FEBRUARY 13TH, 2025

NO, INPLACE_VECTOR SHOULDN'T HAVE AN ALLOCATOR

NEVIN LIBER
nliber@anl.gov

PRESENTATION
P3636R0

PAPER
P3581R0

LEWG LOVES TO RE-LITIGATE

LEWG LOVES TO RE-LITIGATE INPLACE_VECTOR

EXAMPLE: BAD_ALLOC, AGAIN

Wg21telecons2024/P0843#Library-Evolution-2024-01-30

- Agreed upon in Varna
- Almost re-litigated at Library Evolution Telecon 2024-01-30
  - No new technical information

- POLL: We want to revisit the status quo in the paper: “Inplace_vector throw a “bad_alloc” when exceeding max_size

SF	F	N	A	SA
3	4	2	5	4

- Outcome: No consensus for a change

TOKYO & P3160R0

P3160R0 LEWG LITIGATION

Wg21tokyo2024/P3160

- We care about this case
- It should be a different vocabulary type, such as basic_inplace_vector<T, N, A>
- Doing this in inplace_vector makes it an overly complicated type for the vast majority of users.

- It is not worth the compilation costs, as the vast majority of users will not use this feature
  - Sidebar: P3062R0 C++ Should Be C++ supports this claim with statistics on the usage of PMRs

Query	# files	%
*	34.6M	100%
std::pmr	1.7k	0.0049%

P3160R0 LEWG LITIGATION
Polls

POLL: We should promise more committee time to pursuing "An Allocator-aware inplace_vector," knowing that our time is scarce and this will leave less time for other work.

SF	F	N	A	SA
6	6	4	5	6

- Outcome: No consensus.
  - The room was not supportive of applying this paper to inplace_vector data structure.

TOKYO PLENARY

N4980 WG21 March 2024 Hybrid meeting Minutes of Meeting

Two objections to P0843R11

- One technical (constexpr)
  - Should be resolved by next plenary
  - (Most) P0843 authors agreed this needs to be resolved
- One on allocator support
  - LEWG chair informed WG21 it was already discussed
  - P0843 authors disagreed with waiting a meeting just to re-litigate this
  - On their own initiative, P3160 authors scheduled it for the next SG14 telecon

ST. LOUIS

ST. LOUIS LEWG

- The results of the SG14 meeting were not presented.

ST. LOUIS PLENARY

N4985 WG21 June 2024 Hybrid meeting Minutes of Meeting

- This paper was delayed in Tokyo. Have all the concerns now been addressed?
  - Yes. The paper was changed to limit the types that are constexpr which makes it possible to implement.
  - And there is also implementation experience now.
- Objections in the room.

	In favour	Opposed	Abstain
In Person	54	1	13
Online	16	0	16
Total	70	1	29

- Motion passes.
  - Only ***one*** WG21 member opposed this design

“UNLESS SOMETHING CHANGES WE SHOULD NOT RE-LITIGATE PREVIOUS DISCUSSIONS”
- LEWG CHAIR

P3160R2 - WHAT REALLY CHANGED?

P3160R2
Polls

POLL: If the embeded issues can be solved, would you be OK with adding the allocator template parameter to inplace vector?

SF	F	N	A	SA
6	8	1	3	0

- Being “OK with adding the allocator template parameter” is a far cry from being asked to re-litigate a strong plenary-approved WG21 design!
  - We already knew that some folks were “OK with adding the allocator template parameter”.
  - This is not new information!

P3160R2 SG14 RESULTS

- “A number of attendees voiced support for being able to have fine-grained memory control for allocator-aware objects stored within an inplace_vector whether or not they would use such a facility.”
  - How many?
  - If they aren’t using this C++11 facility how are they the intended audience?

- “Most of the attendees did not object to adding allocator support to inplace_vector”
  - That doesn’t mean it is their preferred design.
  - That doesn’t mean it is the right design.

NO, INPLACE_VECTOR SHOULDN'T HAVE AN ALLOCATOR

- We should not re-litigate this.
- There is no new information here.
- Make it a separate type, such as basic_inplace_vector<T, N, A>.
- Or maybe ~~clump~~ small_vector, while different (models an indefinite number of elements as opposed to inplace_vector, which models up to N elements), meets this need, since it is already in the world of allocation.
    - Am willing to work with the authors of P3160 to put this forward for C++29.

- This research used resources from the Argonne Leadership Computing Facility, a U.S. DOE Office of Science user facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. DOE under Contract No. DE-AC02-06CH11357.


Argonne
NATIONAL LABORATORY
Argonne Leadership
Computing Facility
U.S. DEPARTMENT OF
ENERGY