

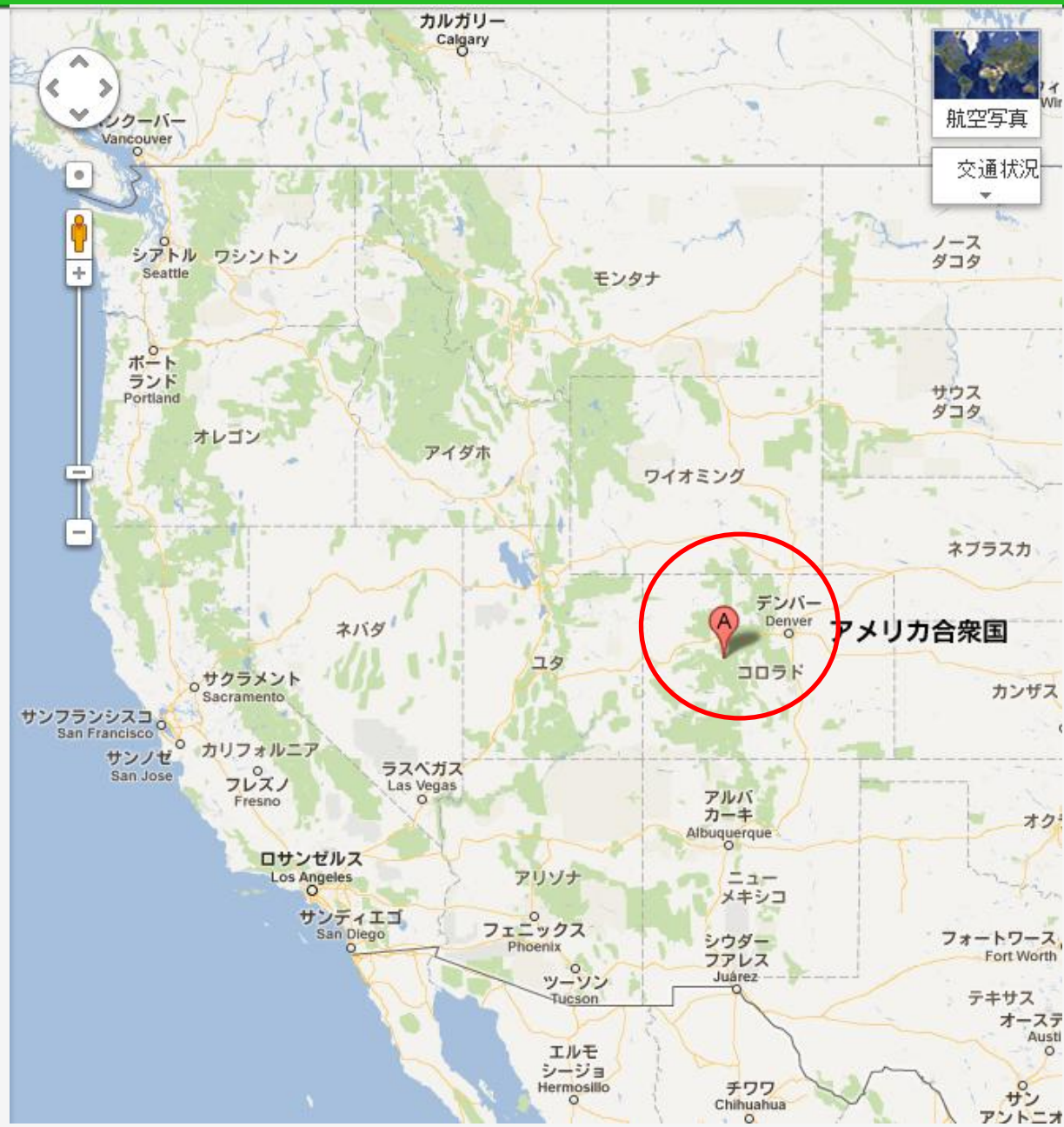
C++Now! 2012 参加レポート

高橋 晶 (Akira Takahashi)
[id:faith_and_brave](#)
[@cpp_akira](#)

- 2011年までBoostConという名前で開催されていたC++のカンファレンス
- 2012年からC++Now!という名前に変わった

- 毎年5月中旬に1週間行われる
 - 2012年は5/13(日)～5/18(金)
- 場所はコロラド州のアспен
 - Aspen Center for Physicsという、各建物に物理学者の名前が付いてる施設を4会場使って行われる

期間と場所



成田



サンフランシスコ



デンバー

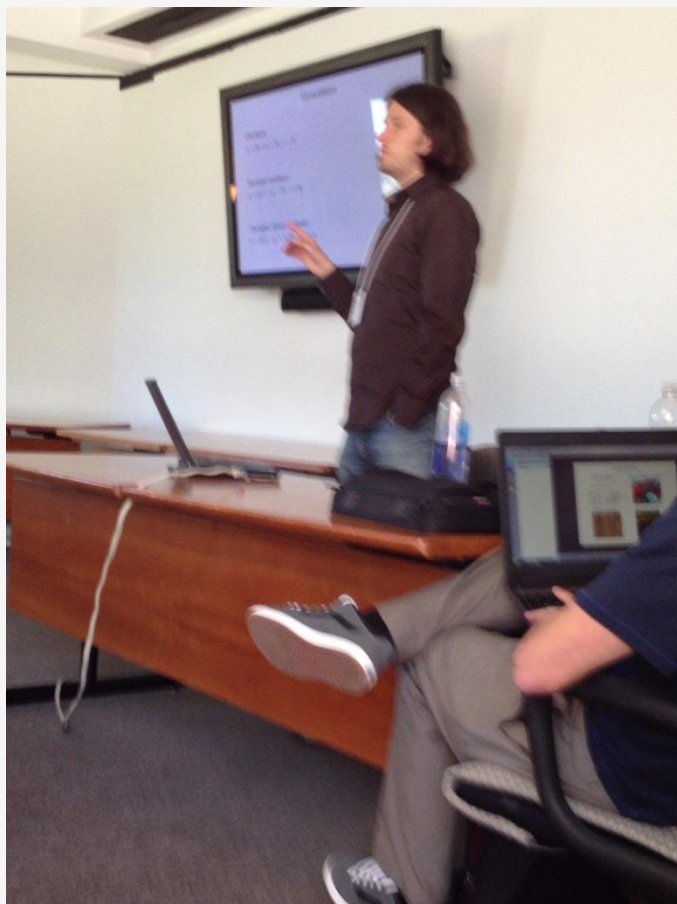


アスペン

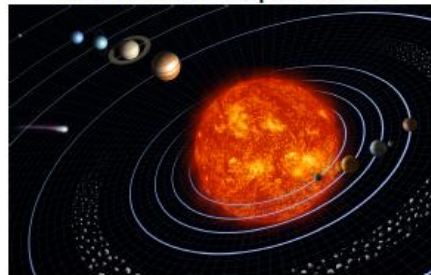
- Library in a Week
- More Useful Computations in the Same Duration: Optimizing Embedded Hard Real-Time Code in C++
- odeint – Solving ordinary differential equations in C++
- Lambda Functions
- Ustring – A Modern Alternative to `std::string`
- Grill the Committee

- 1週間みんなでライブラリを作ろう！というワークショップ
- 毎朝1時間みんなで集まって途中経過やアイデアの発表を行う
- 毎朝1時間作業するのではなく、休み時間などの合間でライブラリを作る
 - 時間ない
- “Replace Output Iterator and Extended Range”というタイトルで発表してきました
 - 「Boost.勉強会 #9 つくば」にて、日本語でも発表しました

- 常微分方程式を解くためのライブラリ
 - 現在Boostにレビューリクエスト中



Newtons equations

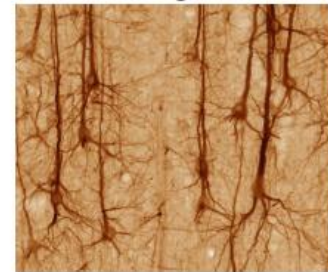


Granular systems

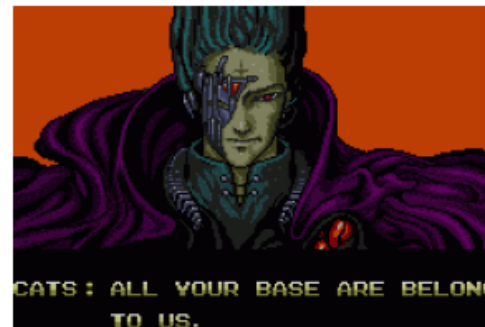
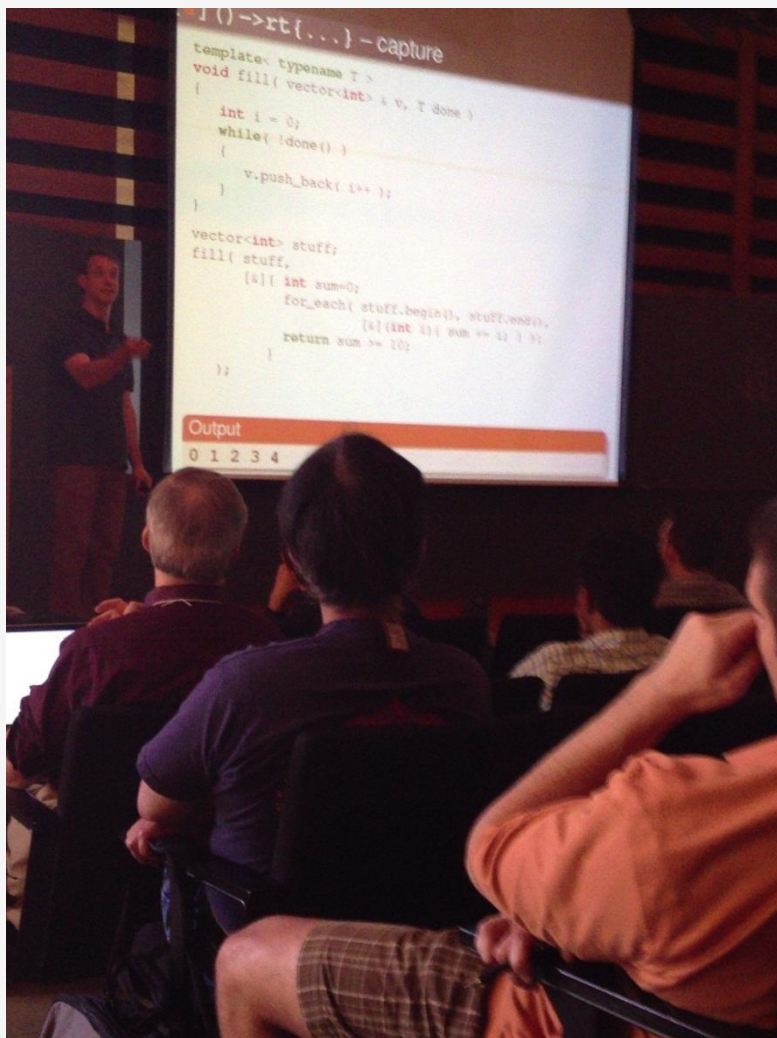


Reaction and relaxation equations (i.e. blood alcohol content, chemical reaction rates)

Interacting neurons



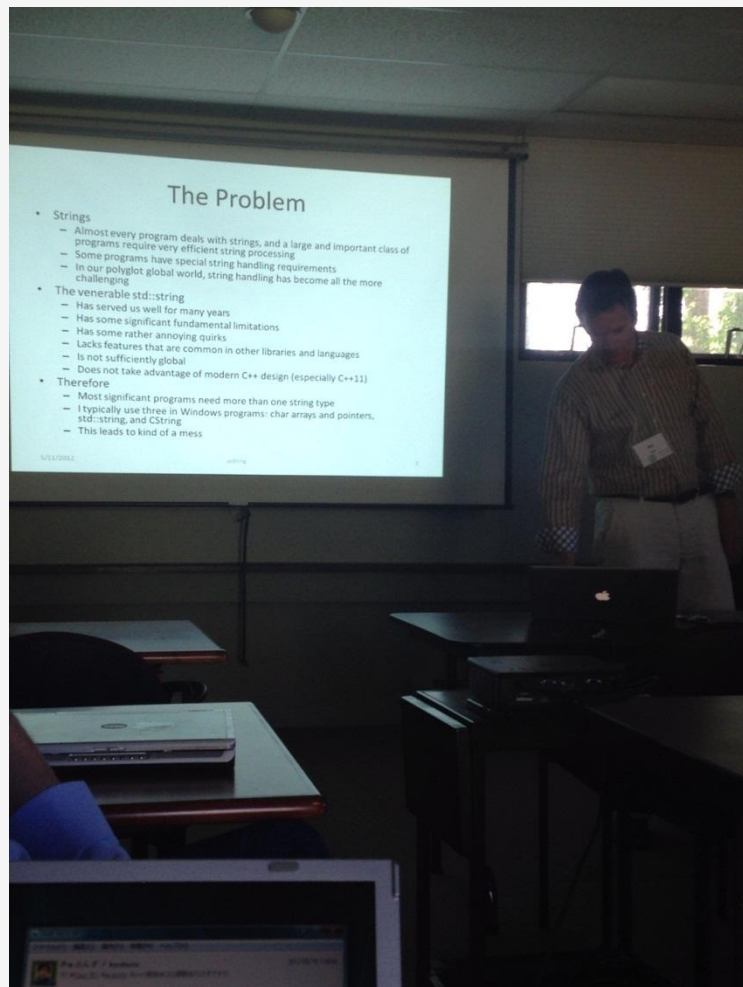
- C++11で導入されたラムダ式の紹介



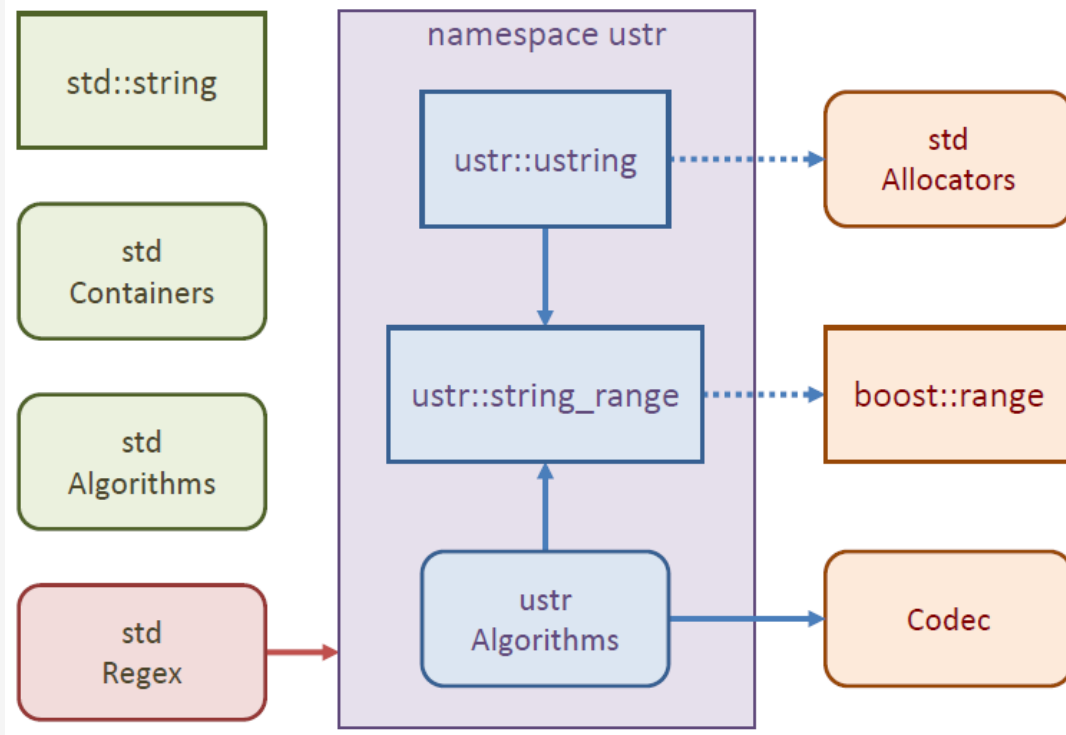
```
for_each( your_base.begin(), your_base.end(),  
          [&my_base](base_t & b)  
          { my_base.push_back( move(b) ); } );
```

```
// all your base are belong to us
```

- Unicode対応文字列をみんなで考えよう、という発表 & ディスカッション



The Ustring Library



Grill the Committee

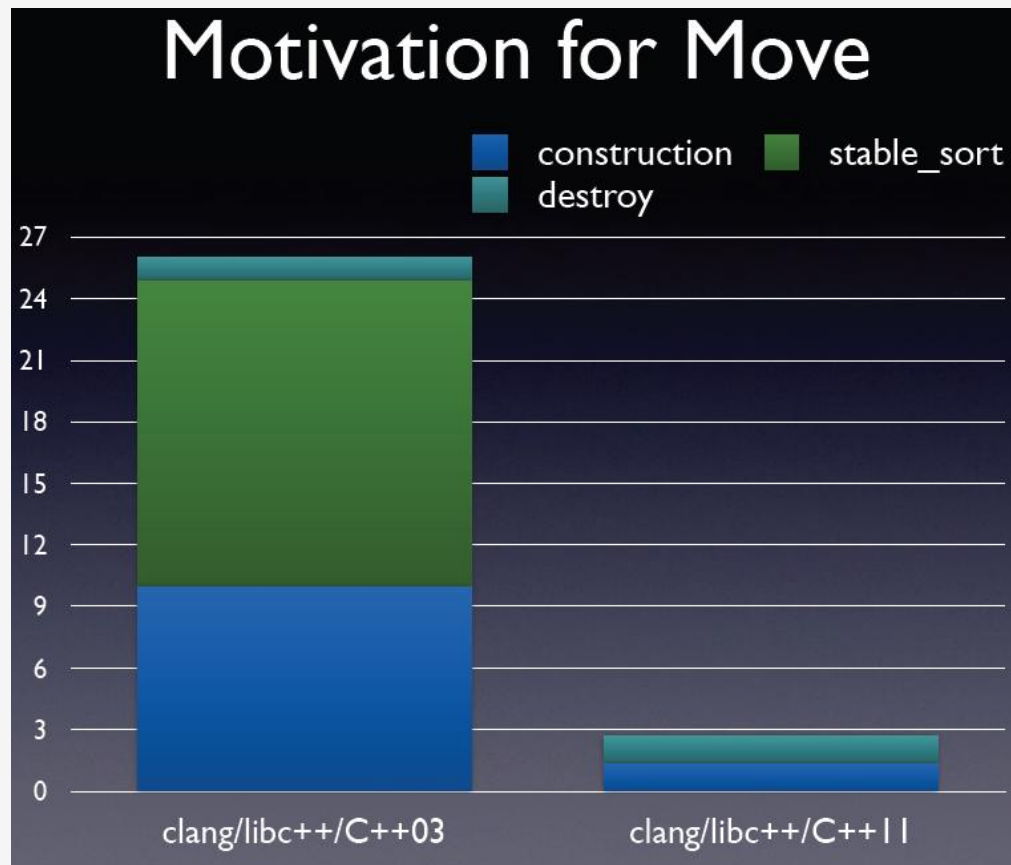
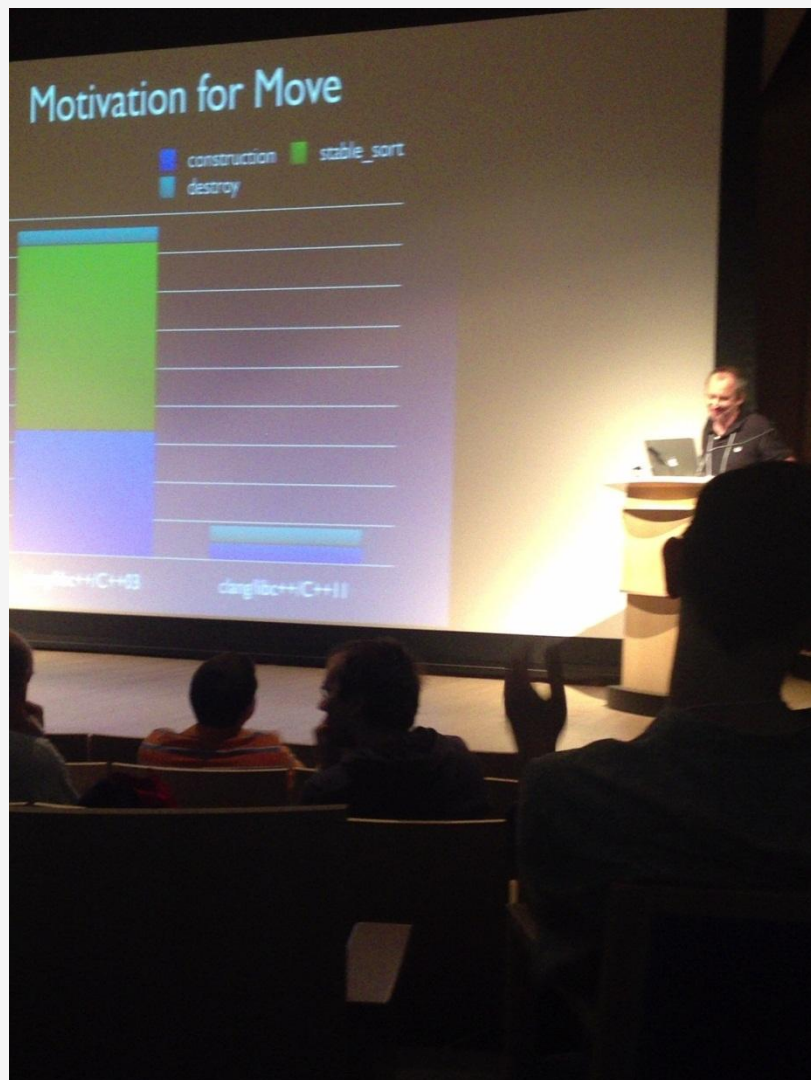
- 標準化委員を問い詰めよう！ ディスカッション



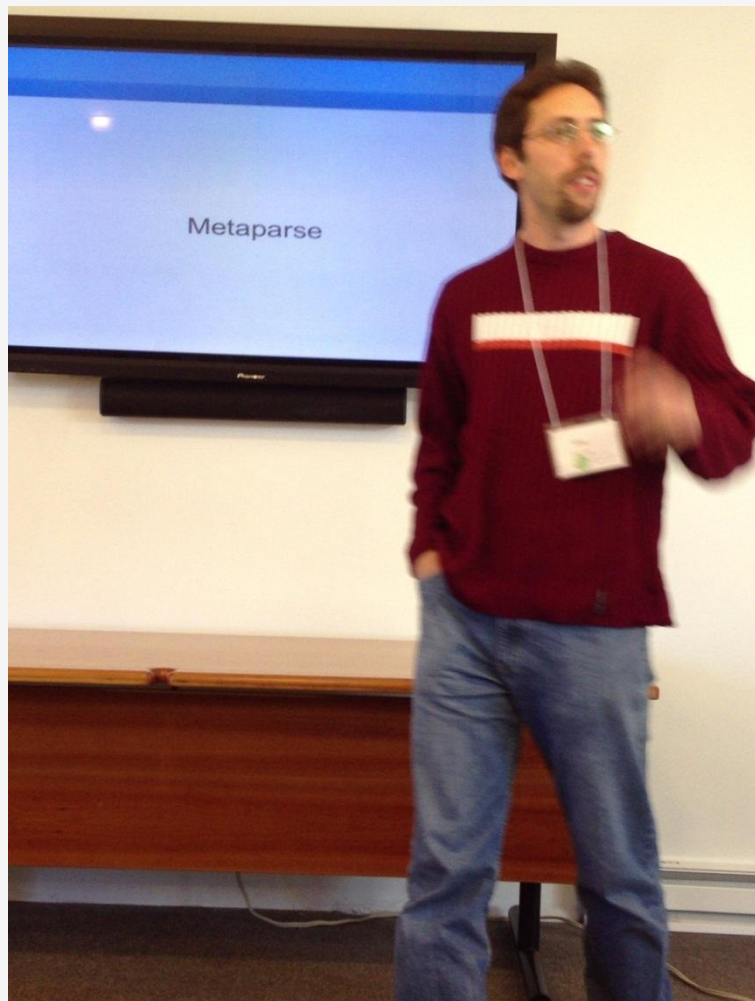
- Moving Forward with C++11
- Metaparse
- Exception-Safe Coding in C++
- Utilizing Modern Programming Techniques for Scientific Software Development
- Pincnic

Moving Forward with C++11

- C++11から導入されたムーブセマンティクスに関するキーノート



- コンパイル時構文解析ライブラリ
 - 文字列リテラルとしてHaskellコードを書いたらC++メタ関数として実行されたり



Building the DSL

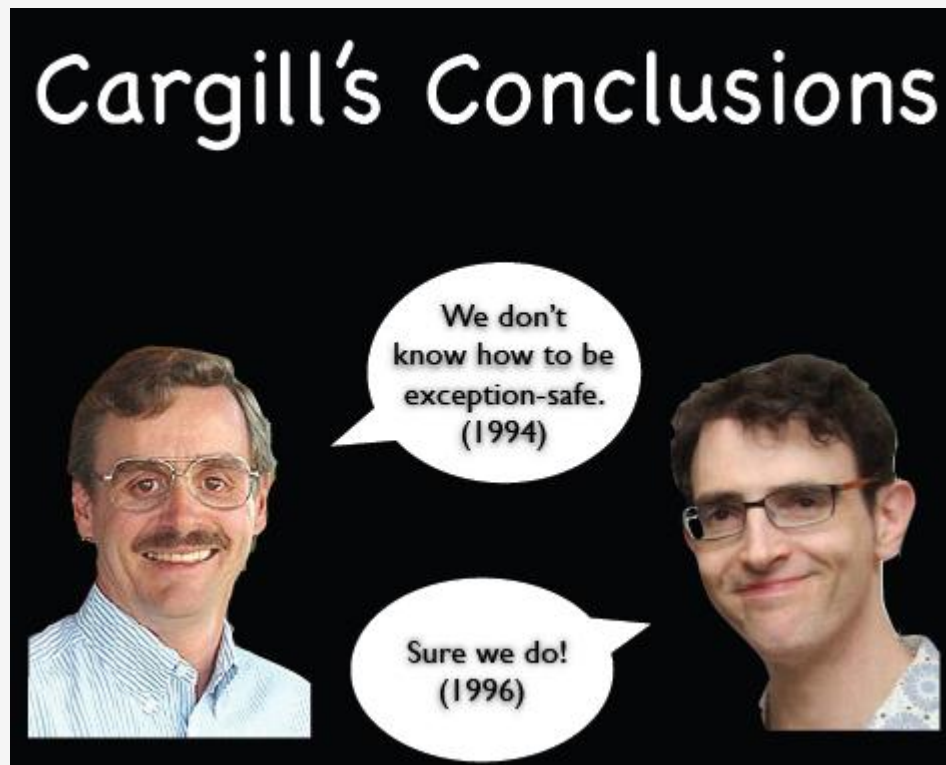
```
typedef
    meta_hs
        ::define<
            _S("fact n = if n == 0 then 1 else n * fact (n-1)")
        >::type

        ::get<_S("fact")>::type
    fact;
```

```
mpl::apply<fact, mpl::int_<3>>::type
```

```
mpl::transform<
    mpl::vector_c<int, 1, 2, 3, 4, 5>,
    fact
>::type
```

- 毎年恒例(?) 例外安全性に関する発表



Tom Cargill : 「例外安全性を持つ汎用コンテナを書くのは不可能だ(1994)」
Dave Abrahams : 「**もちろんできるさ！(1996)**」

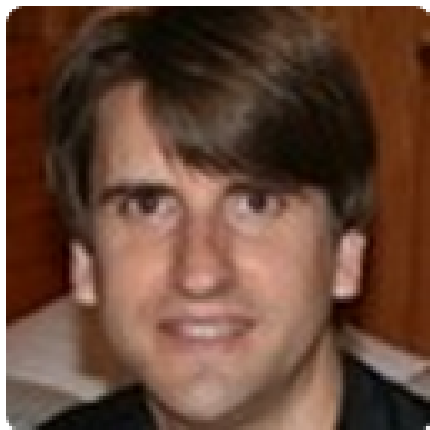
- みんなでひたすら肉を食う



日本語を話せる組↓



- 日本語ペラペラ
- 「C++Now!」という名前を付けたひと。運営メンバ
- Twitter @rayfix ↓



Ray Fix (実際外人)

@rayfix あなたをフォロー中。

Software engineer, environmentalist, hack guitarist and Ultimate player. I also help run a site called Skqueak!
Huntington Beach, CA - <https://www.skqueak.com>

- Modules in C++
- Generic Programming in C++
- ConceptClang
- Boost.Polygon.Voronoi

- C++の次期標準で考えられているモジュールシステムに関するキーノート

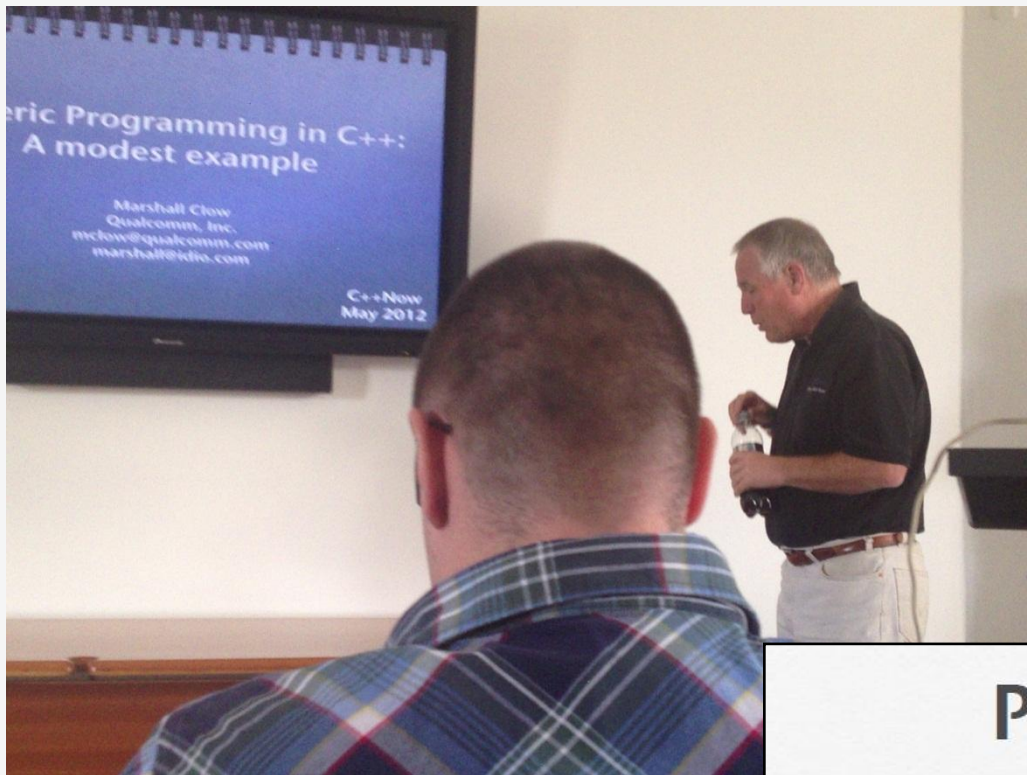


```
export MyLib:
public:
    struct S {
        void f(long x) { this->f((long long)x); }
    private:
        void f(long long x);
    };
    void S::f(long long x) { ... }
```

```
import MyLib;

void g(S x) {
    x.f(42); // Unambiguous.
}
```

- ジェネリックプログラミングの現実の問題を考えてみよう、という発表 & ディスカッション



Problem Definition

On November 1st, on the Boost developers list, Olaf asked:

- > **Does Boost have hex/unhex() that support std::string?**
- > **I can't find them and I think they'd be quite handy to have.**

- 現在Clangで実験的な実装が進められているConceptについて

ConceptClang:
Towards A Compilation Model for C++
Concepts

Andrew Lumsdaine, Larisse Voufo
Center for Research in Extreme Scale Technologies (CREST)
Pervasive Technology Institute at Indiana University
C++Now 2012

Constrained Templates

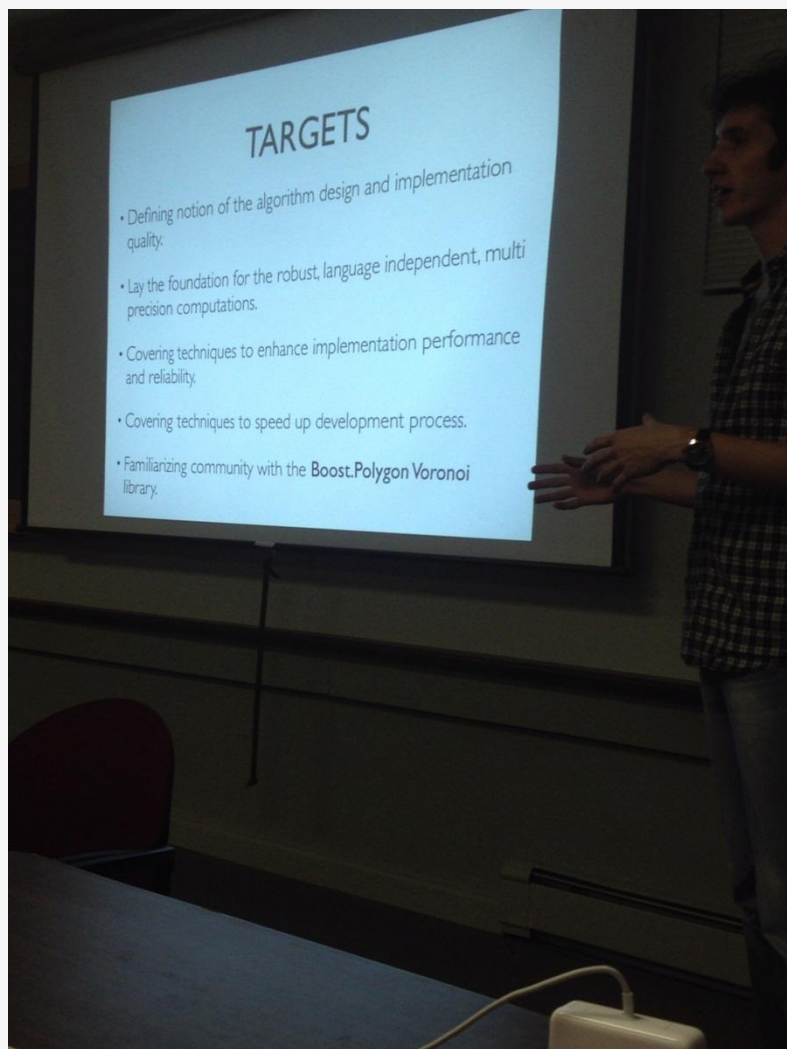
- Extend template definitions:
 - with a **requires** clause: **== Constraints specification.**

```
template<typename I, typename T, typename Op>  
requires(InputIterator<I>, BinaryFunction<Op>,  
         Assignable<InputIterator<I>::value_type,  
                   BinaryFunction<Op>::result_type>)  
T accumulate(I first, I last, T init, Op bin_op);
```

- Using a simplified form:
 - For more complex constraints: use a **requires** clause

```
template<InputIterator I, typename T, BinaryFunction Op>  
requires(Assignable<InputIterator<I>::value_type,  
         BinaryFunction<Op>::result_type>)  
T accumulate(I first, I last, T init, Op bin_op);
```


- Boost.Polygonに提案中のボロノイ図に関する発表



TARGETS

- Defining notion of the algorithm design and implementation quality.
- Lay the foundation for the robust, language independent, multi precision computations.
- Covering techniques to enhance implementation performance and reliability.
- Covering techniques to speed up development process.
- Familiarizing community with the **Boost.Polygon Voronoi** library.

VORONOI DIAGRAM

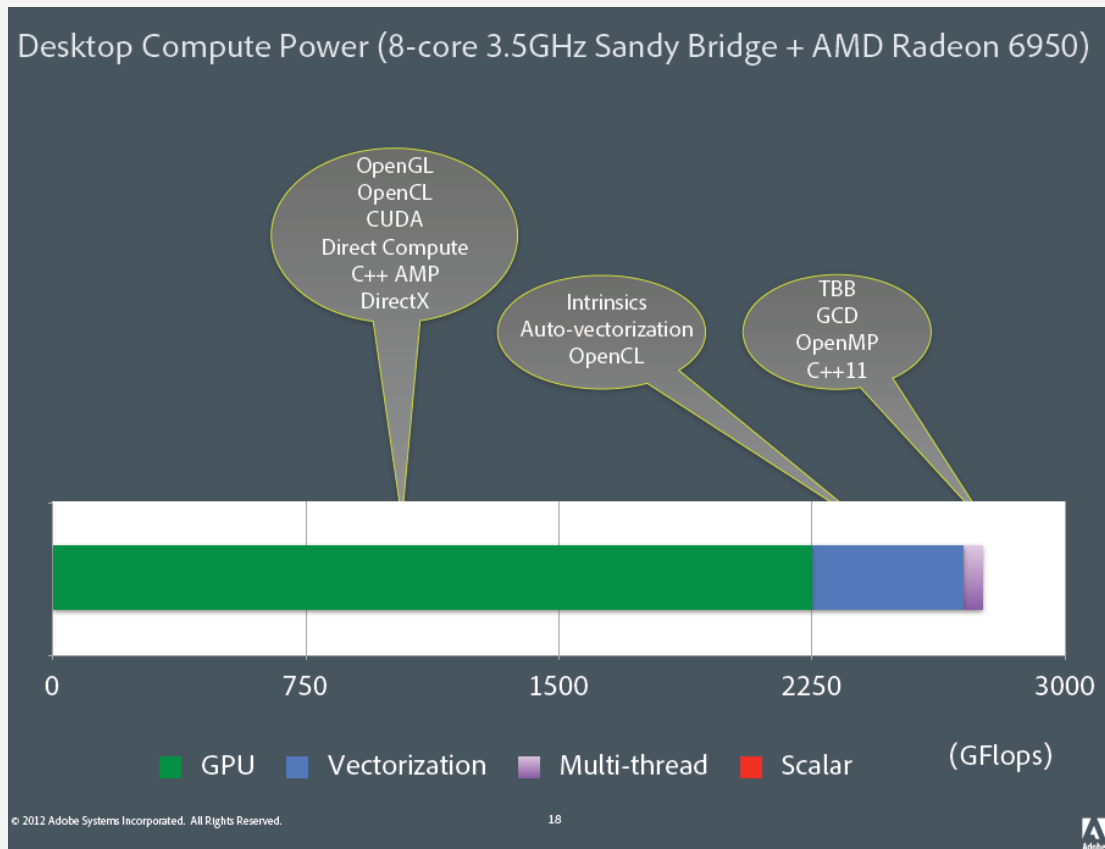
Computational geometry concept named in honor of the Ukrainian mathematician Georgi Voronoi.

Represents partition of the space onto regions, with bounds determined by the distances to a specified family of objects.



- Now What?
- Type to string conversion
- Metaprogramming Applied to Numerical Problems
- Linear programming made easy with Boost.Proto
- Using Boost.Coroutine to untangle a state machine
- Boosting Libraries for TR2
- Preparation of C++Now! 2013

- C++が今どこにいるのか？というキーノート
 - C++が得意とする分野(GPU関係)が標準にないよね、という話があったり



- Boost.ContextベースでBoost.Coroutineを書き直したという話
 - コルーチンの解説(状態マシン、非同期処理など)

Context Switch looks like Function Call

<http://hg.secondlife.com/viewer-release/src/3142ebd0bc5a/indra/lleventcoro.h#cl-204>

```
template <typename SELF>
LLSD postAndWait(SELF& self, const LLSD& event,
                 const LLEventPumpOrPumpName& requestPump,
                 const LLEventPumpOrPumpName& replyPump, ...)
{
    // declare the future
    boost::coroutines::future<LLSD> future(self);
    // make a callback that will assign a value to the future, and listen on
    // the specified LLEventPump with that callback
    LLTempBoundListener connection(replyPump.getPump().listen(listenerName,
                                                                voidlistener(boost::coroutines::make_callback(future))));
    requestPump.getPump().post(modevent);
    // trying to dereference ("resolve") the future makes us wait for it
    LLSD value(*future);
    // returning should disconnect the connection
    return value;
}
```

Context switch



- 標準への提案を成功させるには。ボランティアの募集。Boostからの提案など。



The C++ standards committee wants YOU!

- 来年のC++Now!をどうするか



- C++Now!の資料は、GitHubで公開されています
 - https://github.com/boostcon/cppnow_presentations_2012
- 日本語ページ
 - <https://sites.google.com/site/boostjp/cppnow/2012>
- C++Now!参加レポートのブログ
 - redboltz <http://d.hatena.ne.jp/redboltz/20120519/1337385881>
 - 高橋晶 http://d.hatena.ne.jp/faith_and_brave/20120522/1337650107
 - Boris Schalingの記事(ドイツ語からの日本語訳)
<http://zakkas783.tumblr.com/post/25007975614/c-now-boostcon-c-now-boost-konferenz-im>

Q & A