

Document Number: P0445R0
Date: 2016-10-16
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 Meeting Minutes 2016/09/21- 2016/10/13

Contents

Minutes for 2016/09/07 SG14 Conference Call	2
Minutes for 2016/09/21 SG14 Conference Call	7
Minutes for 2016/10/13 SG14 Conference Call	14

Minutes for 2016/09/07 SG14 Conference Call

Meeting minutes by Michael

Start Time: Wednesday, Sept 7, 2016, 11:00 AM US Pacific Time, 2 PM US Eastern Time

End Time: 1:00 PM US Pacific Time (duration: two hour)

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.

I will take notes for the calls, though it will also be recorded.

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Michael Wong, Adi Shavit, Brett Searle, Carl Cook, Guy Davidson, John McFarlane, Lee Howes, Patrice Roy, Ronan Keryall, Sean Middleditch, Joe Lothar

1.2 Adopt agenda

Approved.

Add NW user groups agenda

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

Approve

1.4 Review action items from previous meeting (5 min)

C++17 is N4604

2. Main issues (125 min)

2.1 Review paper mailings and SG14 position/feedback on any papers from the mailing: 30 min

Previous papers:

1. flat map: Sean M

Looking for help with wording has rough draft, working with some people through CPPCON

2. fixed point: John M, Lawrence, Marco Foco

new draft being prepared, might need custom integer type, introduce separate type ,

setwidth paper : p

present at cppcon

3. ring span: Guy, Arthur

GDC feedback say its too trivial, now it is thread safe, now have working thread safe version present at cppcon

4. Low level bit manip: Vincent presented in Oulu

5. uninit memry algo: Brittany Friedman
Accepted for C++17

6. datapar (SIMD) : M Kretz, M Gaunard, Joel Falcou
passed SG1, going to LEWG

7. Comparing virtual fns: Scott Wardle/Sunil
Approved for more work in EWG

8. Thread constructor attributes: Patrice
Vincent collaborating, moving along
may be a cppcon SG14

9. install distribution packaging: Brian Fitzgerald
Not on

10. unstable move: Brent
No update

11. Class for status/Optional: Lawrence
No update

12. utility class to represent expected monad: Paul Hampson, Vincent Botet
no update

13. hazard pointers and RCU for lock-free programming: Michael
Will publish updated interface at cppcon
updated paper for issaquah
with RCU

14. 2d display: Michael M
approved in SG13

some concerns with Cairo interface, mozilla moving away : additional interface is encouraged
dont have to base new library from scratch even if we dont want Cairo
can cut and paste a new library people do use
mention at CPPCON SG14

15. event handling and interrupt handling for SG13

16. threading Brett
present for SG14

17. non-allocating std function Carl
Nicolas and Carl
SG14 slot

2.4 Status of future SG14 proposals: 25 min

0. Trading thread and recent STAC SG14 meetings

<https://groups.google.com/a/isocpp.org/forum/#!topic/sg14/4WvbE2iaFNI>

<https://groups.google.com/a/isocpp.org/forum/#!topic/sg14/0em3tc5uuwI>

a. Executors/Heterogeneous device support (Michael, Hartmut), weekly meeting minimal executors

b. EH lite (see below): connect with MS team

c. CPU/cache/memory affinity (Neil) : no update

d. memory allocation (Guy Davidson): no update

1. multiple small vectors: Gonzalo BG et al,
Nevin liber

2. Exception lite and swift-like exceptions: Sunil, Patrice, and Sean Middleditch
parallism TS now terminates and dont unwind on exceptions
3. basic inplace function: Nicolas Fleury, Sean M, etc al
Carl Cook working on this with Nicolas
4. Interprocess communication: Shaun Croton et al
5. hot set: Brittany Friedman
No update
6. Accessors: Ronan, Lee
a simpler proposal requested, interest from Carter on ArrayRef
update for issaquah
7. std::stack: Matthew Bentley
no update
8. plf::colony/stack: Matthew Bentley
Accepted at cppcon
Ask for a review at SG14
9. FAsT associative container: Allan Deutsch
No updates
10. Alternatives to traverse linked data structures: Marcelo Zimbres
11. Delegates for simulations: Miodrag Milanovic
12. Dynamic/runtime concepts: Zach Laine, Andrew Sutton
13. width/set width: John M
presented at cppcon
14. explicit initializer list constructors: Nicolas Fleury
worked with Patrice, unknown if it has a solution paper
15. popping move-only types from priority queue: Ben Deane
No updates
16. affinity, locality and hints: Neil Horlock
No updates
17. intrusive containers (Guy)
Guy will present Hals paper at cppcon
18. FPGAs (Ronan)
Future
18. lock-free queues: michael/lawrence
Presenting at cppcon
19. half precision:
No update

2.5 Talks we proposed to CPPCON:

1. SG14 update Monday:
2. Lock-free Cocurrency Toolkit for Hazard pointers and RCU
3. Heterogenous computing in C++
- 4 Colony
5. Fixed point Monday
6. Exception alternatives Patrice Tuesday
7. Using C++ for low latency: Patrice
8. SG14 meeting Wed

2.6 Talks we proposed for Meeting C++

1. Trading Carl Cook
2. Concurrency toolkit Michael
3. SYCL Maria
4. SG14 update: Guy
5. SG14 meeting: Saturday
6. hardware and profiling talks
7. games development panel for future

2.7 NW Users group

John and Brett on security panel via F5 Rachel Chang

What is in C++17 and What will be in C++20

Volunteers please?

2.7 Future F2F meetings:

3. CPPCON 2016: Wed, Sept 21: 8:30-5 pm
4. Meeting C++ Games Track: Nov 18-19, 2016, Berlin

2.7 future C++ Standard meetings:

[N4571](#) 2016-11 Issaquah meeting information

[N4573](#) 2017-02 Kona WG21 Meeting Information

2017-07-10-2017-07-15: University of Toronto/Canada

3. Any other business

Reflector

<https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/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>

Likely/Unlikely proposal needs CPU developer

4. Review

None

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

Next call : CPPCON Sept 21

Nov 7-12: C++ Std Meeting Issaquah

Close

Minutes for 2016/09/21 SG14 Conference Call

Meeting minutes by Michael

1. Opening and introductions

[CPPCon 2016 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 ISO CPP.org](#)

[1.4 Review action items from previous meeting \(5 min\)](#)

[1.4.1. All: Consider attending Issaquah Meeting in Nov 7-12, 2016.](#)

[1.4.2. All: Consider attending Kona Meeting in Feb 27-March 4, 2017.](#)

[1.4.3. All: Consider attending Toronto Meeting in July 10-15, 2017.](#)

[2. Main issues \(all day\)](#)

[2.1 Review C++17 progress, SG14 logistics, upcoming C++ Standard meetings , C++ Issaquah mailing deadline \(Oct 17\)](#)

[2.2 compression panel Thursday](#)

[2.2 Review proposals.](#)

[2.3 SG14 Goals and Scope and general discussion](#)

[3. Any other business](#)

[4. Review](#)

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

[4.2 Review action items \(5 min\)](#)

[5. Closing process](#)

[5.1 Establish next agenda](#)

[5.2 Future meeting \(past and future calls\)](#)

CPPCon 2016 Agenda

1. Opening and introductions

Reflector

<https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/sg14>

Code and proposal Staging area

<https://github.com/WG21-SG14/SG14>

General Schedule (matching CPPCon break paid and provided by ISO CPP.org; Thank you to ISO CPP.org): Tentative

Start: 8:30

Review papers: 8:30-10

Morning break 10:00 -10:30

Suspend for keynote: 10:30-12:00

Lunch 12:00 2:00

Review papers & Discussion topics: 2:00-3:00

Afternoon (drinks only) 3:00- 3:15

Discussion topics: 3:15-4:15

Afternoon: 4:15 - 4:45

Discussion topics: 4:45-5:30

End: 5:30

Room

Room 409 Joliot-Curie Meydenbauer Center

Opening remarks about how the wiki works

No read only access

1.1 Roll call of participants

Michael Wong (chair)

Sean Middleditch (co-chair, scribe, thank you)

Paul Hampson

- show quoted text -

1.4.2. All: Consider attending Kona Meeting in Feb 27-March 4, 2017.

<https://isocpp.org/files/papers/n4573.txt>

1.4.3. All: Consider attending Toronto Meeting in July 10-15, 2017.

<https://isocpp.org/files/papers/N4607.pdf>

2. Main issues (all day)

2.1 Review C++17 progress, SG14 logistics, upcoming C++ Standard meetings , C++ Issaquah mailing deadline (Oct 17)

C++17 all feature slides, cd ballot comment <https://wongmichael.com/2016/06/29/c17-all-final-features-from-oulu-in-a-few-slides/>

C++ committee draft <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2016/n4606.pdf>

Our github

<https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/1yRloYFAkhs> Hi all,

Regarding the SG14 github repo, Guy and I have now set up more expansive access permissions for this account.

There are now four "owners" (in github terms), namely Guy, Sean, John and myself. This means there should always be someone available with admin rights.

More importantly, I've set up an SG14 "team". I've configured it so that anyone in this team can accept/merge pull requests, and also add/remove others to/from this team.

If you are interested in helping maintain the SG14 repo (rather than just submitting pull requests), then being added to this team is the way to go.

Use [@WG21](#)-SG14/sg14 to mention the SG14 team in comments. These comments will reach all SG14 github team members.

- show quoted text -

<https://github.com/WG21->

[SG14/SG14/blob/master/Docs/Proposals/NonAllocatingStandardFunction.pdf](https://github.com/WG21-SG14/blob/master/Docs/Proposals/NonAllocatingStandardFunction.pdf)

https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/1Sw_qEdIYes

Fixed point real numbers

http://johnmcfarlane.github.io/fixed_point/papers/p0037.html (up to date draft version)

http://johnmcfarlane.github.io/fixed_point/papers/p0037r2.html

http://johnmcfarlane.github.io/fixed_point/papers/p0381r0.html John [McFarlane](#) - Fixed Point

=

Vishal Oza: How does fixed_type work with standard math? JM: would require new overloads; not part of current proposal. asin is interesting because it has a well-known domain, sqrt should scale to input

Alistair: literals and accuracy with decimal fixed point, is core language support required for hex vs binary forms, clarity concerns JM: user-defined literals can solve some problems, need to support decimal

Arthur: could convert into raw float first and then convert to decimal with UDL JM: 0.1 example of where precision is lost, the UDL support can do what we need with some good meta-programming

Paul Hampson: elastic_integer could use a signed input to pick correct underlying type JM: called an Archetype to make it clear that you'll get something *like* the requested type. have a separate proposal with set_width PH: why have int8_t instead of int? JM: the given type is the smallest size it can use, so int would mean a 3-bit number still uses 32 bits

Scott Wardle: compile-time overhead worth the amount of simplification? JM: simplification is great enough to not be a concern, but it's not optimized for compile times yet. SW: so fixed_number doesn't create new types by elastic_integer does? JM: fixed_number adjusts exponent too

Alistair: does it support negative zero? JM: up to the representation type, but int does not

Alistair: guaranteed round-tripping to binary and fixed point and back, if representation does it? JM: should do, yes

JM: should i rewrite paper to focus on composition of types and less on fixed_point, continue on this path, separate this paper? Guy Davidson: paper could use more examples JM: [GitHub](#) paper has examples Michael Wong: prefer examples in paper Alistair: if you've got the guidance, the next move should be formal wording JM: SG6 involvement important Alistair: make sure standard supports complex works JM: will do MW: nail down the original paper and its scope VO: might end up like concepts Arthur: wouldn't encourage use of complex since it won't be write without a specialization JM: make_signed vs make_unsigned can't be specialized for fixed, but numeric_limits can be, needs to be resolved

Patrice Roy: any way to have SG6 look at it? MW: SG14 monthly call could invite SG6 people JM: now presenting set_width

Alistair: are these specific to fixed? JM: just general for integers Alistair: header choice depends matter because of specialization rules for type_traits JM: yes, needs specialization Stefan: I can see some here and there use cases outside of fixed JM: have an example of a widening multiply, and something similar in boost, in a separate paper because it's not specific to fixed MW: directly to LEWG? what header does it go in? JM: could be a new member to numeric_limits?

Vote to move to LEWG

SF: 8 WF: 15 N: 4 WA: 0 SA: 0

Ring Span

https://github.com/Quuxplusone/ring_view/blob/master/p0059r1.pdf

Guy Davidson - rings =

Alistair: does this impose cost in single-thread case? GD: yes, because feedback was that single-thread case is trivial Alistair: want single-thread version GD: will add a single-thread version to proposal for consistency

Arthur: policy configuration GD: complexity due to push/pop behavior

GD: should I promise push template?

Vittorio Romeo: definitely want both concurrent and non-concurrent versions, because it's inconsistent with other containers

VO: move? GD: emplace Sean Middleditch: is that exception safe GD: no, as safe as underlying type SM: potential problems around try_push/try_pop, worth investigating, problems mentioned with concurrent queues in the past

Alistair: rings I've used would be a span over memory and destruct/reconstruct over Arthur: example of doing this, but low-latency wants fast, no excess construct/destruct/reconstruct

JM: push/pop blocks by default, have you considered future? MW: heavy weight

GD: pusher template a good idea? Arthur: no, have T::operator= Stefan: disagreements in our experience among uses of a T SW: how to prepopulate? GD: it's a span SW: how to push twice, can I pull partial objects? JM: can add pusher down the road Alistair: opinion depends on insert iterator GD: there is iterator support yes PH: expand on T::operator= Arthur: wrapper type can use placement new

?: can I use with an atomic concept, would allow use with coroutines GD: don't know

Hazard Pointers

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2016/p0233r1.pdf> Hazard pointer, Maged Michael Requesting feedback on interface

AM: Why is enum cache_policy a class member? This mechanism will create overhead by creating a separate enum for each instantiation.

MW: Can be hoisted out of the class.

Jon [McFarlane](#): Why are hazptr_user and hazptr_remover classes not namespaces?

MW: Namespace will make them vulnerable to ADL problems.

Omar: Replace std::function from hazptr_obj_reclaim with function pointer

Paul [McKenney](#): Extra allocation is required with std::function which is not popular

Vittorio: Customisation point for specifying function?

Eric: Keep std::function as default, but should offer customisation point

MM: could be traits, could be void*, they are contained.

RCU

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2016/p0279r0.html> Paul [McKenney](#),

RCU AM: Any dependency if you load a pointer into an integer and back again?

PM: Only need to do that to adjust the bottom bits of the pointer, which goes outside of the standard. If we did allow things to be turned to an int, there would be very limited things to do with them anyway.

Patrice Roy: Are we going to be able to do pointer address arithmetic?

PM: Can do. Alternately derive from supplied classes. (Example in paper, figure 7)

VR: Do we want a traits parameter to inject logic? Callback pointer - replace with customisation point?

PM: Derive a fresh class

VR: Lot of work for small variation

SM: Traits are more composable.

More consistent with the standard library not to inherit from standard library classes.

Thread constructor Attributes

<https://github.com/viboese/std-make/blob/master/doc/proposal/thread/p0320r1.md> D0320R1

Thread Attributes.

Current wording does not give a way to know if things have worked. Expect updates for WG21 mtg.

There is precedence in the standard now to pass in error codes to a constructor = cf file system.

Is there any requirement that the application provides a complete attribute set?

A: Due to SG14 concerns, individual elements are preferred.

Q: Is there any reason why the attribute class can't be templeted

A: It's been thought that for things that require runtime support that's not a good way of going

A: Affinity could be done here, but that's too runtime.

Q: Implementation support leaves architecture centric naming differences

A: Standard names may be deined. Maybe feature test macros

A2: Also the example of attributes

Q: Question of initial thread state

A: Mail Patrice with suggestions

Q: Is there support for cache coherency

A: Mail additional suggestions to Patrice

C: In order to be useful, this requires a model of thread attributes - but that's SG1 intrusive containers

<https://rawgit.com/hfinkel/intrusive-containers-proposal/master/c%2B%2Bic.html>  Intrusive Containers

Q: Has the padding effects of having the intrusive parent at the start been considered

A: Question back to Hal

C: Ownership can be determined if eg a list of unique ptr.

C: This could be an implementation for a wrapper against the standard classes

C: The padding effects could be solved by an intermediate class that includes the data, then the intrusive base class

plf::colony/stack

Stack

C: Don't worry about whether the proposal meets the container requiriements. Theyre just common documentation standards.

Some arguments about whether stack is a real container, or just a different storage strategy on top of say vector. The argument is the implementation being separate to the usage.

C: Add an iterator and we get the segmented_vector that we need anyhow.

Sf 0

WF 11

N 9

WA 1

SA 0

likely/unlikely, clay

<https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/TnPaGDHaPwQ> 

C;Unlikely/likely

- show quoted text -

executor interface

Attributes for Likely and Unlikely Branches

https://ctrychta.github.io/branch_hints_proposal.html 

2.3 SG14 Goals and Scope and general discussion

Embedded Programming issues

Financial/Trading issues

Games issues

Missing low level algorithm , Patrice Unit move if no except

<https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/1B9pAcm-NE4> 

Cache control, ties, hot set, align as, pgo, Maxine

<https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/v9wIDLp3L-E>

Uninit memory algorithm, Brian ehler

<https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/8JWJBO8C1Js> D0247r0v0

https://www.google.com/url?q=https%3A%2F%2Feverard.github.io%2Fconstexpr_destroy.html&sa=D&sz=1&usg=AFQjCNEQIbEfmf26zHunb0W3GLVh4teYw A Proposal to Add

Constexpr Modifiers to destroy(_at|_n) algorithms

Flat container, Pubby

<https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/pD9hUHBIRSo>

<http://pubby.github.io/proposal.html>

Comparing virtual functions, Brittany, Paul, Scott

https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/_ouC6nKPnzI

Unroll attr, Nicolas

<https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/6Nq1MK8kdTg>

Explicit init list, Nicolas, Patrice, Brittany

https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/Z_FVbk_bs58

<https://docs.google.com/viewer?a=v&pid=forums&srcid=MTE5NTAwNjk0ODI0NDg0MTc0MjkbMTA3NzEwNzU1MzAyMTc3NjE2MDMBWVUxZ2x6WTIBd0FKATAuMQFpc29jcHAub3JnAXYy>

Generic event system, Giovanni

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.352.9084&rep=rep1&type=pdf>

<https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/n0QKPBMNWy8>

Stl and performance, what people prohibit abs why, Mathew, Patrice, Stefan.

<https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/-dSLcr7slfA> Paying for what

you don't use <https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/VeApd2U774U>

<http://gameangst.com/?p=320> <https://github.com/evmar/webtreemap>

<https://github.com/evmar/bloat>

Additional libraries, geometry, Eigen, tensor flow, Vincent

<https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/68fAAq0-71k>

Segmented stack, Mathew

https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/uT_Ibkdjn8I

https://11950069482448417429.googlegroups.com/attach/346358af9037d/segmented_stack.htm?part=0.1&view=1&vt=ANaJVrHcsjgRmixyNV0I6iSuJy6330cftE1bB4fI8IsVZE1iOn_xP1AehJwPcs1Vg2lnxcFs_N1M_fu1YI9NR24IhBr3mHdc0_N-Io9Zm6tUVBjosarWI

Vector size in bytes, Nicolas

<https://groups.google.com/a/isocpp.org/forum/m/#!topic/sg14/nstQlAc7wmI> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2016/p0122r2.pdf>

3. Any other business

Closed at 6:00 pm. Not able to discuss any other business including meta discussion items.

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 (past and future calls)

Next call : OCT 12 telecon

Oct 12 Telecon 2-4 ET (Oct 17 mailing deadline) Nov 6-12 C++ Std meeting
Meeting C++ Games Track: Nov 18-19, 2016, Berlin

Minutes for 2016/10/13 SG14 Conference Call

Minutes by Michael Wong

Start Time: Thursday, Oct 13, 2016, 10:00 AM US Pacific Time, 3 PM US Eastern Time

End Time: 12: 00 Noon US Pacific Time (duration: two hour)

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.

I will take notes for the calls, though it will also be recorded.

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Michael Wong

Carl Cook

Guy Davidson

Isabella Muerte

Neil Horlock

Ronan Keryell

John Mcfarlane

Sean Middleditch

1.2 Adopt agenda

Yes

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

Yes

1.4 Review action items from previous meeting (5 min)

2. Main issues (125 min)

2.1 CPPCON recap and future plans

1. next year could be in Courtyard marriott
2. time : ran out time , break for only 1 hour lunch, pick up papers that most complete first
3. Voting was too friendly Perceived by some: I think this is normal and we actually one paper
4. straying from our mandate: priority to wards out main mandate low latency or constitution
5. GDC 2017 is more informal meetup, not a formal SG14 because C++ std meeting is in Kona at the same time

2.2 Review paper mailings and SG14 position/feedback on any papers from the mailing: 30 min

jhs@edg.com

for paper numbers and for sending the final paper

1 Additions: Intrusive smart pointer, Isabella Muerte met with Bryce, he will represent, some changes, removed ability to adopt a ptr exclusively, all create ref count of 1 except Boost intrusive rep counter

A number please for Title : SG14, LEWG

2 inplace functions: Nicolas

Pending Nicolas

Neil and I will pitch in as needed

3 Fixed point real numbers, John [McFarlane](#)

working on revision to submit

I will represent

Setwidth:

SG14 LEWG

I will represent

4 Ring span, Guy Davidson

I will represent

SG1, SG14, LEWG

5 Hazard Pointers + RCU Maged + Paul + Michael

SG1, LEWG

6 Thread constructor attributes, Patrice

Not on

7 Intrusive containers, Guy Davidson

Hal Finkel presenting

10 Comparing virtual Functions , Scott Wardle (not present)

Not on

2 Non allocating std function, Nicolas

same as inplace functions

8 plf colony/stack, plflib.org Matt Bentley

Not on but actively worked on
Sean Middleditch will present
9 Likely/unlikely
Clay updating
Does he need representation
move_if_noexcept algorithms, Patrice Roy Maxine not present
constexpr destroy, Brian Ehlert (not present)
unroll attribute, Nicolas
Explicit initializer list, Nicolas
Generic event system, Giovanni (not present)

Sean: flat container
flat map done, then flat set, then multi version

Ronan
Accesor paper
subsumed to Array_ref
working on Khronos

2.3 Other papers aimed at Issaquah meeting: 25 min

Neil

1. timing precision to micro seconds due to financial requirement to have precise time associated with trades, for compliance and regulatory forensic tracking
current std: Chrono is only precise to milliseconds?
actually MiFid2 requires nanosecond precision, which needs to be linked to atomic clocks
Carl: Optiver is interested in this
Precision: how many of those digits are meaningful
accuracy: how far is it off by the time you read it
need to report to microsecond +/-

2. Heterogeneous executors
nanostream, executed on mini cpus on fpga fabric, build custom cpus
takes C++ and converts it fpga nanocores
nano cores, gpu cores, xeon phi
best performance distribution load balancing
Michael with discuss this at Codeplay

3. Michael presented current design of Minimal Executor proposal

2.4 Future F2F meetings:

- 1. HPC C++ at Lawrence Berkeley: Oct 17, 18, Oakland**
2. Meeting C++ Games Track: Nov 18-19, 2016, Berlin
3. Embed.io : Feb 18, Bochum

2.7 future C++ Standard meetings:

[N4571](#) 2016-11 Issaquah meeting information
[N4573](#) 2017-02 Kona WG21 Meeting Information
2017-07-10-2017-07-15: University of Toronto/Canada
<http://isocpp.org/files/papers/N4607.pdf>

3. Any other business

Reflector

<https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/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

Next call is a F2F meeting : Nov 7-12 C++ Std meeting Issaquah

Tentative next call is Dec 14

