

Document Number: N4441  
Date: 2015-04-10  
Authors: Michael Wong  
Project: Programming Language C++, SG5 Transactional Memory  
Reply to: Michael Wong <michaelw@ca.ibm.com>

## **SG5: Transactional Memory (TM) Meeting Minutes 2015/03/23-2015/04/06**

### **Contents**

Minutes for 2015/03/23 SG5 Conference Call .....	2
Minutes for 2015/04/06 SG5 Conference Call .....	5

## Minutes for 2015/03/23 SG5 Conference Call

Minutes by Maged (This Meeting is after all NB comments closed for the TM TS on March 10, 2015)

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

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Maged, Mike Spear, Jens Maurer, Michael Wong, Michael Scott, Victor

1.2 Adopt agenda

Agreed

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 Action Item: Review NB Comments

1.4.7. All: Consider attending Lenexa Kansas Meeting in May49, 2015

[http://www.openstd.](http://www.openstd.org/jtc1/sc22/wg21/docs/papers/2014/n4059.pdf)

[org/jtc1/sc22/wg21/docs/papers/2014/n4059.pdf](http://www.openstd.org/jtc1/sc22/wg21/docs/papers/2014/n4059.pdf)

This meeting is where we will consider PDTS Ballot resolution and comments, and meet formally as an SG group

to process those comments.

Michael: Ideal time to attend is Tuesday, Wednesday and Thursday. Friday is Mock plenary and will require one

TM expert.

2. Main issues (50 min)

2.1 Discuss new ISO process where DTS is not needed and TM PDTS has passed.

Jens: The TS is officially approved. We still need to respond to the comments. We can change the TS text with

small fixes. Should not make big changes when there is no chance for NBs to review and comment.

2.2 Review NB comments

JP1: Concern about impact of TM support on programs that don't use TM.

Michael

Scott: We can say that we were mindful of this issue and were careful not to create overhead for nonTM

programs

Michael

Spear: Three areas of potential overhead that we can say that TM does not create additional performance degradation. (1) tm safety does not prevent compilers from generating the most optimized version of

nonTM

code (2) nonTM

memory access and (3) nonTM

synchronization.

Jens:

The response to comments should be focused on the specific issues raised in the comments.

Jens:

There will always be code generated for the nonTM case.

Jens:

There might be overhead in the initialization of local static variables. There is already overhead in synchronization. TM implementation may follow the same synchronization mechanism and if at all add minor

overhead in comparison to the conventional overhead of multithreaded local static and does not necessarily add

overhead to the nonTM

path.

JP2: A functionlocal

static variable initialization should be transactionalunsafe.

The initialization in an atomic

execution needs to be synchronized with nonatomic

executions.

Jens: We don't want to do that. The overhead is very low.

Michael Scott: The intent of the comment might be that there is concern that tm safe prevents the use of static

local in nonTM

code.

Maged: My reading of this is that they are concerned that tm safe implies performance overhead. We can address

this by saying that tm safe does not necessarily add performance overhead.

US1: Memory ordering requirements of transactions are problematically strict. Even empty or purely local

transactions have observable synchronization effects and can usually not be removed by an optimizing

compiler. This introduces a performance penalty when transactional library code is reused in a clearly threadlocal

context.

Two formulations:

Hans: "There is a global total order of execution for all outer blocks. If, in that total order, T1 is ordered before T2,

and T1 and T2 perform conflicting operations, then the end of T1 synchronizes with the start of T2."

Mark Batty: "This order on transactions induces an order on memory access such that if T1 is ordered before T2

then all memory accesses in T1 are ordered before those in T2. This induced order on memory accesses is

consistent with happens before."

M Wong: There is preference for Mark Batty's formulation.

M. Scott: Prefers Hans's formulation

Maged: Do both formulations allow the relaxation of Hans's example of a thread creating another thread inside a

synchronized block and the second thread uses an empty synchronized block to await data initialization by the

parent thread.

M. Scott, Victor: Yes

Jens: Mark did work on formal verification of the C++ memory model.

M Scott: We should get Hans's take on Mark's formulation

Jens: Will use Mark's formulation and get Hans's opinion.

Victor: Mark's does not mention "synchronizes with"

M. Wong: Next call April 6. The last meeting before the responses are due by April 10. M. Wong has responses for most comments.

AI: Jens preparing response to the NB comments including JP1

AI: M. Scott prepare response to JP2

3. Any other business

4. Review

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

N4301 is the official working draft

See N4301 and N4302

<http://www.openstd.org/jtc1/sc22/wg21/docs/papers/2014/n4301.pdf>

<http://www.openstd.org/jtc1/sc22/wg21/docs/papers/2014/n4301.pdf>

<http://www.openstd.org/jtc1/sc22/wg21/docs/papers/2014/n4302.pdf>

<http://www.openstd.org/jtc1/sc22/wg21/docs/papers/2014/n4302.pdf>

4.2 Review action items (5 min)

5. Closing process

5.1 Establish next agenda

5.2 Future meeting:

Next call: April 6

Past and future Chat agendas

Dec 1: review Urbana meeting. Done

Dec 15:Chandlers issue continue. Done

Jan 12: Review a possible NB comment; Chandler issue review.

Jan 26: Review Canadian comment; review Hans drafting

Feb 9: review revised Cdn comment, review feedback for Han's rewrite of Chandler's issue.

Cancelled

Feb 23:LWG Cologne meeting, Michael away, Cancelled

Mar 9: Michael away, Cancelled

Mar 23:Review NB comment Done

Apr 6: Review NB comments and make decision on any remaining item (Mailing deadline)

Apr 20: Parallel 2015, Michael away

May 4: C++ Std meeting

## Minutes for 2015/04/06 SG5 Conference Call

Meeting minutes by Hans

April 6 minutes:

Attendees:

Hans Boehm

Victor Luchangco

Jens Maurer

Maged Michael

Michael Scott

Tatiana Shpeisman

Michael Spear

Michael Wong

Skip directly to US1:

Hans: Suspects the current US1 resolution misinterpreted Mark's statement. Was intended as an additional requirement on original statement.

Michael Scott: Why does Mark's statement as addition change anything?

Hans: It may matter because the total order on transactions no longer always contributes to happens-before. Transactions touching disjoint locations are not necessarily ordered by happens-before.

Victor: Why we can't just choose a different total order that is consistent with happens before.

Michael Spear, Victor, Michael Scot, Hans: Discussion of whether the difference is really observable. Unclear, but some of us are more comfortable disallowing clearly bogus transaction orders, especially in the context of memory\_order\_consume subtleties.

Jens proposed alternate wording:

"There is a global total order of execution for all outer blocks. If, in that total order, T1 is ordered before T2, and T1 and T2 perform conflicting expression evaluations, then the end of T1 synchronizes with the start of T2. This order induces an order on memory accesses (including modifications) such that if outer block T1 is ordered before

outer block T2, then all memory accesses in T1 are ordered before those in T2. This induced order on memory accesses is consistent with happens-before."

Hans, Michael Scott, Victor, ...: More discussion of wording. Need to clarify that "This order induces" refers to total order. Otherwise the sentence is meaningless.

Lenaxa: Michael, Victor, Hans, and Jens will attend.

Agreement that we don't need to discuss responses other than to US 1.

Michael Scott sent around a revised formulation:

"There is a global total order of execution for all outer blocks. This total order, together with program order, induces a total order on the memory accesses of the outer blocks. The induced order on memory accesses is consistent with happens-before. The total order on transactions `_contributes to_` happens-before, as follows: If, in the total order on transactions, T1 is ordered before T2, and T1 and T2 perform conflicting expression evaluations, then the end of T1 synchronizes with the start of T2. This order induces an order on memory accesses (including modifications) such that if outer block T1 is ordered before outer block T2, then all memory accesses in T1 happen before those in T2."

Consensus that last sentence is not needed and that "induces a total order" is not correct, since sequenced before is not total within a thread.

Michael Scott sends out yet another version:

"There is a global total order of execution for all outer blocks. This total order induces an order on the memory accesses of the outer blocks such that every access in a given block is ordered before every access in each subsequent block. This induced order on memory accesses is consistent with happens-before. The total order on transactions `_contributes to_` happens-before, as follows: If, in the total order on transactions, T1 is ordered before T2, and T1 and T2 perform conflicting expression evaluations, then the end of T1 synchronizes with the start of T2."

Tatiana: Combine second and third sentence.

Victor: Remove "The total order on transactions `_contributes to_` happens-before, as follows:"

General agreement with the last two.

Michael W: Cancel April 20th call

Finalize US 1 text on mailing list. Victor just sent out update:

"There is a global total order of execution for all outer blocks. If, in the total order on outer blocks, T1 is ordered before T2, and T1 and T2 perform conflicting expression evaluations, then the end of T1 synchronizes with the start of T2. The order on memory operations within outer blocks induced by the total order on outer blocks must be consistent with happens-before."

No objections to other fixes on Jens' proposed list.

Next meeting may be as late as June 1.