Document Number: P0360R0
Date: 2016-05-30
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 2016/02/17-2015/05/25

Contents

Minutes for 2016/02/17 SG14 Conference Call

Meeting minutes by Michael

1.1 Roll call of participants
Sunil, Billy Baker, Lee Howes, Brent friedman, Tony Tye, Michael Wong, Hartmut, John Mcfarlane, Al Grant;

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)
1.4.1. All: Consider attending SG14 F2F meeting hosted by Google at GDC 2016
Tentative: Monday March 14
https://groups.google.com/a/isocpp.org/forum/?fromgroups#!topic/sg14/qnbWDK9t0gY
This is the finalized date and location.
Have about 20 people signed up. Will have an evening social after the meeting.

1.4.2. All: Consider attending Jacksonville Fl, C++ Std meeting Feb 29-Mar 5.
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4555.pdf
This will be the 2 meetings for last chance for C++17. So C++17 items take precedence, Non C++17 items will not likely be looked at until Wed/Thurs/Fri/Sat.

2. Main issues (125 min)

2.1 Review SG14/SG1 logistics, for upcoming telecons and meetings: 5 min

2.2. Activision/Blizzard proposal on Standardized packaging: 45 minutes
Authors did not call-in. Skipping this section.

2.4 Review proposals for SG14 submitted for Jacksonville meeting: 45 min

Papers needing proxy by this group.
Hartmut on HPX
John Mcfarlane : fixed point: Michael, Lawrence: minor api changes, resizing strategy; not for C++17; waiting on Lawrence's proposal implementation.

Brent Friedman: uniniialized memory; with examples in industry; Billy Baker; LEWG approved; LWG; C++17;
Michael: Khronos SYCL; Concurrency toolkit; consume ordering; RCU; hazard pointers; SG5 TM position

We won't have time to discuss in detail, but we can go over major questions and seek opinions and feedbacks

P0037R0 Fixed point real numbers John McFarlane LEWG SG14/SG6: Lawrence Crowl
P0038R0 Flat Containers Sean Middleditch LEWG SG14: Patrice Roy
P0039R0 Extending raw_storage_iterator Brent Friedman LEWG SG14: Billy Baker
P0040R0 Extending memory management tools Brent Friedman LEWG SG14: Billy Baker
P0041R0 Unstable remove algorithms Brent Friedman LEWG SG14: Billy Baker
P0059R0 Add rings to the Standard Library Guy Davidson LEWG SG14: Michael
P0130R0 Comparing virtual functions Scott Wardle, Roberto Parolin EWG SG14: Michael
P0125R0 std::bitset inclusion test methods: Michael/Walter

2.3 Ongoing topics placeholder (which we won't likely get to but will schedule)
Exceptions/RTTI
SIMD vector and Matrixes
Allocators
GPU/Accelerator design
Array View and Bounds checking

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
Next call : Feb 24

Dec 9: Nvidia's Agency With Jared and Mike Garland - DONE
Jan 13: 1st vector/SIMD call - 2 hours: Pablo - Done

Jan 20: AMD's HCC compiler with Ben Sander - 2 hours
Jan 27: 2nd vector/SIMD call- 2 hours:Matthias proposal? Other proposal? Cancelled
Feb 3: LSU's HPX runtime with Hartmut Kaiser/Andrew Richards - 2 hours- Done
Feb 10: possibly 3rd vector/simd call: Matthias Kretz/Joel Falcou, Matthias Gunard- Done
Feb 12: C++ Std meeting mailing deadline (no call) - all papers submitted
Feb 17: Uniform install/Packaging proposal
Feb 24: low level bit manipulation proposal

Minutes for 2016/02/24 SG14 Conference Call

Meeting minutes by Michael

1. Opening and introductions

1.1 Roll call of participants

Sunil, Ben deane, Guy Somberg, Brian Fitz, Lee Howes, Iwo, Billy Baker, Michael Wong, Guy, Neil Horlock,

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)
1.4.1. All: Consider attending SG14 F2F meeting hosted by Google at GDC 2016
Tentative: Monday March 14
https://groups.google.com/a/isocpp.org/forum/?fromgroups#!topic/sg14/qnbWDK9t0gY
This is the finalized date and location.

1.4.2. All: Consider attending Jacksonville Fl, C++ Std meeting Feb 29-Mar 5.
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4555.pdf

2. Main issues (125 min)

2.1 Review paper mailings and SG14 position/feedback on any papers from the mailing: 30 min
https://groups.google.com/a/isocpp.org/forum/?fromgroups#!topic/sg14/yymPjDyJD60
At Jacksonville, Convener will be asking for feedback on admitting the following major work projects into C++17

File System TS

Parallelism TS1

Concepts TS

Library Fundamentals TS1

as well as:

Special Math IS

If you have any specific feeling for these and want to convey that through proxy, please discuss at this meeting.

2.2. low level bit manipulation proposal: 30 minutes

P0237R0	On the standardization of fundamental bit manipulation utilities	Vincent Reverdy, Robert J. Brunner

HPC exascale computing, data structures are implemented as hash tables, need it to be optimized, so need bit manipulations
small library to manipulate single bits
today we use reference member or dynamic_bitset to access bits in vector,
other proposals limit to std::bitset
we use bit_iterator, bit_reference, bit_pointer, bit_value
slide 19:
position of a bit within object:
2 options: bit corresponds to that in memory, or binary value
slide 21: bit_reference
unsigned only, signed is platform dependent, can always reinterpret_cast

are a bit and a bool the same? wierd behaviour for bits
pg 11 of paper
or make bit exactly like a bool
or a 1 bit long unsigned integer

pg 30, fig 5
bit_value disambiguate what a bit is and a

•

std::bit_value

emulating an independent, non-referenced bit

- std::bit_reference

  emulating a reference to a bit

- std::bit_pointer

  emulating a pointer to a bit

- std::bit_iterator

  , based on the preceding classes and emulating an iterator on bits

can pick and choose which part

pg22: Guy: bit_pointer; what is bracket?
if n is bigger then number of bits, then mod to arithmetic
hmm, no bracket operator work like that, if you go off the end of underlying type, may throw exception; should get feedback on it
ben: bit_pointer and bit_reference, pulling out non-aligned boundary for a network stream, array of bits , use that memory to pack a bitstream, pull things out of there, want to convert from a bit_pointer to an unsigned int
call dereference operator on bit_reference, on pg 20, line 53-54 on current position of bit

pg 24: layout of bit_iterator

2.3 C++ packaging proposal: 30 min

P0235R0	A Packaging System for C++	Guy Somberg, Brian Fitzgerald	2016-02-05	2016-02	Evolution, SG14

time consuming to consume C++ packages
like OpenSSL, download then install perl, should be trivial to use these libraries
need a standardized layout for packages
section 3: pg 3: #using package syntax
compile it into your code and include whatever header is needed, download and unpackaging is QOI
not meant to replace gmake, cmake or build systems
take one part of it so that it is standardized to allow easy integration
this is not the same as module inclusion
other options can be configured using options for other architectures
#using path: paths can be file or directory names
key idea is that packages has a standard layout
3.2 shows the layout
adds the include dirs to a temporary include paths,
3.4 shows packages used with modules
have done implementation in clang
Manifest file picked json as representation, clang already have yaml parser
4. A Standing document
somethings cannot be put into the standard, but we can use SG10's mode
4.3:
pragma package_source
where to get the package and what protocol
4.3.2: pragma package_version_scheme
rules on what version is allowed
4.4 package build settings
options for build
5. the rest is just alternative design choices

why pragmas are needed? 5 customers;
using
authoring
developing compiler
committee members wg 21 and 14
preprocessor is needed because we are including headers, preprocessor clang preproc actually calls the lexer
so when packages need to include headers we need to be in the preprocessor
options; could skip headers, would need to write a wrapper
adapt modules, but wg14 would also not be able to accept this
or force
using to include headers
Michael: #using should bridge on Google's module which must also process macros
Neil: investment banking also need this for standard package inclusion
need namespace or naming for this too
Brian: need a hierarchy naming, we punted
Michael: for investment banking, would need certificate signing to make sure an inclusion is safe

Ben: doubling down on section 6 on why this does not compete with cmake gmake,
use case for zlib and openssl and show what it be like
OK
Boost would be a good one
.

2.4 Review SG14 proposals: 25 min
We won't have time to discuss in detail, but we can go over major questions and seek opinions and feedbacks

want a clean way to printout the fixed point real numbers
%f, %g, and cout-like
Brian: Blizzard has something like this; does not want rounding

Minutes for 2016/05/25 SG14 Conference Call

Minutes by Michael Wong
Brent Friedman, Nicolas Guillemot, Ronan Keryell, Scott Wardle, Sunil Srivastava, Michael Wong, Nevin Liber, Guy Davidson, Tom Rodgers, Carl Cook, Hans Boehm, Rene Rivera
Meeting recorded.
25 limit on the phone line

1.2 Adopt agenda
Approved.

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

Approved.

1.4 Review action items from previous meeting (5 min)
Brent , Ronan, Michael going to FInland.

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
Will there be an update for Oulu?

2. fixed point: John M, Lawrence, Marco Foco
No replies from Lawrence, working on a type similar to Lawrence. Please put in a paper, as Lawrence sometimes cannot attend international meetings.

3. ring span: Guy, Arthur
Don't think we can update it. Aim for the next meeting. Plan on wordings.
4. Low level bit manip: VIncent
Vincent not on.
5. uninit memry algo: Brent Friedman
Looks good. Plan to submit.
Guy likes this from Creative Assembly
EASTL has most of them
it is everywhere.
Brent will be in Oulu.

6. datapar (SIMD) : M Kretz, M Gaunard, Joel Falcou
I saw it in STAC London.
7. Comparing virtual fns: Scott Wardle/Sunil
Plan to send something to form a paper
Sunil and Scott will not be there.
Scott to send latest and Michael will share it on google doc
8. Thread constructor attributes: Patrice
Patrice not here
9. install distribution packaging: Brian Fitzgerald
Have not heard any updates.
10. unstable move: Brent
Plan to get it into Boost, but Boost is discussing 2.0,
no plan for a paper yet
11. Class for status/Optional: Lawrence
12. utility class to represent expected monad: Paul Hampson, Vincent Botet
13. hazard pointers and RCU for lock-free programming: Michael
will be updated and reviewed in SG1
14. 2d display: Michael M
Added to SG13.
Is this related Cairo?

15. Small vector: Nevin
not planning to present in Oulu
got enthusiastic support
inline vector not as much

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. Heterogeneous device support (Michael, Hartmut)

b. EH lite (see below)
c. CPU/cache/memory affinity (Neil, Michael, Guy)

d. memory allocation (Guy Davidson)
pitching something for GDC 2017, Carl and Nevin, and Rene mentioned composable allocators
a future presentation for new form allocators, after Aug 7, say Sept
1. multiple small vectors: Gonzalo BG et al
covered by Nevin's proposal, up to indefinite number of elements
inlien vector does not allocate, up to n elements
2. Exception lite and swift-like exceptions: Sunil, Patrice, and Sean Middleditch

sent a page on EH proposals,
plan to write a paper for Finland, but not present it,
to start disscussion
but wait till Issaquah to present it
share write it on google doc
3. basic inplace function: Nicolas Fleury, Sean M, etc al, Cark Cook
 Carl has started a paper, will keep working on it, not for this meeting
4. Interprocess communication: Shaun Croton et al
 HFT all doing the same thing, IPC with shared memory, many spin a core just to wait for a message
hopefully by attacking threads first, we can solve some of the similar problems with processes
big change to C++
Carl and Neil both interested
5. hot set: Brent Friedman
not a proposal yet, still looking feedback, seems to be a lot of interest
HFT wants ways to keep things hot, but using attributes
6. Accessors: Ronan, Lee
C++ assumes flat address space, not always the case
accessors describe on the hardware level how to access data, read/write/ nontemporal access, or use special HW bus for high-speed i/o
mroe finely describe how allocation and ptr access is done, ring span can be implmented with an accessor to describe modulo addressing
Michael also writing on named lambdas, and async functions returning futures
7. std::stack: Matthew Bentley
8. plf::colony/stack: Matthew Bentley
Not here.
9. FAst associative container: Allan Deutsch
Allan not on.
Not ordered.
10. Alternatives to traverse linked data structures: Marcelo Zimbres
Not on.
11. Delegates for simulations: Miodrag Milanovic
Not on.
12. Dynamic/runtime concepts: Zach Laine, Andrew Sutton
Not on.
13. width/set width: John M
 Part of the fixed point proposal. like to break it out as it is smaller piece. Is ther a use case for this?
Might try to put something together for deadline.
14. explicit initializer list constructors:Nicolas Fleury
Not on.
15. popping move-only types from priority queue: Ben Deane
Not on.
16. affinity, locality and hints: Neil Horlock
Have a group now for it: Carl, Neil, Guy, Michael

17. intrusive containers (Guy)
big push from London STAC
Boost author may be able to help. Michael will contact him.
18. FPGAs (Ronan)
Is there any workload in C++ for fpga? Some.
Financial people seems to use it a lot, but use it as BSD sockets
19. lock-free queues: michael/lawrence
Michael interested in this
Carl likes to help
20. half precision: Sean M
Not here.

2.5 Talks we proposed to CPPCON:
1. SG14 update
2. Lock-free Cocurrency Tookit for Hazard pointers and RCU
3. Heterogenous computing in C++
4 SPMD programming
5.

2.6 Future F2F meetings:
1. June 13 STAC NY, looking for a host for Tues June 14th: 10-11:30
2. Monday June 27: HFT for C++ Amsterdam hosted by Optiver, Carl Cook
3. CPPCON 2016: Wed, Sept 21: 8:30-5 pm
4. Meeting C++ Games Track: Nov 18-19, 2016, Berlin
SG14 update

2.7 future C++ Standard meetings:

N4570 Oulu Meeting Information
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
  - o 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
  - o 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)
Michael: talk to Boost.intrusive author

5. Closing process

5.1 Establish next agenda

Please propose specific proposal reviews for June 1/8, else I will cancel those meetings.
We will schedule an Embedded-focused discussions after the Oulu meeting.

5.2 Future meeting
Next call : June 1 (back to 2-4 ET) unless there is no specific proposals to discuss

May 25: review all papers pre-mailing deadline (May 30)
June 1: proposal reviews
June 8: proposal reviews
June 20-25: C++ Std Meeting Oulu, Finland
Closed 4:20 pm ET.