--> This could be an evolving document, though, which wouldn't be good for an ISO paper.
--> Some clarification about the meaning of this action item. The topic is {when/if TM is incorporated into C++} {what will *new* additions to the C++ standard need to do in order to be compatible with C++TM}

2. Main issues (50 min)

(Before 2.1, we briefly discussed whether we should be taking efforts to broaden participation in SG5. More industry involvement would be nice.)

2.1. Review in_transaction proposal (draft to be sent out separately)

Prior discussion at https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/fn73ZddwmoA

Brief summary of last meeting: there were four different proposals, because the document was intended to spur discussion, not to make a recommendation. The goal is not just to know if within a transaction, but also to allow putting unsafe code statically in the bounds of a transaction while ensuring that it never runs from within a transaction

Two proposed mechanisms used an in_transaction() function, with the second adding an assertion of not-in-transaction. The third mechanism was a new keyword. The fourth was transaction_wrap.

The first mechanism seems to require an undecidable computation by the compiler. The second does not, but it does not encourage good structured programming. The third adds a keyword, which is not particularly appealing. Function overloading (transaction_wrap) is interesting, but the GCC implementation has unclear semantics.

The new proposals were a modification of transaction wrap, which would make the compiler behavior for instrumenting transaction-safe functions a bit cleaner. The other would be a function that takes two lambdas. It calls the first when called transactionally, and the second otherwise.

The differences are few, but matter in two senses: ergonomics (including the need for keywords) and compatibility with C. But note that C++ has been choosing lambda-based solutions at the expense of C compatibility, so this second sense may not be too significant.

Also, Jens had pointed out a relationship to constexpr_if.

2.2. Review defects and fixes (Michael to file a bug on https://issues.isocpp.org/describecomponents.cgi)

Brief discussion of rules for function pointer conversion, which was added by Jens recently. More follow-up between Michael Wong and Jens is needed.

We also need to be sure to start updating Michael Wong's email address on new documents.

2.3. Continue Retry discussion
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib__PFfc

There is a code pattern that can achieve retry, but we do not have syntactic sugar. If we have retry as a keyword, it will probably be used more. Is that a good thing or a bad thing? Also, if there is a keyword, there will be an expectation of efficiency, which may be the key point in whether people use it or not.

Sooner or later, we will need condition synchronization, and there are not viable alternatives to retry. And the kludge won't have good performance.

Specifying it will require us to specify the relationship between retry and synchronized blocks, and the relationship between retry and nontransactional code (i.e., transaction_safe functions).

It's not clear that retry() will only be called from shallow lexical scopes.

On the other hand, if we write up something, it will help us to identify issues that we might not be thinking about right now.

Suggestion: let's put this on the back-burner, on a list of ideas and tasks that we don't want to drop off, but that we aren't actively dealing with right now.

3. Any other business

Who will host SG5 calls? Victor is currently hosting them, which works unless Victor cannot attend the meeting. That will work indefinitely.

Only one meeting before we have to write the in_transaction paper. It must be published by May 30, 9:00 AM EST.

4. Review

4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]
N4513 is the official working draft (these links may not be active yet until ISO posts these documents)
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4513.pdf

N4514 is the published PDTS:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.pdf

N4515 is the Editor's report:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.html

Github is where the latest repository is (I have updated for latest PDTS published draft from post-Leneaxa):
https://github.com/cplusplus/transactional-memory-ts

Bugzilla for filing bugs against TS:
https://issues.isocpp.org/describecomponents.cgi

4.2 Review action items (5 min)

Victor: create "possible future topics" list with Retry and 1.4.3

1.4.2 remains (Michael Wong)

1.4.1 remains (Victor and Mike Spear)

5. Closing process

5.1 Establish next agenda

5.2 Future meeting
Next call: May 16 (Michael Wong in Chicago, but can call in)

May 30: may be cancelled due to Memorial Day (Possibly reschedule to May 23rd)

June 13: last call before C++ meeting.

Past and future Meeting dates
April 4: ARCS in Germany
April 18: Khronos keynote
May 2 : bank holiday UK
May 16: STAC conference
May 30: memorial day US; mailing deadline May 30.
June 13: week before C++ Std meeting Oulu

Minutes for 2016/05/16 SG5 Conference Call

Minutes by Maged Michael

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

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

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

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Victor, Hans, Maged

1.2 Adopt agenda

Adopted

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

Approved

1.4 Review action items from previous meeting (5 min)

1.4.1. File defects and fixes at https://issues.isocpp.org/describecomponents.cgi (Michael Wong)
- Replace terminate with abort

- Add transaction cancellation to list of allowed copy elision
- Check for changes needed due to new parallel algorithms being added to C++17

Carry over

1.4.2. Edit in_transaction proposal (Mike Spear and Victor)
- Show option with lambdas
- Show option with “real” overloading of memcpy (instead of transaction_wrap)

Victor sent update on 5/16.

2. Main issues (50 min)

2.1. Review in_transaction proposal (draft to be sent out separately)

Prior discussion
at https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/fn73ZddwmoA

Discussion of the updated proposal sent by Victor on 5/16 at 3:03pm EDT

Victor: 6 proposals

(1) Two code blocks runs one if in transaction and the other when not

(2) Boolean function - compiler checks safety

(3) Same as (2) except that the compiler need not check statically for being not in a transaction

(4) Similar to (1) except a function selects between two in-transaction and not-in-transaction functions - No keyword needed

(5) Similar to transaction_wrap that is in gcc

(6) Similar to (5) except using overloading instead of a transaction_wrap keyword

Hans: prefers 4 - least in causing complications

Victor: (2) is most likely to cause complications

Victor: (4) advantage just a library function

Hans: (6) may have complicated overloading resolution

Victor: Might need to visit the issue raised by Mike Spear if something special needs to be done under HW transactions.

Hans: In transaction applies to atomic blocks only or synchronized blocks too?

Victor: both in the current proposal

Maged: We should consider the various motivations because there are varying motivations: want to avoid conflicts, don't want HW transactions to fail, want to make speculation short...

Hans: We should consider whether synchronized blocks are in transactio if executed non-speculatively

AI: Victor continu editing Discussion section

2.2. Review defects and fixes (Michael to file a bug on https://issues.isocpp.org/describecomponents.cgi)

3. Any other business

4. Review

4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]
N4513 is the official working draft (these links may not be active yet until ISO posts these documents)
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4513.pdf

N4514 is the published PDTS:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.pdf

N4515 is the Editor's report:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.html

Github is where the latest repository is (I have updated for latest PDTS published draft from post-Leneaxa):
https://github.com/cplusplus/transactional-memory-ts

Bugzilla for filing bugs against TS:
https://issues.isocpp.org/describecomponents.cgi

4.2 Future backlog discussions:

4.2.1 Write up guidance for TM compatibility for when TM is included in C++ standard (SG5)

4.2.2 Continue Retry discussion
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib__PFfc

https://groups.google.com/a/isocpp.org/forum/#!topic/tm/7JsuXIH4Z_A

4.3 Review action items (5 min)

AI: Victor continu editing Discussion section

5. Closing process

5.1 Establish next agenda

5.2 Future meeting
Next call: May 23

Past and future Meeting dates
April 4: ARCS in Germany : DONE
April 18: Khronos keynote: DONE
May 2 : bank holiday UK: DONE
May 16: STAC conference chicago:

May 23: move May 40 meeting to 23
May 30: memorial day US; mailing deadline May 30.
June 13: week before C++ Std meeting Oulu

Minutes for 2016/05/23 SG5 Conference Call

Minutes by Hans Boehm
Attendees: Michael Spear, Michael Wong, Victor Luchangco, Maged Michael, Hans Boehm

Discussion of in_transaction proposal

Michael Wong: Need to narrow down options.

Michael Spear: Favor 4 (select) and 6 (linker)

Michael Spear and Michael Wong:
How comfortable are we in talking about weak linkage etc. in the standard?
Not a notion in the standard.
Can we disambiguate with just name mangling? Need to provide transaction-safe library code when the default implementation one is unsafe.
Want to be able to implicitly use existing code in a way that calls e.g. a transaction-safe memcpy.

This doesn't seem to correspond to traditional overload resolution.

Michael Wong:
C compatibility also favors option 6.

Hans:
WG21 tends to favor clean C++ solutions more than C compatibility. Would be better to have up front commitment from WG14 that they're actually going to care.

Michael Spear:
Verification may be hard due to overridden functions.

Michael Spear:
Transaction_only keyword?

Victor:
Makes sense, but ...

Michael Spear:
What about synchronized blocks?

Victor: Addressed at end of section 4.

M. Spear:
What if we call an unsafe function inside a synchronized block, which then calls a transaction-safe function. Does option 4 still have this issue?

Victor, Hans:
Should still think about allowing the implementation to choose what to do inside synchronizaed blocks, even for option 4.

Option 5 and 6 require that calls inside synchronized blocks consistently be resolved to transaction-unsafe versions, or we let the implementation decide.

We'll now all work on the Google docs version. Deadline is this Sunday.

Action items:

All: Work on document before Friday. Add some more plusses and minusses.
Split Synchronized block discussion into its own section.

Michael Wong: Get a document number. Send out doocument to reflector.

No meeting until after Finland.

On Fri, May 20, 2016 at 6:41 AM, Michael Wong <fraggamuffin@gmail.com> wrote:
Start Time: Monday, 23 May 2016, 12:00 PM US Pacific Time (07:00 PM in GMT)
End Time: 1:00 PM US Pacific Time (duration: one hour)

NOTE: New phone numbers (for this meeting only)

Toll-free Dial-in number:	+1(866)682-4770
Local US number:	+1(408)774-4073
Local Germany number:	+49 89 1430 2323

(Let me know if you need a local number for a different country.)

Conference code: 8918244
Security passcode: 1234

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 (the person who took notes at the last meeting is moved to the end)

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

Agenda:

1. Opening and introductions

1.1 Roll call of participants

1.2 Adopt agenda

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

1.4 Review action items from previous meeting (5 min)

1.4.1. File defects and fixes at https://issues.isocpp.org/describecomponents.cgi (Michael Wong)
- Replace terminate with abort
- Add transaction cancellation to list of allowed copy elision
- Check for changes needed due to new parallel algorithms being added to C++17

1.4.2. Edit in_transaction proposal (Mike Spear and Victor)
- Show option with lambdas
- Show option with “real” overloading of memcpy (instead of transaction_wrap)

2. Main issues (50 min)

2.1. Review in_transaction proposal (draft to be sent out separately)

Prior discussion at https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/fn73ZddwmoA

3. Any other business

4. Review

4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]
N4513 is the official working draft (these links may not be active yet until ISO posts these documents)
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4513.pdf

N4514 is the published PDTS:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.pdf

N4515 is the Editor's report:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.html

Github is where the latest repository is (I have updated for latest PDTS published draft from post-Leneaxa):
https://github.com/cplusplus/transactional-memory-ts

Bugzilla for filing bugs against TS:
https://issues.isocpp.org/describecomponents.cgi

4.2 Future backlog discussions:

4.2.1 Write up guidance for TM compatibility for when TM is included in C++ standard (SG5)

4.2.2 Continue Retry discussion
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib__PFfc
https://groups.google.com/a/isocpp.org/forum/#!topic/tm/7JsuXIH4Z_A

4.3 Review action items (5 min)

5. Closing process

5.1 Establish next agenda

5.2 Future meeting
Next call: No call June 13. TBD after Oulu meeting.

Past and future Meeting dates
April 4: ARCS in Germany : DONE
April 18: Khronos keynote: DONE
May 2 : bank holiday UK: DONE
May 16: STAC conference chicago:DONE
May 23: move May 40 meeting to 23
May 30: memorial day US; mailing deadline May 30. CANCELLED
June 13: week before C++ Std meeting Oulu