Document Number: P2244R0
Date: 2020-10-25
Authors: Michael Wong
Project: Programming Language C++, SG14 Games Dev/Low Latency/Financial Trading/Banking/Simulation/Embedded
Reply to: Michael Wong <michael@codeplay.com>

SG14: Low Latency/Games Dev/Embedded/Financial Trading/Banking/Simulation Meeting Minutes 2020/04/08-2020/08/21

Contents

Minutes for 2020/04/08 SG14 Conference Call 2
Minutes for 2020/05/13 SG14 Conference Call 10
Minutes for 2020/06/11 SG14 Conference Call 19
Minutes for 2020/07/08 SG14 Conference Call 24
Minutes for 2020/08/12 SG14 Conference Call 28

Minutes for 2020/04/08 SG14 Conference Call

> 1.1 Roll call of participants
>
Derek Haines
Henry Miller
James renwick
Andrews Weis
Antony Peacock
ben Craig, Ben Saks
BillyBaker
Connor Horman
Max Gardner
Patrice Roy
Paul Bendien
Sophia Poirier
Staffan Tjernstrom
Steven Varga
Jens Maurer
Inbal Levi
Michael Wong
Dan Raviv?
Charles Bay
Matthew Butler

1.2 Adopt agenda
>
Approve.

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

>
> 1.4 Action items from previous meetings
>
> 2. Main issues (125 min)
>
> 2.1 General logistics
>
> Prague summary
>
> Mailing deadline is monthly on 15th

>
> All meetings going online.
>
> 2.2 Paper reviews
>
>
> Low-cost Deterministic C++ Exceptions for Embedded Systems
>
<https://www.research.ed.ac.uk/portal/files/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf> Low-cost
> Deterministic C++ Exceptions for Embedded Systems James Renwick James
> Renwick et al.
>
BC: passing an address, so you can indirect jmp instead of return?
leave optimzation off for now, what it this just look like normal code

when propagating exception up the call stack, just do a return
BC: regular processor, branch cost is small, but on gpu divergign control
throws is high, throwing an exception on gpu is already expensive
for gpu instead of terminate might do a HW specific thing
CH: too many banches will cross bank boundaries;
Even with a nave implementaion, we can see n times better

ST: precompield external libraries 1K allocated for stack and with only 256
bytes from the stack, what happens? Right we dont deal with dynamic library
yet, but it doesnt make teh exception object unique, and nothing is unique
on the exception object here
DLL and static libraries crossing boundary may be UB here or an ODR
violation

JM: table reduction does not match waht you said about text increase
maching?
because unwind library is gone

BC: version of clang?
clang 6

CH:on function size vs binary size;
should the catch block, which can be called out to another function,

buffer on where to put EH object and it falls back to the heap, so this
can do the same thing
need to inject parameter, which could thow., need to distinguisj C and C++
function so need throw specifier added, abi change
or pass by TLS

a C language linkage, dont inject, standards says thata fn pointer inside a structure,
type of ptr
extern C allowed to throw, so we can detect what linkage a fn has
C adapting deterministic exception
we just need implmentation with a compielr flag
parameter injection will be an abi break

exception_ptr was intended to allow copying to support bot hreference counting heap allocated as is eager copy implementations

you know the size because you call the copy constructor and not just memcpy it
resolve copy construction when you write the throw expression so you know the size of the xception object
but if pass exception ptr through to another functions object, that could take a dynamic size,
if you can pass exception ptr around, can assign it to global variable,
stack space does not matter, you have a life time problem; so it has to be on the heap to solve this
so if eh object lives on stack, cant touch the heap, the nwe have a problem; yes you can outlive, so copy an exception ptr object stackframe,
then for these reasons
exception_ptr is aimed at heap implementation
so we propose similar that is agnostic, encapsulate exception object, call get_current exception, then you are responsible for this object, no reference count, so you move it around

cross noexcept bounday with an exception ptr, want to pass exeption through noexcept boundary without passing exception object as parameter for the rethrowing case

want std exception ptr that has shared ownership and unique ownership of the exception object

no language or library changes in this paper
its an implementation report
should we advise exception ptr is toxic

Paul: claims is standard conformant, leaves out what noexcept specifier mean

Ben: it would no changes, there are proposal would be changing stuff substantially, address cost benefit analysis of those papers

some of the mitigation: adding noexcept blocks to mark large parts as noexcept, change the calling convention to make it so a prebuilt C library can work with this

PB: C assert is verymuch a hint of adoption

Exception_ptr: can we ban it in free_standing, or use Rust send

exception_ptr on the heap is fine, problem is when you move it, it is nolonger available in the catch block, if you copy exception its fine as implementation only allow move only objects

if it is moved onto the heap, then its a problem, but OK if you copy from stack to heap

open the github (limited form)and form a group and tthen come back to discuss bridfing mechanism

please reply to this thread if inetersted in joining this Exception Handing for Embedded implementation group.

P2057r0
> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2057r0.pdf> SG14
> SG19 Past, Present and Future status Michael Wong Michael Wong et al.
>
>
PB: numeric proposal an ommission?Yes please add

> 2.2.1 any other proposal for reviews?
>
MW: Interested in Networking RPC proposal? Yes
ST: Error handling compendium?

> 2.3 Domain-specific discussions
>
> 2.3.1 SIG chairs
>
> - Embedded Programming chairs: Ben Craig, Wouter van Ooijen and Odin
> Holmes, John McFarlane
> <
>
>
http://wiki.edg.com/bin/edit/Wg21belfast/McFarlane?topicparent=Wg21belfast.SG14CPPCON2019-09-17;nowysiwyg=1>
>
>

> - Financial/Trading chairs: Stephan TJ, Carl Cooke, Neal Horlock,
> Mateusz Pusz, Clay Trychta,
> - Games chairs: Rene Riviera, Guy Davidson and Paul Hampson
> - Linear Algebra chairs: Bob Steagall, Mark Hoemmen, Guy Davidson
>
> 2.4 Other Papers and proposals
>
> 2.5 Future F2F meetings:
>
> 2.6 future C++ Standard meetings:
> https://isocpp.org/std/meetings-and-participation/upcoming-meetings
>
> - 2020-11: (New York, tentative)
> - 2021-02-22 to 27: Kona, HI, USA
>
> 3. Any other business
> Reflector
> https://lists.isocpp.org/mailman/listinfo.cgi/sg14
> As well as look through papers marked "SG14" in recent standards committee
> paper mailings:
> http://open-std.org/jtc1/sc22/wg21/docs/papers/2015/
> http://open-std.org/jtc1/sc22/wg21/docs/papers/2016/
>
> Code and proposal Staging area
> https://github.com/WG21-SG14/SG14
> 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 meeting
>
> Time: May 13, 2020 02:00 PM Eastern Time (US and Canada)
>
> Time: June 10, 2020 02:00 PM Eastern Time (US and Canada)
>
> Time: Jul 8, 2020 02:00 PM Eastern Time (US and Canada)
>
> Time: Aug 12, 2020 02:00 PM Eastern Time (US and Canada)
>

> Time: Sep 9, 2020 02:00 PM Eastern Time (US and Canada)
>
> Time: Oct 14, 2020 02:00 PM Eastern Time (US and Canada)

Minutes for 2020/05/13 SG14 Conference Call

Michael's notes

>> 1.1 Roll call of participants
>>
> Adam getchell, Guy Davidson, Ben Craig, Ben Saks, Charles Bay, COnor Horman, Henry Miller, Inbal Levi, John McFarlane, Malte Kiessling, Paul Bendien, Matthew Butler, Rene iviera, Ronan Keryell, Ronen Friedman, Sophia Poirer, Billy Baker, Michael Wong, Jens Maurer.

1.2 Adopt agenda
>>
>> 1.3 Approve minutes from previous meeting, and approve publishing
>> previously approved minutes to ISOCPP.org
>>
>> 1.4 Action items from previous meetings
>>
>> 2. Main issues (125 min)
>>
>> 2.1 General logistics
>>
>> Discuss inverted meeting for June
>>
>
2 AM time ET, UTC 6 AM for Asia/Australia

> Mailing deadline is monthly on 15th
>>
>> All meetings going online.
>>
> End of August

> 2.2 Paper reviews
>>
>> P2057r0
>> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2057r0.pdf>
>> SG14
>> SG19 Past, Present and Future status Michael Wong Michael Wong et al.
>>
>> D2057R1:
>>
>>

>> https://docs.google.com/document/d/1LmBo46a7meB-QH-ZxLbxQMinAaJMvKOjR4gssuaEL18/edit#
>>
>> Herb's Assumptions paper:
>>
>> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2064r0.pdf
>>
>
EWG in Prague:
assumptions and assertions are different things
SG12 safety critical related, high performance and low latency, HPT
how to ensure correctness in code
related to contracts
a core of contracts violation, UB which causes a lot of issues, assert if it a bug or not, and it is an implementation matter
would then like a program to trap through the testing of an expression, then optiming based on those assertions
leaving it as an implementation defined, specified then you can have the best of both worlds
fast vs safe code is not the only way
teh predicate being tested in a performnat vs safety program is actually the same

BC: assert cause different things happen vs no assert, nullchecks, if fn deref ptr, and is without asserts then there is no warning
as soon as you had an if looking for null,, but if you have an assert, it would not check your fact,
clang UB sanitizer builtin_unreachable will trap for you
so static analyzers are missing a trick

RF: assert for safety and performance is different, would liek clear definiteion of the attack vectors, distinction between assume to be true and never checked, vs those that are checked

do you want a variety? places where you want to trap bugs vs optimize, especially contracts trying to please everyone
JM: I want something very declarative, to say in std a bug has occurred here and take a step back, and leave it to implementer to do trap, terminate or exception,

RF: multiple compilers environment can be handled that way
JM: could that be likely/unlikely?
want to trap on these things

IL: using assume when you have more information then the compiler, having this in the standard as a compiler extension

can do now with impl specific, cant advertise assert as a contract violation, makes it hard for compiler to optimize, makes it hard for reader to know what they can and cannot do with that function, benefit of contract is put the check on the outside of the function

compiler option or standardize? go for simple solution in contracts

invite Herb onto the call
feedback on contracts

2.2.1 any other proposal for reviews?
>>
>> Discuss future direction of detreministic C++ exception group :
>>
>> Low-cost Deterministic C++ Exceptions for Embedded Systems
>> <
>>
https://www.research.ed.ac.uk/portal/files/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf>
>>
>> Low-cost
>> Deterministic C++ Exceptions for Embedded Systems James Renwick James
>> Renwick et al.
>>
>> 2.3 Domain-specific discussions
>>
>> 2.3.1 SIG chairs
>>
>

Time: June 10, 2020 02:00 PM Eastern Time (US and Canada) : Games: Guy

Time: Jul 8, 2020 02:00 PM Eastern Time (US and Canada) : Embedded: Ben Craig

Time: Aug 12, 2020 02:00 PM Eastern Time (US and Canada) : Fiinance: Jens, Stephan

Time: Sep 9, 2020 02:00 PM Eastern Time (US and Canada) : Security: Mathew Butler

Time: Oct 14, 2020 02:00 PM Eastern Time (US and Canada): Games: Rene

> - Embedded Programming chairs: Ben Craig, Wouter van Ooijen and Odin
>> Holmes, John McFarlane
>> <

>>
>>
http://wiki.edg.com/bin/edit/Wg21belfast/McFarlane?topicparent=Wg21belfast.SG14CPPCON2019-09-17;nowysiwyg=1>
>>
>>
>> - Financial/Trading chairs: Stephan TJ, Carl Cooke, Neal Horlock,
>> Mateusz Pusz, Clay Trychta,
>> - Games chairs: Rene Riviera, Guy Davidson and Paul Hampson
>>
>
apprehension, templates,compilation sensitive

> - Linear Algebra chairs: Bob Steagall, Mark Hoemmen, Guy Davidson
>>
>> 2.4 Other Papers and proposals
>>
>> 2.5 Future F2F meetings:
>>
>> 2.6 future C++ Standard meetings:
>> https://isocpp.org/std/meetings-and-participation/upcoming-meetings
>>
>> - 2020-11: (New York, tentative)
>> - 2021-02-22 to 27: Kona, HI, USA
>>
>> 3. Any other business
>> Reflector
>> https://lists.isocpp.org/mailman/listinfo.cgi/sg14
>> As well as look through papers marked "SG14" in recent standards
>> committee
>> paper mailings:
>> http://open-std.org/jtc1/sc22/wg21/docs/papers/2015/
>> http://open-std.org/jtc1/sc22/wg21/docs/papers/2016/
>>
>> Code and proposal Staging area
>> https://github.com/WG21-SG14/SG14
>> 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 meeting
>>
>> Time: June 10, 2020 02:00 PM Eastern Time (US and Canada)
>>
>> Time: Jul 8, 2020 02:00 PM Eastern Time (US and Canada)
>>
>> Time: Aug 12, 2020 02:00 PM Eastern Time (US and Canada)
>>
>> Time: Sep 9, 2020 02:00 PM Eastern Time (US and Canada)
>>
>> Time: Oct 14, 2020 02:00 PM Eastern Time (US and Canada)
>>
>> On Wed, May 13, 2020 at 1:51 PM Michael Wong <fraggamuffin_at_[hidden]>
>> wrote:
>>
>>> I'm interested. Thanks John, we are light on agenda this month so I am
>>> adding this to the discussion. Thanks.
>>>
>>> On Wed, May 13, 2020 at 1:46 PM John McFarlane <john_at_[hidden]>
>>> wrote:
>>>
>>>> It's a little late -- both for this meeting and -- because it came out
>>>> two mailings ago but is anyone interested in a brief discussion of Herb's
>>>> P2064R0
>>>> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2064r0.pdf> Assumptions?
>>>> More generally, does anyone on SG14 think that contracts are a topic of
>>>> interest for our domains?
>>>> John
>>>>
>>>>
>>>>
>>>> On Wed, 13 May 2020 at 18:17, Michael Wong via SG14 <
>>>> sg14_at_[hidden]> wrote:
>>>>
>>>>> Good point. I will in future. This should be 1800 UTC. Thanks.
>>>>>
>>>>> On Wed, May 13, 2020 at 12:54 PM Guy Cpp via SG14 <
>>>>> sg14_at_[hidden]> wrote:
>>>>>
>>>>>> Small nit: would it be possible to list times in UTC rather than
>>>>>> locally adjusted times? I have a calendar appointment that says 18:00 BST
>>>>>> but I'm pretty sure it's 19:00 BST, ET+5. UTC would clear this all up (I
>>>>>> assume it's an old recurring calendar appointment).
>>>>>>
>>>>>> See you at... 19:00? 18:00 UTC?

>>>>>>
>>>>>> Cheers,
>>>>>> G
>>>>>>
>>>>>> On Wed, 13 May 2020 at 14:55, Michael Wong via SG14 <
>>>>>> sg14_at_[hidden]> wrote:
>>>>>>
>>>>>>> Topic: SG14 Low Latency Monthly. Hi all, I have turned on high
>>>>>>> security for
>>>>>>> this call in light of recent events.
>>>>>>>
>>>>>>> Hi,
>>>>>>>
>>>>>>> Michael Wong is inviting you to a scheduled Zoom meeting.
>>>>>>>
>>>>>>> Topic: SG14 monthly Apr 2020-Oct 2020
>>>>>>> Time: Apr 8, 2020 02:00 PM Eastern Time (US and Canada)
>>>>>>>
>>>>>>> Time: May 13, 2020 02:00 PM Eastern Time (US and Canada)
>>>>>>>
>>>>>>> Time: June 10, 2020 02:00 PM Eastern Time (US and Canada)
>>>>>>>
>>>>>>> Time: Jul 8, 2020 02:00 PM Eastern Time (US and Canada)
>>>>>>>
>>>>>>> Time: Aug 12, 2020 02:00 PM Eastern Time (US and Canada)
>>>>>>>
>>>>>>> Time: Sep 9, 2020 02:00 PM Eastern Time (US and Canada)
>>>>>>>
>>>>>>> Time: Oct 14, 2020 02:00 PM Eastern Time (US and Canada)
>>>>>>>
>>>>>>> Join from PC, Mac, Linux, iOS or Android:
>>>>>>> https://iso.zoom.us/j/819108882?pwd=L0pPZm5QRWlTYXdJditvU1JLdjFYUT09
>>>>>>> Password: 013549
>>>>>>>
>>>>>>> Or iPhone one-tap :
>>>>>>> US: +13462487799,,819108882# or +14086380968,,819108882#
>>>>>>> Or Telephone:
>>>>>>> Dial(for higher quality, dial a number based on your current
>>>>>>> location):
>>>>>>> US: +1 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or
>>>>>>> +1
>>>>>>> 669 900 6833 or +1 253 215 8782 or +1 301 715 8592 or +1 312 626
>>>>>>> 6799
>>>>>>> or 877 853 5247 (Toll Free)
>>>>>>> Meeting ID: 819 108 882
>>>>>>> Password: 013549

>>>>>>> International numbers available:
>>>>>>> https://iso.zoom.us/u/abhaIjFKLZ
>>>>>>>
>>>>>>> Or Skype for Business (Lync):
>>>>>>> https://iso.zoom.us/skype/819108882
>>>>>>>
>>>>>>> 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 to ISOCPP.org
>>>>>>>
>>>>>>> 1.4 Action items from previous meetings
>>>>>>>
>>>>>>> 2. Main issues (125 min)
>>>>>>>
>>>>>>> 2.1 General logistics
>>>>>>>
>>>>>>> Discuss inverted meeting for June
>>>>>>>
>>>>>>> Mailing deadline is monthly on 15th
>>>>>>>
>>>>>>> All meetings going online.
>>>>>>>
>>>>>>> 2.2 Paper reviews
>>>>>>>
>>>>>>> P2057r0
>>>>>>> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2057r0.pdf>
>>>>>>> SG14
>>>>>>> SG19 Past, Present and Future status Michael Wong Michael Wong et
>>>>>>> al.
>>>>>>>
>>>>>>> 2.2.1 any other proposal for reviews?
>>>>>>>
>>>>>>> Discuss future direction of detreministic C++ exception group :
>>>>>>>
>>>>>>> Low-cost Deterministic C++ Exceptions for Embedded Systems
>>>>>>> <
>>>>>>>
https://www.research.ed.ac.uk/portal/files/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf>

>>>>>>>
>>>>>>> Low-cost
>>>>>>> Deterministic C++ Exceptions for Embedded Systems James Renwick
>>>>>>> James
>>>>>>> Renwick et al.
>>>>>>>
>>>>>>> 2.3 Domain-specific discussions
>>>>>>>
>>>>>>> 2.3.1 SIG chairs
>>>>>>>
>>>>>>> - Embedded Programming chairs: Ben Craig, Wouter van Ooijen and
>>>>>>> Odin
>>>>>>> Holmes, John McFarlane
>>>>>>> <
>>>>>>>
>>>>>>>
http://wiki.edg.com/bin/edit/Wg21belfast/McFarlane?topicparent=Wg21belfast.SG14CPPCON2019-09-17;nowysiwyg=1>
>>>>>>>
>>>>>>>
>>>>>>> - Financial/Trading chairs: Stephan TJ, Carl Cooke, Neal Horlock,
>>>>>>> Mateusz Pusz, Clay Trychta,
>>>>>>> - Games chairs: Rene Riviera, Guy Davidson and Paul Hampson
>>>>>>> - Linear Algebra chairs: Bob Steagall, Mark Hoemmen, Guy Davidson
>>>>>>>
>>>>>>> 2.4 Other Papers and proposals
>>>>>>>
>>>>>>> 2.5 Future F2F meetings:
>>>>>>>
>>>>>>> 2.6 future C++ Standard meetings:
>>>>>>> https://isocpp.org/std/meetings-and-participation/upcoming-meetings
>>>>>>>
>>>>>>> - 2020-11: (New York, tentative)
>>>>>>> - 2021-02-22 to 27: Kona, HI, USA
>>>>>>>
>>>>>>> 3. Any other business
>>>>>>> Reflector
>>>>>>> https://lists.isocpp.org/mailman/listinfo.cgi/sg14
>>>>>>> As well as look through papers marked "SG14" in recent standards
>>>>>>> committee
>>>>>>> paper mailings:
>>>>>>> http://open-std.org/jtc1/sc22/wg21/docs/papers/2015/
>>>>>>> http://open-std.org/jtc1/sc22/wg21/docs/papers/2016/
>>>>>>>
>>>>>>> Code and proposal Staging area
>>>>>>> https://github.com/WG21-SG14/SG14

>>>>>>> 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 meeting
>>>>>>>
>>>>>>> Time: June 10, 2020 02:00 PM Eastern Time (US and Canada)
>>>>>>>
>>>>>>> Time: Jul 8, 2020 02:00 PM Eastern Time (US and Canada)
>>>>>>>
>>>>>>> Time: Aug 12, 2020 02:00 PM Eastern Time (US and Canada)
>>>>>>>
>>>>>>> Time: Sep 9, 2020 02:00 PM Eastern Time (US and Canada)
>>>>>>>
>>>>>>> Time: Oct 14, 2020 02:00 PM Eastern Time (US and Canada)

Minutes for 2020/06/11 SG14 Conference Call

Topic: SG14 Low Latency Monthly.

This meeting is Games focused and is chaired by Guy Davidson.

Hi,

Michael Wong is inviting you to a scheduled Zoom meeting.

Topic: SG14 monthly Apr 2020-Oct 2020

Time: Jun 11, 2020 02:00 AM Eastern Time (US and Canada) 6 UTC
Jun 11, 2020 02:00 AM 6UTC
Jul 8, 2020 02:00 PM 18 UTC
Aug 12, 2020 02:00 PM 18 UTC
Sep 9, 2020 02:00 PM 18 UTC
Oct 7, 2020 02:00 PM 18 UTC

Join from PC, Mac, Linux, iOS or Android:
https://iso.zoom.us/j/819108882?pwd=L0pPZm5QRWlTYXdJditvU1JLdjFYUT09
Password: 013549

Or iPhone one-tap :
US: +13462487799,,819108882# or +14086380968,,819108882#
Or Telephone:
Dial(for higher quality, dial a number based on your current location):
US: +1 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or 877 853 5247 (Toll Free)
Meeting ID: 819 108 882
Password: 013549
International numbers available: https://iso.zoom.us/u/abhaIjFKLZ

Or Skype for Business (Lync):
https://iso.zoom.us/skype/819108882

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 to ISOCPP.org

1.4 Action items from previous meetings

2. Main issue

s (125 min)

2.1 General logistics
CPPCON logistics
CPPCON SG14 plans
CPPCON Embedded Track: Ben Saks
future C++ Standard meetings:

2.2 Paper reviews

1. Colony by Matthew Bentley,
Colony Matthew Bentley https://wg21.link/P0447

2. Affinity by Gordon Brown al:

Affinity Gordon Brown, Ruyman Reyes, Michael Wong, Mark Hoemmen, Jeff
Hammond, Tom Scogland, Domagoj Å ariÄ‡ https://wg21.link/p1436

2.2.1 any other proposal for reviews?

Discuss future direction of detreministic C++ exception group :

Low-cost Deterministic C++ Exceptions for Embedded Systems
<
https://www.research.ed.ac.uk/portal/files/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf>

Low-cost
Deterministic C++ Exceptions for Embedded Systems James Renwick James
Renwick et al.

2.3 Domain-specific discussions

2.3.1 SIG chairs

- Embedded Programming chairs: Ben Craig, Wouter van Ooijen and Odin
Holmes, John McFarlane

<
http://wiki.edg.com/bin/edit/Wg21belfast/McFarlane?topicparent=Wg21belfast.SG14CPPCON2019-09-17;nowysiwyg=1>

- Financial/Trading chairs: Stephan TJ, Carl Cooke, Neal Horlock, Mateusz Pusz, Clay Trychta,
- Games chairs: Rene Riviera, Guy Davidson and Paul Hampson
- Linear Algebra chairs: Bob Steagall, Mark Hoemmen, Guy Davidson

2.4 Other Papers and proposals

2.5 Future F2F meetings:

2.6 future C++ Standard meetings:
https://isocpp.org/std/meetings-and-participation/upcoming-meetings

- 2020-11: (New York, tentative) Cancelled.
- 2021-02-22 to 27: Kona, HI, USA

3. Any other business
Reflector
https://lists.isocpp.org/mailman/listinfo.cgi/sg14
As well as look through papers marked "SG14" in recent standards committee paper mailings:
http://open-std.org/jtc1/sc22/wg21/docs/papers/2015/
http://open-std.org/jtc1/sc22/wg21/docs/papers/2016/

Code and proposal Staging area
https://github.com/WG21-SG14/SG14
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 meeting

Time: Jul 8, 2020 02:00 PM Eastern Time (US and Canada) : Embedded: Ben Craig

Time: Aug 12, 2020 02:00 PM Eastern Time (US and Canada) : Fiinance: Jens, Stephan

Time: Sep 9, 2020 02:00 PM Eastern Time (US and Canada) : Security: Mathew Butler

Time: Oct 14, 2020 02:00 PM Eastern Time (US and Canada): Games: Rene

Thanks Matthew for the linnks. Here are the minutes from June SG14 call. Thanks also to Matthew for the minutes.

Chaired by Guy Davidson.

No objections to previous minutes
No action items from prev meeting

Cppcon:
will go ahead, online/offline ratio undecided
sg14 face to face cppcon will not happen, will have regular online meetup instead
submissions still (sort of) open
reviewers for papers for embedded track appreciated

Future CPP standard meetings:
ISO & Insights have put face-to-face meetings on hold for moment
Currently moratorium on forwarding papers on, but may not continue due to length of covid lockdowns

Papers:

Colony:
Wording is progressing with help from Jens
Question around having reserve function - whether necessary:
- not present in deque
- performance detriment when present in previous benchmarks (for full reserve implementation)
- question whether necessary regarding shifting allocation out of hot loop code - suggestion that this is solved to an extent via allocators when this is necessary
- also question around whether is necessary to stop colony from allocating lots of small memory blocks when the rough number of total elements is known in advance - suggestion that this is solved by adjusting minimum block capacities
- suggestion that we can always add this later in standardization process

- inclusion may force necessity of another function to free up unused memory blocks - as opposed to shrink_to_fit, which is not required to keep iterators to elements valid
- retaining memory blocks where all elements have been erased - could be problematic in terms of creating uncertainty for developers as to whether a memory block is freed
- question for later paper revision - should shrink_to_fit consolidate container and remove skips, or reallocate elements to maximum capacity blocks and create best-case cache locality

Question around supplying min/max block capacity limits to constructors/functions - whether invalid (out-of-range) supplied limits should be silently corrected, or if functions should throw/return an error code/some combination of these:
- Constructors can throw anyway, so throwing or defining as UB most logical
- Settle for throwing as more straightforward
- Is skipfield necessary - seems to be for any future implementation
- Nathaniel Niel's suggestion of alternative index - unclear as to what implementation would be - may get back to Matt later

Affinity:
New paper, 2155, following on from 1436
Summary of changes
Overview:
- p1436 -executor-centric with hints - required knowledge of abstraction of executor
- Switch to algorithmic-centric properties - describes what freedoms implementation has, semantics of algorithm
- Can be applied more generically to more executors
- Executor properties + execution policy properties
- Algorithm can execute what user is requesting based on executor's hardware abstraction
- Definition of 'locality interference' - metric

Jens wording critique:
- should contain editorial instruction that this should apply on top of p0443
- virtual machines/GPU's/explicit hardware descriptions do not fit within C++ abstract machine, should be moved to proposal
- 'bulk algorithm' concept/word is not existing word in C++ standard, needs explicit explanation if used
- 'must' too ambiguous, use 'shall'. 'should' similarly cannot be used
- only italicize terms when defining them, not elsewhere
- 'higher constructive interference' not defined, 'hardware constructive interference' is
- changes/additions to abstract machine potentially affects ability of

compiler to optimize
- work items needs explaining

CPP deterministic exception group: nothing
Embedded: Nothing to report
Financial: nobody present
Games: Linear algebra reviewing next week in LEWG online, BLAS & operator overload, 1500 UTC monday
Kona is next CPP standards meeting in line

Minutes for 2020/07/08 SG14 Conference Call

Topic: SG14 Low Latency Monthly

This meeting is Embedded focused and is chaired by Ben Craig.

Hi,

Michael Wong is inviting you to a scheduled Zoom meeting.

Topic: SG14 monthly Apr 2020-Oct 2020

Time: Jun 11, 2020 02:00 AM Eastern Time (US and Canada) 6 UTC
Jun 11, 2020 02:00 AM 6UTC
Jul 8, 2020 02:00 PM 18 UTC
Aug 12, 2020 02:00 PM 18 UTC
Sep 9, 2020 02:00 PM 18 UTC
Oct 7, 2020 02:00 PM 18 UTC

Join from PC, Mac, Linux, iOS or Android:
https://iso.zoom.us/j/819108882?pwd=L0pPZm5QRWlTYXdJditvU1JLdjFYUT09
Password: 013549

Or iPhone one-tap :
US: +13462487799,,819108882# or +14086380968,,819108882#
Or Telephone:
Dial(for higher quality, dial a number based on your current location):
US: +1 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or 877 853 5247 (Toll Free)
Meeting ID: 819 108 882
Password: 013549
International numbers available: https://iso.zoom.us/u/abhaIjFKLZ

Or Skype for Business (Lync):
https://iso.zoom.us/skype/819108882

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 to ISOCPP.org

1.4 Action items from previous meetings

2. Main issues (125 min)

2.1 General logistics

Inverted meeting frequency?

CPPCON logistics
CPPCON SG14 plans
CPPCON Embedded Track: Ben Saks
future C++ Standard meetings:

2.2 Paper reviews

Review of
Freestanding (previously p0829)

Freestanding Library: Easy Utilities	Ben Craig	https://wg21.link/P1642
Freestanding Library: Rewording Status Quo	Ben Craig	https://wg21.link/P1641
Freestanding Language	Ben Craig	https://wg21.link/P1105

Fixed point (and others p0037)

CNL (fixed point, elastic numbers et. al)	John McFarlane	https://wg21.link/P0037 https://github.com/johnmcfarlane/cnl#further-reading

ring_buffer (P0059 does this need a new champion?)

Ring Buffer	Guy, Dan Raviv, Matthew Butler, Inbal Levi	http://wg21.link/p0059

Ring buffer minutes
Inbal working on non-atomic version of the ring buffer, but currently busy with executors
trying to think about customization points
may be ready by next meeting

Matthew is helping, Steffan TJ interested based on inhouse implementations in finance
Paul B also can help, do you have any github site? Yes
its in SG14 github with version of SG14 paper and repo; ay be out of date
different people want different things
Guy: perhaps a separate telecon for interested parties

2.2.1 any other proposal for reviews?

SG14/SG19 features/issues/defects:

https://docs.google.com/spreadsheets/d/1JnUJBO72QVURttkKr7gn0_WjP--P0vAne8JBfzbRiy0/edit#gid=0

Audio: from the perspective of myself as the representative from Apple on the domain. Apple is definitely not against an audio library in the STL, however we do think the approach of the current proposal has enough problems to leave it serving no developer audience effectively (and not just on our platforms). Our paper detailing these critiques and suggestions for design is: http://wg21.link/P1746

The subsequent audio use cases paper that Michael references is http://wg21.link/P2054 and we were supposed to do a follow up survey based on it, however that has indeed stalled due to some disagreement between authors as to whether we are ready for that or need to expand the use cases further.

Linear Algebra:
C++ LA paper aiming for a slot for LEWG, BLAS is done
updating it for concepts and require clauses, should be a shorter paper.

Discuss future direction of detreministic C++ exception group :

Low-cost Deterministic C++ Exceptions for Embedded Systems
<
https://www.research.ed.ac.uk/portal/files/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf>

Low-cost
Deterministic C++ Exceptions for Embedded Systems James Renwick James
Renwick et al.

2.3 Domain-specific discussions

2.3.1 SIG chairs

- Embedded Programming chairs: Ben Craig, Wouter van Ooijen and Odin Holmes, John McFarlane
<http://wiki.edg.com/bin/edit/Wg21belfast/McFarlane?topicparent=Wg21belfast.SG14CPPCON2019-09-17;nowysiwyg=1>

- Financial/Trading chairs: Stephan TJ, Carl Cooke, Neal Horlock, Mateusz Pusz, Clay Trychta,
- Games chairs: Rene Riviera, Guy Davidson and Paul Hampson
- Linear Algebra chairs: Bob Steagall, Mark Hoemmen, Guy Davidson

2.4 Other Papers and proposals

2.5 Future F2F meetings:

2.6 future C++ Standard meetings:
https://isocpp.org/std/meetings-and-participation/upcoming-meetings

- 2020-11: (New York, tentative) Cancelled.
- 2021-02-22 to 27: Kona, HI, USA

3. Any other business
Reflector
https://lists.isocpp.org/mailman/listinfo.cgi/sg14
As well as look through papers marked "SG14" in recent standards committee paper mailings:
http://open-std.org/jtc1/sc22/wg21/docs/papers/2015/
http://open-std.org/jtc1/sc22/wg21/docs/papers/2016/

Code and proposal Staging area
https://github.com/WG21-SG14/SG14
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 meeting

Time: Jul 8, 2020 02:00 PM Eastern Time (US and Canada) : Embedded: Ben Craig

Time: Aug 12, 2020 02:00 PM Eastern Time (US and Canada) : Fiinance: Stephan TJ

Time: Sep 9, 2020 02:00 PM Eastern Time (US and Canada) : Security: Mathew Butler

Time: Oct 14, 2020 02:00 PM Eastern Time (US and Canada): Games: Rene

Minutes for 2020/08/12 SG14 Conference Call

Minutes for todayâ€™s meetingâ€¦

Ben Craig: Minutes
Staffan Tjernstrom: Chair
Michael Wong
Arthur O'Dwyer
Andrew Lumsdaine
Basit Ayantunde
Billy Baker
connor horman
Derek Haines
Henry Miller
Inbal Levi
Maxime Laine
Muklek Bokth Choudhury
Robert Douglas
Ronen Friedman
Hubert Tong

Wong: Inverted meetings: maybe once every three or four months while coronavirus is going. CppCon now fully online. Kona meeting is in Jeopardy. Bulgaria, is the next most likely F2F.

Ben: Pitch I've heard is that 6 months after sports start is when we'll have a new F2F

Inbal: Virtual plenary is in the works

Staffan: All we know is that we don't know.

Wong: Affinity was reviewed in SG1, good to move forward into LEWG. Low level one needs more discussion. Now the paper is talking more about interference patterns. Adjacency locality property work was discussed in an inverted meeting.

Staffan: I see us doing what needs to be done to get executors done, maybe these will follow along.

Inbal: Technical difficulties with ring buffer meeting yesterday. Gathering the various inputs for ring buffer. May want to move some ring buffer things from ring buffer paper to P1958 and P0260. Need use cases. Jens suggested that this be a ring span.

From Maxime LainÃ© to Everyone: 01:22 PM
sure, but I can't see any more use cases as those already there sadly.. (perhaps something driver related for networking but seems a bit stretched out)

From Inbal Levi to Everyone: 01:25 PM
Jens have suggested something along the lines of ring_span, so that's one direction. But we need to figure out if Guy's popping-strategy (from P0059) is something that is worth importing to P1958

Staffan: LEWGI wasn't thrilled with P1958 interface, but didn't hate it either. LEWGI wanted to see how things differed with ring buffer.

Staffan: P0593 implicit object creation. Adopted in the core language as a DR. start_lifetime_as isn't in the library yet. Very confident that it will be in C++23.

Staffan: P0593 is important to finance because of pipeline applications, similar to niall's load / unload objects out of memory ideas.

Hubert: std::launder won't help with the serialization things. start_lifetime_as only lets you claim that there is an object, previous references need to go through std::launder.

Staffan: Probably wait to talk about member layout control until Rene is able to make it on.

Wong: September is probably the right time to reconnect with Renwick to get forward progress on the paper and implementation.

Inbal: Trying to summarize decisions that are made in the low cost exception space for a cppcon talk.

Ben: Paper in the works to discuss error handling tradeoffs. Will send it to Wong and Inbal.

Staffan: Big finance concern is just getting C++20 implemented and seeing the impact, like with

jthread and the sync primitives. Seeing real at-scale impact with concepts and modules.

Wong: How well does the networking proposal work for finance? Maybe there's an RPC proposal?

Staffan: Anything that helps low level timing stability or development throughput is of interest. In finance, lots of people have the same idea at about the same time, but only the first 50% of people or so will make money on it.

Wong: How do contracts feel in your space?

Staffan: Vital, just not that everyone has realized that what they are doing is contracts work.

Wong: Bloomberg is largely doing it for verification.

From Robert Douglas to Everyone: 01:46 PM
Are there any papers in flight, that anyone is aware of, for attributes for hot/cold, a-la likely/unlikely?

Robert Douglas: gnu has hot / cold as well. Has anyone seen a paper along those lines?

Staffan: likely / unlikely don't strictly do that, it does help. Not for data though.

Staffan: finance ends up living in a weird world where the most common things need to be marked "unlikely" for performance.

From Me to Everyone: 01:50 PM
Timur Doumler is also trying to drag an assume attribute back from contracts

From connor horman to Everyone: 01:52 PM
I'd be interested in seeing that as well.

Robert Douglas: I usually see this for error handling functions that are marked both as unlikely and cold.

Wong: Could use a paper exploring this.

From Inbal Levi to Everyone: 01:53 PM
Yes. But to my humble understanding, the support is vague.

From Maxime LainÃ© to Everyone: 01:53 PM
hot/cold pragmas/markers seems a bit out of language, more of a compiler feature, but I'm open for discussion as well

From Inbal Levi to Everyone: 01:53 PM
P1774R3

Staffan: Would like to have an offline discussion talking about hot / cold.

Actions:
* Wong to talk with Renwick
* Staffan + Robert to talk about object layout
* Revisit object layout space when Rene is on the call

From: SG14 <sg14-bounces_at_[hidden]> On Behalf Of Matthew Butler via SG14
Sent: Wednesday, August 12, 2020 10:14 AM
To: Low Latency:Game Dev/Financial/Trading/Simulation/Embedded Devices <sg14_at_[hidden]>
Cc: Matthew Butler <mbutler_at_[hidden]>
Subject: [EXTERNAL] Re: [SG14] SG14 Aug 12 meeting

Me as well.

On Wed, Aug 12, 2020 at 9:10 AM RenÃ© Ferdinand Rivera Morell via SG14 <sg14_at_[hidden]<mailto:sg14_at_[hidden]>> wrote:
Same for me.
On Wed, Aug 12, 2020, 11:07 AM Guy Cpp via SG14 <sg14_at_[hidden]<mailto:sg14_at_[hidden]>> wrote:
Apologies: I won't be able to attend this month's meeting. Have a great time!

Cheers,
G

On Tue, 11 Aug 2020 at 17:06, Tjernstrom, Staffan via SG14 <sg14_at_[hidden]<mailto:sg14_at_[hidden]>> wrote:

Topic: SG14 Low Latency Monthly

This meeting is Finance focused and is chaired by Staffan TjernstrÃ¶m.

Hi,

Michael Wong is inviting you to a scheduled Zoom meeting.

Topic: SG14 monthly Apr 2020-Oct 2020

Time: Jun 11, 2020 02:00 AM Eastern Time (US and Canada) 6 UTC
Jun 11, 2020 02:00 AM 6UTC
Jul 8, 2020 02:00 PM 18 UTC
Aug 12, 2020 02:00 PM 18 UTC
Sep 9, 2020 02:00 PM 18 UTC
Oct 7, 2020 02:00 PM 18 UTC

Join from PC, Mac, Linux, iOS or Android:

Password: 013549

Or iPhone one-tap :
US: +13462487799,,819108882# or +14086380968,,819108882#
Or Telephone:
Dial(for higher quality, dial a number based on your current location):
US: +1 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or 877 853 5247 (Toll Free)
Meeting ID: 819 108 882
Password: 013549
International numbers available: https://iso.zoom.us/u/abhaIjFKLZ$>

Or Skype for Business (Lync):
https://iso.zoom.us/skype/819108882$>

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 to ISOCPP.org

1.4 Action items from previous meetings

2. Main issues (125 min)

2.1 General logistics

Inverted meeting frequency?

CPPCON logistics
CPPCON Embedded Track: Ben Saks
future C++ Standard meetings:

2.2 Paper reviews

2.2.1 any other proposal for reviews?

SG14/SG19 features/issues/defects:

https://docs.google.com/spreadsheets/d/1JnUJBO72QVURttkKr7gn0_WjP--P0vAne8JBfzbRiy0/edit#gid=0QCiCUI56OFLF7QJGeMDjmTacAwLxEYi$>

These are more status updates than full-blown reviews this time around.

Affinity https://github.com/cplusplus/papers/issues?q=is:issue+P1436$>

System topology discovery https://github.com/cplusplus/papers/issues?q=is:issue+P1795$>

Ring Buffer https://github.com/cplusplus/papers/issues?q=is:issue+P0059$>

Object Creation https://github.com/cplusplus/papers/issues?q=is:issue+P0593$>

Member layout control https://github.com/cplusplus/papers/issues?q=is:issue+P1605$>

Discuss future direction of detreministic C++ exception group :

Low-cost Deterministic C++ Exceptions for Embedded Systems
<
https://www.research.ed.ac.uk/portal/files/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdfI3Qg!-Cx7ecFH_AJfg5_2VUHA4q_4m6HrgQCiCUI56OFLF7QJGeMDjmTacHGnA89n$>>

Low-cost
Deterministic C++ Exceptions for Embedded Systems James Renwick James Renwick et al.

2.3 Domain-specific discussions

2.3.1 SIG chairs

- Embedded Programming chairs: Ben Craig, Wouter van Ooijen and Odin Holmes, John McFarlane

- Financial/Trading chairs: Staffan TjernstrÃ¶m, Carl Cooke, Neal Horlock, Mateusz Pusz, Clay Trychta,
- Games chairs: Rene Riviera, Guy Davidson and Paul Hampson
- Linear Algebra chairs: Bob Steagall, Mark Hoemmen, Guy Davidson

2.4 Other Papers and proposals

2.5 Future F2F meetings:

2.6 future C++ Standard meetings:

https://isocpp.org/std/meetings-and-participation/upcoming-meetings$>

- 2020-11: (New York, tentative) Cancelled.
- 2021-02-22 to 27: Kona, HI, USA

3. Any other business
Reflector
https://lists.isocpp.org/mailman/listinfo.cgi/sg14$>
As well as look through papers marked "SG14" in recent standards committee paper mailings:
http://open-std.org/jtc1/sc22/wg21/docs/papers/2015/$>
http://open-std.org/jtc1/sc22/wg21/docs/papers/2016/_$>

Code and proposal Staging area
https://github.com/WG21-SG14/SG14$>
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 meeting

Time: Sep 9, 2020 02:00 PM Eastern Time (US and Canada) : Security: Mathew Butler

Time: Oct 14, 2020 02:00 PM Eastern Time (US and Canada): Games: Rene

Kind Rgds
Staffan Tj.
pp Michael Wong