P2057R0: SG14 SG19 Past, Present and Future Status

Date:	2020-01-13 (Prague)
Project:	ISO JTC1/SC22/WG21: Programming Language C++
Audience:	SG14, SG19, WG21
Authors:	Michael Wong, Ben Craig, Paul Bendixen, Matthew Bentley, Inbal Levi, Rene Riviera, Steffan Tjernstrom, Mark Hoemmen, Ronen Friedman
Contributors:	SG14, SG19
Reply to:	**fraggamuffin@gmail.com**

Revision History

Revision	**Description**
P2057R0	Initial version based

Introduction

Given that SG14 Low latency, Games, Finance, Embedded, Simulations SG has been in existence since 2015, and the recently created SG19 Machine Learning was recently created in 2018, it seems it is good to see what features we have developed for C++, what features we are working on now, as well as what features we are interested in.

In general, it is good to show periodic status of these SGs so people outside to this group can have visibility from outside, and so that we can see what has been done, and what we are aiming for and such that people can comment and join the group's effort to increase collaboration.

We continue to hold monthly zoom telecons

- SG14: 2nd Wednesday 2-4 ET

- SG14 Linear Algebra: 1st Wednesday 3-5 ET
- SG19: 2nd Thursday 1-3 ET

This table was built by SG14 and SG19 group based on our group recollection. It includes not just features, but defects and issues of interest to the group.

As a summary:

- 2 SG14 features have been adopted into C++
- 7 SG14 features are now in progress
- 3 SG19 features are now in progress
- 2 SG14 features are stalled, but are being restarted
- A number of proposals are starting

This table includes features/issues that are started outside of SG14/19, but are of interest to the domains of these groups.

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

The table is embedded as as follows:

Feature/issue/defect	Author/proposer	Paper	Reason	Status	Remarks	Domain	Github	Reflector	Related Papers	
memory management	Brittany Friedman	https://wg21.link/P0040		Adopted in C++17		Games				
likely unlikely	Clay Trychta	https://wg21.link/P0479		Adopted in C++20		Finance				
Affinity	Gordon Brown et al	https://wg21.link/p1436		SG14-SG1-LEWG		All				
Topology discovery	Gordon Brown et al	https://wg21.link/p1795		SG14-SG1		All				
Freestanding Library: Easy Utilities	Ben Craig	https://wg21.link/P1642		SG14-LEWG		Embedded			P0829R2	
Freestanding Library: Rewording Status Quo	Ben Craig	https://wg21.link/P1641		SG14-LEWG		Embedded				
Freestanding Language	Ben Craig	https://wg21.link/P1105		SG14	D2013	Embedded				
Linear Algebra	Guy Davidson, Bob Steagall	https://wg21.link/P1385		SG14-LEWG		LA				
Linear Algebra BLAS	Mark Hoemmen et al.	https://wg21.link/P1673		SG14-LEWG		LA				
z-order_curve data packing	Jeremy Ong	New idea		SG14	This could be implemented as an mdspan (P0009) Layout. Layout	Games		https://lists.isocpp.org/sg14/2019/12/0274.php		
executor with embedded scheduler	Detlef Vollmann	New idea		SG14-SG1		Embedded		https://lists.isocpp.org/sg14/2019/12/0277.php		
Intrusive pointers	Isabella Muerte	https://wg21.link/p0468		SG14-LEWG	http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2019/p1351r0	Embedded				
Deterministic Exceptions	Herb Sutter	https://wg21.link/P709		SG14-LEWG		All			P0323R3	
status_code and standard error	Niall Douglas	https://wg21.link/P1028		LEWG	Blocked by NB comment processing	All				
Function failure annotation	Jens Gustedt and Niall Douglas	http://www.open-std.org/jtc1/sc22/wg14/www/do		WG14	Awaiting prototype C compiler	All				
Colony	Matthew Bentley	https://wg21.link/P0447		SG14-LEWGI	Jonathan Wakely and Guy Davidson, hard to name	Games, Finance, HI	https://github.com/mattreecebentley/plf_colony			
Ring Buffer	Guy, Dan Raviv, Matthew Butler	http://wg21.link/p0059		stalled in SG1	would std::destroy_at etc. change the whole pop traits, iterators?	Games, Finance			P1976R0	
Object relocation ("trivially relocatable")	Arthur O'Dwyer	https://wg21.link/P1144		SG14-LEWG		Embedded				
fixed_capacity_vector(fixed sized containers)	Gonzalo Brito gadeschi	https://wg21.link/P0843		LEWG		Embedded/Games				
flat map	Zach Laine	https://wg21.link/P0429		LWG	originally SG14 Guy Davidson proposal	Games, Finance				
member layout control	Rene Riviera	https://wg21.link/P1605		SG14-EWG	RF: we should folow P1112 & P1847. And we better come up with a better alternative for specifying struct layout	Games, Finance, Embedded				
numerics	Alexander Zaitsev	https://wg21.link/P1889		SG6 lead		Finance			P1890R0	P1719R2
Legendre polynomials	Neil Horlock/QUB	New idea		SG6/19						
improving debug builds	John McFarlane	https://wg21.link/P1832		SG14-SG15		Games				
Exception Handling Size benchmark	Ben Craig	https://wg21.link/P1640		SG14-DG		Embedded				
Exception Handling Speed bechmark	Ben Craig	https://wg21.link/P1886		SG14-DG		Embedded				
Statistical Functions	Richard Dosselman et al	https://wg21.link/P1708		SG19		ML				
Differentiable programming	Marco Foco, Max Rietmann, Vassil Va	New idea		SG19	P-Number assigned: 1707	ML				
Numerical Differentiation	Marco Foco, Vassil Vassilev	New idea		SG19/SG6		ML				
Graph Data Structures	Phil Ratzloff et al	https://wg21.link/P1709		SG19		ML				
Physical Units	Mateusz Pusz	https://wg21.link/P1935		SG6/16		All				
C++ exceptions and alternatives	Bjarne Stroustrup	https://wg21.link/P1947		WG21		All			R0824R1	
Portable Optimization Hints	Timur Doumler	https://wg21.link/P1774		EWG		All				
Named Return Value Copy Elision	Anton Zhilin	New idea		Proposals list	https://gist.github.com/Anton3/594141354ff9625db0b85775799312	All?				

Thread Costructor Attributes (controlling new threads' stack size)	P0320: Vicente J. Botet Escriba;
	P0484: Patrice Roy, Billy Baker, Arthur O'Dwyer;
		https://wg21.link/P0320R1
		https://wg21.link/P0484r1
				Dormant?	The ability to control creation-time-only aspects of new threads is mandatory in some real-time projects. The exact mechanism we choose is not really important - but we should present a solution.	Embedded				See also: Patrice's presentation https://www.youtube.com/watch?v=iDztwNhIVVM

PI mutexes	none?				std::mutex has no standard facility to set priority inheritance, making standard mutexes unusable in multithreaded real-time systems. (and - once we're at it - we should add P ceiling, too)	Embedded				
Fixed-capacity type erasure (inplace_function, in	Arthur O'Dwyer	SG14 github only								
Fixed-capacity type erasure (function_ref)	Vittorio Romeo	https://wg21.link/P0792		LWG	probably C++2b					
Fixed-capacity data structures (fixed_capacity_ve	Gonzalo Brito Gadeschi	https://wg21.link/P0843		LEWG	see also fixed_ring from https://wg21.link/p0059r0					