Document Number: P0686R0
Date: 2017-06-19
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 2017/03/09-2017/06/14

Contents

Minutes for 2017/03/09 SG14 Conference Call

Meeting minutes by Billy Baker
Notes by Billy Baker. Thank you.
Note that recent clarification of ISO directive seems to forbid revealing clearly speaker in a public forum. There is a concern that people would not be able to speak freely if that were the case.
This is why we have decided to not identify explicitly the speaker in these notes any more, and we no longer can post committee discussions verbatim from the Kona meeting.
I am sorry as this seems to reduce our transparency. But if seems best to see what follows through from this before acting differently.

---+ SG14 Telecon March 8, 2017

Michael Wong

Billy Baker (scribe)

Ronan Keryell

Calos van Rooijen

Odin Holmes

Andreas Fertig

Arthur O'Dwyer

Emil Fresk

Philippe Groarke

Wouter van Ooijen

Tal Lancaster

---++ Review Action Items

No action items

---++ Kona Review

Coroutines TS was pushed out

This would be the stackless version of coroutines

Other styles of coroutines include stack based which is still be worked on

The modules TS did not get pushed out, there was no consensus on the current wording

A small C++ steering committee (Bjarne, Michael Wong, Howard Hinnant, Daveed Vandevoorde, Beman Dawes) was also created, the small group might request work to be performed in particular areas

The small group did discuss embedded including dynamic memory allocation, real-time, exceptions

---+++ Paper Review

Hazard pointers and RCU was discussed in Kona, compared with Google proposal

Proposals look good, all 3 were liked, all 3 could be possible TS material

Executors received good feedback in Kona, possible smaller set of customization points

Fixed-point is being merged with another proposal based on Kona discussions SG6

Colony was well-received in Kona, a name change could be needed

likely/unlikely was discussed in Kona

(feedback for committee) More information about ring span discussion would be nice such as concurrent issues in ISRs

(Action for Michael) Odin and Lawrence Crowl should discuss

Google proposal on RCU is from Geoffery Romer and Andrew Hunter P0561R0, RAII, high-level API

Example is reading a configuration file

---++ Embedded Discussion

A partition of the embedded space is needed

Eventual entries could be added to the C++ Core Guidelines / GSL for embedded

A guideline to not use volatile in embedded could be similar to the general guideline of not using raw pointers

Some sort of wrappers for volatile for correct usage could be useful

Generally, we say not to use volatile anywhere, prefer atomics

For the embedded space, volatile with registers/mapped memory is needed to prevent read elison

Memory fences can address other classic volatile usage

Volatile can be compiler specific in that some embedded compilers may generate full memory fences, gcc is different

Self-assignment of volatile has been broken in some compilers

C++ Core Guidelines does have CP.8 and CP.200 on volatile

Business logic should not use volatile based on note for CP.200

Embedded domain also has a slightly different view of global objects

An error policy that defaults to throw would be fine if it could be extended to do something else for an embedded domain

(Action for next meeting) Write a guidelines for embedded specific issues.

A guideline about putting the floating point type in a policy would be good.

---++ Complexity in the Standard Library

Is it worst case?

There are some open issues on this

Would prefer a worst case

The question is about determinism, push to a deque is usually constant

How do you prove that when using the standard library that you will never take longer than X amount of time?

Not sure of a path forward, just voicing the concern

Reviewing C library implementations is easier than reviewing C++ standard library implementations

Not sure what the guarantees would look like

Analysis of generated assembly which does not have indirect calls to find longest path is one way

Cache effects may not be available in such an analysis

Crypto code is another area where determinism is a concern

Decorations such as noexcept let the user know some of what a function is doing, nothing like that exists for blocking/locking/allocating nature

The big-O issue is a problem but not sure how to solve it, may just need to use other containers

---++ Next Call

April 12, 2017

Will there be a face-to-face at ACCU?

Not sure, but probably not

What about C++Now?

No

Minutes for 2017/05/10 SG14 Conference Call

Meeting minutes by Michael

Wouter, Carlos, Guy Davidson, John McFArlane, Klemens, MAteusz Pusz, Patrice Roy, TOny Tye, Billy Baker, Michael, Mathew, Odin, + a few others that joined after roll call
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
-github
no conclusion that it is like a boost repository
anyone can setup a repository within the github to fit their personal needs
welcome Paul to make github to be more presentable as a separate repository
-future C++ meetings
CPPCON June 11 deadline
-F2F
ACCU: Herb did closing keynore

C++ paper deadline June 19 after June 14th meeting

2.2 Review Executors explanatory paper

Hi SG1 and other folks interested in executors,

We have written a paper explaining the design of our executors proposal (P0443 [1]). In it, you will find our vision for executorsin C++ as well as the rationale informing our design choices. We plan to keep this document synchronized with our proposal as it evolves in order to maintain a record of the design.

After reading through this first rough draft, we'd appreciate your feedback on the design of the proposal. Since it is still a draft, please don't circulate this paper for now.

To allow us enough time to incorporate your feedback into a revision of our proposal for the Toronto meeting, please submit your feedback by Thursday, May 11.

Thanks in advance for your help!
How do locks and synchronizations interact with executors?
Can this work with a precise timing executor for I/O devices?

Will changing a flat memory model require change in language?
Changing to different memory regions. hierarchical memory, send to Odin.

[1] wg21.link/P0443
And here is the spec paper that was presented in Kona:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2017/p0443r1.html

2.3 Embedded domain discussions

repository of Embedded domain libraries
Kvasir Library with meta programming C++: some assumptions on communication, thread, a collection of library
Safeint
Fixpoint
Boost Unit
coroutine on Cortex
core guidelines exposed after seeing these library in action

not call it embedded, but call it performant , or safe, or real-time, or memory constraint so that it cross domains
with the goal of standardization in STL

they have differnt characteristics to STL:
event based instead of thread based
does not assume infinite memory
may not behave well with exceptions
safety critical
determinisitic vs non-deterministic

Separation of embedded concerns
1. hard real-time
2. constrained memory

Interest in starting a safety security SG from WG23, will do an evening session in Toronto

AI: what to do?
start something minimal, wide audience, contrast it to STL,
explore where the tools can take us
things to put in the library:
1. fixed point
2. rings

3. vector no growth with small buffer optimization, dynarray
4.fixed maximum sized flexible array

a new repository for tools that people find useful and share, with the aim of getting a library together. (without aim of production quality, or driving for standardization)

2.4 Papers and proposals
1 Additions: Intrusive smart pointer, Isabella Muerte

2 inplace functions: Nicolas

3 Fixed point real numbers, John McFarlane
morphing into numeric types library, for safe int, precise rounding mode
to match Lawrence's proposal
plan to write joint papers on the new combined approach
will talk at C++now next week

4 Ring span, Guy Davidson
Concurrent ring span deferred to concurrent queue,
Patrice presented

5 Hazard Pointers + RCU Maged + Paul
HP moves to wording, RCU contineus work
Working wordings
6 Thread constructor attributes, Patrice

7 Intrusive containers, Guy Davidson
backlog
Guy working on this

8 plf colony/stack, plflib.org Matt Bentley

Guy working on this as well
9 Likely/unlikely
presented and a revised paper is asked for

10 Comparing virtual Functions , Scott Wardle

2.5 Future F2F meetings:
C++Now: May 15-19
John M presenting on fixed point and the idea of the repository; Odin attending

CPPCON Sept 27

2.6 future C++ Standard meetings:
2017-07-10-2017-07-15: University of Toronto/Canada
https://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
Guy to cross post SG13 paper on 2D api on Cairo APi to try address some of the issues with stateful to stateless
John M to look

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

June 14: pre-review before June 19 deadline
Have a look at progress of the repository, possibly present this idea in Toronto.

5.2 Future meeting

June 14
July 12 (cancelled as it is same day as Toronto C++ Standard meetign)
Aug 9
Sep 13
Oct 11 (before DST switch)

Minutes for 2017/06/14 SG14 Conference Call

Minutes by Michael Wong

1.1 Roll call of participants

Allen Deutsch, Billy Baker, Brett Searle, Guy Davidson, John McFarlane,Klemens Morgenstern,Patrice Roy, Paul Bendixen, Michael Wong, Arthur O'dwyer, Philippe Groarke, Shay Morag, Fgarelli, Alexander Arsenault

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

2. Main issues (125 min)

2.1 General logistics

-github, Conan

-future C++ meetings, talks

-F2F

Iibrary mentioned at cppnow,

Conan is a dependency manager, add a line to a file, add a Line to your make,uses jfrog

Dependency managers is a touchy issue for c++

If you need a paper number: jhs@edg.com

As a reminder, the mailing deadline for the pre-Toronto mailing is 2017-06-19 at **14:00 UTC.**

Also as a reminder, the paper guidelines and mailing procedures can be found here:

https://isocpp.org/std/standing-documents/sd-7-mailing-procedures-and-how-to-write-papers

I am sending this reminder out early because I will be only checking email sporadically in the two weeks leading up to the mailing deadline.

I'll be able to assign paper numbers until the mailing deadline, but it would make my life easier if you could get requests in before June 1st. After that date it may take a day or two to get a response.

Thanks,

John.

2.2 Paper reviews

2.2.1 Registers for C++

Klemens D. Morgenstern

klemens.morgenstern@mw-sc.de

Registers presented by Klemens

Do we need to support signed bit fields?or is this for bitfield compatibility
Much of the paper is similar to bitfield
Good use case is 12bit ADC in micro controller
Registers makes it even more closely related to micro controller
Can we have signed or unsigned, and pick a default
How do I interpret these bits as a number
Removing a potential useful features, despite the sharp edges, is not a good way to go
Default to signed int should be written out
Not having defaults will have fewer surprises
Save these as open questions at the end of the proposal
Standard byte is a non integer representation of single byte
Can we contrainted existing bitfield in implementation
What about undesirable padding
Replacement or substitute bitfield, depends on how people use it
Alternative to register , but register makes sense in embedded world
Can it be library solution? But need compiler warnings.Odin Holmes Kvasir is similar to a library solution
May be try it first?
Library feature will be a tuple, angle brackets, expression templates
Can we take address?
anyone wants to help, feedback or co-author or proxy?
Arthur to send comment and feedback, may be co-author?
Get a paper number from John Spicer

2.2.2 slot_map Container in C++

Allan Deutsch

allan.d@digipen.edu
guarantees O(1) for insert
on erase element, increments generation counter
design issues are hwo to implement udnerlying container
like deque: fixed sized arrays: can't iterate over it with pointer, iterators jumps between tables, but get constant time insert even on allocation of new chunk of memory
indirection table, gets gap in memory, but faster lookups
these are the 2 that requires feedback
can it be a container adapter
iterator should be fine,

does not solve issue of gaps in memory, worst iteration time,
can we have AOS, SOA conversion, but no one knows how to get that
suggest we use indirection table, then specify container as a template parameter, that keeps all elements packed towards the front, and no gap in memory
is there another proposal from games side on flat map that also uses holes in memory and iteration using a pointer array, but not mentioning keys
flat map is an ordered container, this does not specify ordering, solves different problems, map optimized for people wantign contiguous memory
container adapter or using its own udnerlyng type
some suggest an adapter to punt on the design descision on AOS/SOA
or just pick one, SOA is what it seems we are coming close to
possible to do SOA with a reflection proposal

Please proceed to get a paper from
jhs@edg.com

2.2.3 any other proposal for reviews?

2.3 Embedded domain discussions

2.4 Other Papers and proposals

1 Additions: Intrusive smart pointer, Isabella Muerte

Not present

2 inplace functions: Nicolas Now Carl and Patrice

Carl might attend Alburquerque

3 Fixed point real numbers, John McFarlane

minor interface update to setnum paper

4 Ring span, Guy Davidson

Concurrent ring span deferred to concurrent queue,

now thrown back to just ring span, Guy coming to Toronto
bounced back and forth
Paul will help,
Arthur saids we should have iterators

5 Hazard Pointers + RCU Maged + Paul

wording being worked on

HP moves to wording, RCU contineus work

6 Thread constructor attributes, Patrice

plan to have new paper for Monday
working with Billy

7 Intrusive containers, Guy Davidson

with Hal, no feedback

8 plf colony/stack, plflib.org Matt Bentley

Have implemented a lot of changes based on Jonathan Wakely's feedback post-review. Still have work to do on the proposal pre: toronto.. seems Patrice will present it

9 Likely/unlikely

Clay's paper was reviewed and encouraged for another version

10 Comparing virtual Functions , Scott Wardle

Not present

11. Heterogenous computing: Channels and Managed pointers

a set of papers to encourage heterogeneous computng post executors
hairy bits remain in executors
but executors does not give heterogeneous computing,
channels and managed pointers can do it without massive memory model change
good for distribute computing, involving HPX LSU

2.5 Future F2F meetings:

CPPCON Sept 27

SG14 will be in courtyard marriott

2.6 future C++ Standard meetings:
2017-07-10-2017-07-15: University of Toronto/Canada

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

Hotel deadline passed june 9, will be sending an email about late hotel bookings

3. Any other business

Access to reflector needs SG or attendance
Arthur O'Dwyer
Guy Davidson
Allan Deutsch
John McFarlane

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: Aug 9

July 12 (cancelled as it is same day as Toronto C++ Standard meeting)
Aug 9
Sep 13
Oct 11 (before DST switch)