Document Number: P2031R0
Date: 2020-01-13
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: Meeting Minutes 2019/10/08-2020/01/07

Contents

Minutes for 2019/10/08 SG14 Conference Call

1.1 Roll call of participants
>
Michael Wong, joshua Cannon, Andreas Fertig, ben Craig, Cgarles Bay, Guy Davidson, John, McFarlane, Matthew Butler, Rene Riviera, Staffan Tj. Jan Wilmans , tanki zhang, Domargo Saric, Paul Preney

> 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
>
> review CPPCON SG14 meeting
>
> http://wiki.edg.com/bin/view/Wg21belfast/SG14CPPCON2019-09-17
>
>

2.2 Paper reviews
>
> 2.2.1 Discuss Possible paper agenda for Belfast SG14 F2F
>
> 1. Linear Algebra by Bob Steagall/Guy Davidson Passed SG14 headed for LEWG
>
> 2. BLAS by Mark Hoemmen Passed SG14 headed to LEWG
>
> Any other papers?
>
Belfast SG14 F2F morning Friday
LEWG will be lookign at some wording and SG14 can react to changes from LEWG
Free standing?
Affinity, and topology discovery
Debug control:
Member layout control: dont need proxy.

> 2.2.2 Error Size Benchmarking by Ben Craig Update.
>
>
> P1866R0: Error speed benchmarking is ready if people want to talk about
> that.
>
>
>
> https://raw.githack.com/ben-craig/error_bench/master/error_speed.html
>
>
> each time it runs, 31 nops is executed, similar code is there for
exception , half a mllion iterations
this will help get all the cases covered
tested 3 abstract programs
1. write to a global -putting 0 into global int: gcc and clang has a
missing optimzation, but not inherent to EH, VS 64 happy path was a bit
faster

2. write to a parameter passe in by pointer, *value=0 instead of global
int=0; it flattens out the performance between throwing exception and abort
for all the 64 bit targets

3. when we return a value we still get flat performance,
x86 exception is not using table based exception, and is using a linked
list of longjmp, so it never looks good
gcc and clang are both 64 bit

could it be the ABI costing this much?
can confirm Ben's 32 bit MSVC because of SEH, thats why llvm had trouble,
MS seems to wrap their C++ EH ontop of SEH
which works in C

You can also throw C++ EH through C code, this is part of COM unification
on C and C++

There was also a lot of TLS behind the scene with SEH

Did you try with addign noexcept? no, you get same binary with
fnoexception, then if you miss something then things get bad
Happy is no error was signal
number of cycles per function call

Sad path is some error was signaled, DO ERROR is true

all 3 cases EH overwhelms 100x even in the best case x64 msvc, worst case was clang
but MSVC has doen the best at improving EH runtime cost
GCC8 may have a minor update but was not benchmarked
one time costs or per depth, hard to separate 2 frames from the one frame
determinism is a huge costs in EH

can we let developer choose which tradeoff
return code are slower on happy path
but eh is slower on sad path
no choice today
can we have different fine grain choises depeneding on happy or sad path,like move landing path code? yes, but it is still not enough

return_nt_struct in write a global happy path may already have a patch that supports register usage
knows from the abi that the caller does not have to clobber certain registers so I am confused my gcc and clang fails to do something like this
yes they do use non-volatile registers, and so you use old value of nv register, so you can get back old value

using attributes to help compiler? Can we do that to show MSVC developers believe likely and unlikely will make big differences
herception, tightening the compiler implementation for exception
https://github.com/psiha/err/blob/master/include/boost/err/result_or_error.hpp
search for "likely"

possibly evening session?
MS seems to always ask for pgo as the default answer for speed issues, btu you cant use it for static libraries
passingstruct of 2 ptrs, then ref struct is the best choice, oppistion to outcome or common wisdom of passing by val on small things

compiler explore like idea of benchmarkign quick code: quickbench

isocpp.org publish paper

P1640R0: Error size benchmarking
>
>
> https://raw.githack.com/ben-craig/error_bench/master/error_size_benchmarking.html
> <
>
> https://urldefense.proofpoint.com/v2/url?u=https-3A__raw.githack.com_ben-2Dcraig_error-5Fbench_master_error-5Fsize-5Fbenchmarking.html&d=DwMGaQ&c=r2dcLCtU9q6n0vrtnDw9vg&r=bHyceIQQHQvbfTSw

vF3b5ym3XCQIh0_iFRNJbNk-FCc&m=_OFSroXnnYHKfBQqw8TVSac0et4fEQ80IMeaj-lWcD4&s=LGjT-TVB94ptHzUmdPNh4LJr1eMpKuAcmL7pQSWzxxA&e=>
>
>
> 2.2.3 PTF/Colony?
>
> 2.2.4 Linear Algebra update from Sept 4
>
> Next call: Nov 6: 3 PM ET
>
> 2.2.5 any other proposal for reviews?
>
> 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
>
> some game studios still blast templates
clang profile compiler is reputed to be very good
C++20 has a list that support each domains as a paper and a talk
Better iostream ? PLEASE!
https://github.com/fmtlib/fmt
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2018/p0645r1.html
PP used it in his PHD course. IOstream is made to do everything, should allow u to select exception, locale,
use new std::function with inline namespaces? yes 2 proposals P0792 func_ref is not owning and P0288 any invocable, both did not make it into C++20
aspects of std::function that can be improved by chaning implementation
MS STL copies boost function design just to get a checkmark, but it does not work
transfer this email into GG doc for all to add

> 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
>
> - *2019-11-04 to 09: Belfast, Northern Ireland;* Archer Yates
> -
> - 2020-02-10 to 15: Prague, Czech Republic
>
> - 2020-06-01 to 06: Bulgaria
> - 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
>
> TBD
>
> Oct 9: mailing deadline Monday Nov 7
>
> Nov 13: cancelled due to C++ Std meeting
>

Minutes for 2019/12/11 SG14 Conference Call

Michael's notes

> 1.1 Roll call of participants
>
Michael Wong, Connon Horman, Dan Raviv, Guy Davidson, Inbal Levi, Billy Baker, John MacFarlane, Matthew Butler, Max Gardner, Mark Hoemmen, Neil Horlock, Paul Bendixen, Rene Rivera, Ronen Friedman, Steffan TJ, Paul Preney, Eduardo Costa, CHarles Bay

> 1.2 Adopt agenda
>
Yes

> 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
>
> recap C++ 20 meeting
>
Mailing deadline is Jan 13
report on Numerics - P1889, or P1890 safety
concerns from BSI on missing rationales from the indvidual papers from the omnibus papers
Discussion on need for rationale in TS
AI: MW to ask if rationale should be kept in general

Linear Albegbra progressing
take traits in syntax paper and theer would be an engine call blas
assume yo ualso add mathematical optimzations
yes blas wrapper has existng optimized library that is in C or tuned by chip vendors
mdspan is the iterator idiom for matrices, it i s multidimentional iterator

Neil asked QUB has a question about associative Legendre polynomials and as it is different then unassociative versions, would we need to feed that proposal through SG14, 6?

Suggest SG6 and SG19 as this is related to differential calculus
Suggest posting question to SG19

>
> 2.2 Paper reviews
> P1436R2 <https://wg21.link/p1436r2> Executor properties for
> affinity-based execution Gordon Brown Gordon Brown, Ruyman Reyes, Michael
> Wong, H. Carter Edwards, Thomas Rodgers, Mark Hoemmen P1436R2
> <http://wiki.edg.com/bin/view/Wg21belfast/P1436R2> 60 minutes
> P1795r1 <https://wg21.link/p1795r1> System topology discovery for
> heterogeneous & distributed computing Gordon Brown Gordon Brown, Ruyman
> Reyes, Michael Wong, Mark Hoemmen, Jeff Hammond, Tom Scogland, Domagoj
> Å ariÄ‡ P1795R1 <http://wiki.edg.com/bin/view/Wg21belfast/P1795R1> 60
> minutes
>
> 2.2.5 any other proposal for reviews?
>
>
> 2.3 Domain-specific discussions
>
>
> I would like to ask each Domain chair to consider the following so we can
> put out a call for proposal:
>
> 1. What are the features in recent C++ since SG14 formation (C++17,20)
> that are useful in your domain?
>
> 2. What features you would like to pursue in SG14 for your domain? For
> example, for embeded : inplace containers, intrusive pointers, fixed sized
> containers
>
>
> 2.3.1 SIG chairs
>
> - Embedded Programming chairs: Ben Craig, Wouter van Ooijen and Odin
> Holmes, John McFarlane
>

Freestanding : a implementation is available to try out by Paul Bendixen in
a GCC branch
https://gitlab.com/avr-libstdcxx
The docker image showshow to build it, the gcc project is a fork of the
official gcc with changes in the p0829-XXX branches
On freestanding, a comment. I don't know if this is already a proposal or
being talked about, but is there a reason why exceptions are required for
freestanding? because freestanding is already in the standard, and it has

exceptions,
P0829 has an implementation, this is the library, which adds things
The other proposal is for language and proposes remove things but is still controversial

Contracts for safety

> - Financial/Trading chairs: Stephan TJ, Carl Cooke, Neal Horlock,
> Mateusz Pusz, Clay Trychta,
>
likely unlikely by Clay is in C++20
affinity and topology discovery
Neil: pipelines
intrusive
inplace
and fixed sized containers

- Games chairs: Rene Riviera, Guy Davidson and Paul Hampson
>
memory management for Games for C++17 from Brent Freedman
P00040
flat map by ach Laine
ptf colony by Matthew Bentley
ring buffer by Guy P0059, fell off at Rapperswil, concerned about single producer/single consumer, policy decisions, and whetehr it should be concurrent, Dan Raviv, and Matthew BUtler coudl help
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2017/p0059r4.pdf
Mark asked would std::destroy_at etc. change the whole pop traits thing?

Rene: more control over the compilation and operation of C++. I have the member layout control (competing) paper, for example.
threading/executors related controls is high. Anything to improve debugging capabilities and experience. Better performing containers.

> - Linear Algebra chairs: Bob Steagall, Mark Hoemmen, Guy Davidson
>

Propose to start a reflector discussion
Does executors serve our domains
exceptions is still a hot issue

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
>
> - *2019-11-04 to 09: Belfast, Northern Ireland;* Archer Yates
> -
> - 2020-02-10 to 15: Prague, Czech Republic
>
> - 2020-06-01 to 06: Bulgaria
> - 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
> Dec 11, 2019 02:00 PM
> Jan 8, 2020 02:00 PM: Jan 13 is mailing deadline
> Feb 12, 2020 02:00 PM
> Mar 11, 2020 02:00 PM

Staffen's notes:

1.1 Roll call of participants

Michael Wong, Guy Davidson, John McFarland, Billy Baker, Connor Horman, Dan Raviv, Eduardo Costa, Inbal Levi, Matthew Butler, Max Gardner, Mark Hoemmerman, Neil Horlock, Paul Bendixen, Paul Preney, Rene Riviera, Ronen Friedman

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

recap C++ 20 meeting

John was in SG6 (Numerics) and WG32 (Vunerabilities). The big paper is turning into something like a TS.

Guy spent his time in LEWGI. BLAS paper was forwarded to LEWG. Syntax paper probably will only see a small move forward for Prague given the timing. Anthony Peacock is writing a SIMD implementation of the Syntax.

Mark H Good feedback from SG6 and LEWGI on BLAS.

2.2 Paper reviews
P1436R2<https://wg21.link/p1436r2>

Executor properties for affinity-based execution

Gordon Brown

Gordon Brown, Ruyman Reyes, Michael Wong, H. Carter Edwards, Thomas Rodgers, Mark Hoemmen

P1436R2<http://wiki.edg.com/bin/view/Wg21belfast/P1436R2>

60 minutes

P1795r1<https://wg21.link/p1795r1>

System topology discovery for heterogeneous & distributed computing

Gordon Brown

Gordon Brown, Ruyman Reyes, Michael Wong, Mark Hoemmen, Jeff Hammond, Tom Scogland, Domagoj Å ariÄ‡

P1795R1<http://wiki.edg.com/bin/view/Wg21belfast/P1795R1>

60 minutes

2.2.5 any other proposal for reviews?

John talks about P1889, with changes in P1890. Based on P0101. In the manner of a TS of a few years ago. SG6 also covered some more LEWG-like concerns (how do you return a number vs an error).

BSI met on Monday, also talked about John's paper(s). They had a concern about losing the rationale from the earlier papers. Guy brings John up-to-date as John has not had a chance to read the minutes as of yet.

Mark H made the point that rationale's need to be kept up to date, which adds information and work.

2.3 Domain-specific discussions

Neil Horlock wonders if a prospective paper on unassociative Legandre polynomials is something we would want to look at. Suggestion is to push that to SG6. Inbal mentions ULP are used in differential algebra. Michael says that differential algebra is being discussed in SG19, they have a call tomorrow. Neil might be able to post a summary of the paper on the SG19 list by then.

Inbal has a question on how the BLAS paper co-exists with the Syntax paper.

Guy: One of the reasons why LEWGI wants a reference implementation of BLAS on top of the "syntax". Mark describes how it would work from the BLAS perspective, analaguous to how Eigen could call BLAS.

John was wondering how sensible it was to try to follow the patterns of the standard algorithms. MarkH reponds that MDSpan is the iterator for multi-dimensional domains. The BLAS paper

takes the existing standard, and essentially wraps it for multi-dim. But there is a corner case that MarkH will cover offline with JohnMcF.

Guy mentions that adding special functions to C++ is a difficult topic. The authors are encouraged to speak with Walter Brown.

I would like to ask each Domain chair to consider the following so we can put out a call for proposal:

1. What are the features in recent C++ since SG14 formation (C++17,20) that are useful in your domain?

2. Brittany Freedman - P0040

3.

2. What features you would like to pursue in SG14 for your domain? For example, for embeded : inplace containers, intrusive pointers, fixed sized containers

Guy: Containers. Flat_Map and Colony.

John McF: Come up with where we stand on the various topics that are coming up

Guy presents about the P0059 paper on Ring Buffers, it went into suspension after Rapperswill. As audio garners momentum, this will reinvigorate it. There are many, many choices.

Paul Bendixen presents on the paper split that Ben is working on to push through Freestanding. He has an implementation, can give us a github page.

Staffan to start a thread on the reflector to gather ideas.

Paul Bendixen responds to a question about exceptions and freestanding from the chat.

2.3.1 SIG chairs

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

- *2019-11-04 to 09: Belfast, Northern Ireland;* Archer Yates
-
- 2020-02-10 to 15: Prague, Czech Republic

- 2020-06-01 to 06: Bulgaria
- 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
Dec 11, 2019 02:00 PM
Jan 8, 2020 02:00 PM: Jan 13 is mailing deadline
Feb 12, 2020 02:00 PM
Mar 11, 2020 02:00 PM

Minutes for 2020/01/07 SG14 Conference Call

> 1.1 Roll call of participants
>
Michael wong, Ben Craig, Antony Peacock, billy Baker, Connor Horman, Guilherme, Guy davidson, Inbal Levi, Jesun Firoz, Jhon Adams, Marco Foco, Max Gardner, Maxime Laine
Ronen Friedman, Staffan Tj, Derek haines, tanki Zhnag, Eduardo Costa, Ben Saks, Matthew Butler

> 1.2 Adopt agenda
>
Yes

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

> 1.4 Action items from previous meetings
>
N/A

> 2. Main issues (125 min)
>
> 2.1 General logistics
>
> prepare for Prague meeting and mailing deadline Jan 13
>
> 2.2 Paper reviews
>
> - D2013,Freestanding Language: Optional ::operator new revised D2013 to
> D2013R0.2 by Ben Craig
>
> https://raw.githack.com/ben-craig/freestanding_proposal/master/core/new_delete.html
>
>
> This is a part that is torn out of P1105R1 Leaving no room for a
> lower-level language: A C++ Subset
>

embedded may not have a heap, so make it hard to use, this makes it impleemtation defined

will provide all the replaceable functions or none of them
placement new still exists

presence of operator new is implementation defined, AND user does not provide one, then it is ill-formed
IFNDR is not needed

there is usage experience in this
BS replaced global operator delete with something that does not exists; so if someone accidentally uses the delete in question, the nprogram does not build

STJ mentions providing no throw T placement new and needed to check for 0, as user is surprised that may have a try catch in it

RF clarifies and codifies existing 2 or so minor variations
go to EWG first before may be going to LEWG
Forward P2013 as is with the minor editing quotes
SF/f/n/a/sa
9/10/0/0/0
approves to go to EWG

>
> - host notification (keyboard input, mouse input,by Guy Davidson
>
defer discussion to after the call

- SG14 domain features by SG14/Michael
>
https://docs.google.com/spreadsheets/d/1JnUJBO72QVURttkKr7gn0_WjP--P0vAne8JBfzbRiy0/edit#gid=0
please feel free to update this spreadsheet on features and issues

- affinity by Gordon
>
defer

>
> 2.2.1 any other proposal for reviews?
>
> 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
>
> - 2020-02-10 to 15: Prague, Czech Republic
>
> - 2020-06-01 to 06: Bulgaria
> - 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
> Dec 11, 2019 02:00 PM
> Jan 8, 2020 02:00 PM: Jan 13 is mailing deadline
> Feb 12, 2020 02:00 PM
> Mar 11, 2020 02:00 PM