Library in a Week: Algorithms in C++11

Jeff Garland

C++Now – May 2012

Alternate Title:

Why C++11 is the awesomest language to Write & Write Algorithms!

Nod to Chris Kohlhoff (boost.asio)

What is this session?

- Goals
  - Build Boost library extensions
  - Enlarge the developer community
  - Opportunity to work collaboratively on useful capabilities
  - Learn by doing
  - Learn more about Boost tools and development standards
  - Learn more about C++11 features

Ways to participate

- Please participate!
  - Session(s) are meant to be interactive
  - Shaped/run by the participants
- Things you can do
  - Research, Write, Present an Algorithm
  - Write documentation
  - Write tests
  - Come to morning sessions and provide input

STL Algorithms Review

- Non-Mutating
  - for_each
  - find
  - find_if
  - adjacent_find
  - find_first_of
  - count
  - count_if
  - mismatch
  - equal
  - search
  - search_n
  - find_end
- Mutating
  - copy
  - copy_n
  - copy_backward
  - Swap
    - swap
    - iter_swap
    - swap_ranges
  - transform
  - Replace
    - replace
    - replace_if
    - replace_copy
    - replace_copy_if
  - fill
  - fill_n
  - generate
  - generate_n
  - Remove
    - remove
    - remove_if
    - remove_copy
    - remove_copy_if
  - unique
  - unique_copy
  - reverse
  - reverse_copy
  - rotate
  - rotate_copy
  - random_shuffle
  - random_sample
  - random_sample_n
  - partition
  - stable_partition

STL Algorithms Review

- Sort
  - sort
  - stable_sort
  - partial_sort
  - partial_sort_copy
- nth_element
- Binary search
  - lower_bound
  - upper_bound
  - equal_range
  - binary_search
- merge
- inplace_merge
- Set operations on sorted ranges
  - includes
  - set_union
  - set_intersection
  - set_difference
  - set_symmetric_difference
- Heap operations
  - push_heap
  - pop_heap
  - make_heap
  - sort_heap
  - is_heap
- Minimum and maximum
  - min
  - max
  - min_element
  - max_element
  - ETC!

C++ 11 – New Algorithms

- Sort
  - is_sorted
  - is_sorted_until
  - partial_sort_copy
- Mutating
  - copy_if
  - copy_n
  - copy_backward
  - Move
    - move
    - move_n
- Heap operations
  - is_heap_until
  - Is_heap
- Minimum and maximum
  - minmax
  - max
  - min_element
  - max_element

  - ETC!

C++11 Algorithm Usage

- Check collection for element

```
#include <algorithm>
#include <string>

bool is_green( const std::string& s )
{
  return ( s == "Green“ );
}

int main()
{

    std::vector<std::string> vs = {"Red", "Green", "Blue", "Orange"};

    if ( std::all_of(vs.begin(), vs.end(), is_green) )  {
      //….
```

C++11 Lambdas

```
#include <algorithm>
#include <string>

int main()
{

  std::vector<std::string> vs = {"Red", "Green", "Blue", "Orange"};

  if ( std::all_of ( vs.begin(), vs.end(),
                        [] (const std::string& s)  {return s == "Green";} ) )  {
     //….
```

C++11 Challenge: what’s this?

```
int min_val = std::min(  { 1, 2, 3, 4, 0 } );

 // min_val == 0  -- of course…


        T min ( initializer_list<T> t )
```

The Problem

- STL and C++11 Algorithms only scratch the surface
- Ideally Boost would contain large collection of additional useful algorithms
- One purpose of workshop is to build up the algorithm collection in Boost.Algorithm

What is Boost.Algorithm?

- An official Boost Library – Marshall Clow
  - To be added in Boost 1.50
  - Provides C++11 and other algorithms
  - Marshall wants your contributions!
- Other related libraries
  - String Algo – string algorithms
  - Range – variety of range based stl algorithms

Extending Algorithms - Range

```
//Range variant
template<class Collection, typename Function>
bool all_of(const Collection& c, Function f)
{
  return std::all_of(c.begin(), c.end(), f);
}
```

```
if (all_of( vs,    [](const std::string& s) {return s == "Green";}))
{
  //...
```

Ranges

- Boost.Range provides for simplification of writing for collection
- Volunteer to give mini-tutorial (10 minutes) tomorrow?

Approximate plan

- day 1: Get Organized
  - Selection of focus algorithms
  - Assignments and teaming
- day 2: Selected Presentations
  - Algorithm sketches / Illustrations of C++11 Code
- day 3: Code and docs
  - Algorithm Presentations and Review
- day 4: Coding and Any Redesign
  - Algorithm Presentations and Review
- day 5: Wrap up – future directions

Algorithm Technical Brainstorm

- What algorithms would you like to see?
  - Howard’s list (combinations, permutations)
  - Random sample (not really in 2003, in sgi)
  - Adobe libraries
  - Tuple algorithms (see boost.fusion)
  - Levenstein distance…
  - Depth_first_search
  - Boost::process – side session…

Algorithm Technical Brainstorm

- How can C++11 features be used to write algorithms?
- Challenge: extend a std or boost algorithm using C++11

For Tuesday

- Sebastion: Overview of Boost.Range
- Sean Parent: Overview of Adobe ASL
- Others…

Resources

- Boost.Algorithm
  - https://github.com/mclow/Boost.Algorithm
- Boost Range
  - http://www.boost.org/doc/libs/1_49_0/libs/range/doc/html/index.html
- Adobe Source Libraries (ASL)
  - Open source C++ library
  - Uses elements of Boost
  - Provides some additional algorithms:
  - http://stlab.adobe.com/group__algorithm.html
- SGI STL
  - Provides some additional algorithms beyond STL
  - http://www.sgi.com/tech/stl/table_of_contents.html