Lessons from C++

John V. W. Reynders
Los Alamos National Laboratory

SuperComputing ‘98

November 13, 1998

Scientific SuperComputing

- Rites of Passage
- The C++/Fortran Battle
- High Performance
- Language Design/Features

Rites of Passage

- Cost of rewrite to a physicist is too high
  - Demonstrated C++ codes which outperformed their Fortran equivalents
    - Able to deploy a wider range of object optimizations such as array compression
    - parallel encapsulation enabled “hidden” multi-threading
  - Demonstrated C++ codes which outscaled their Fortran equivalents
    - Dynamic load balancing (e.g. particle swap and balance)
- Developed full-scale physics codes:
  - multi-material hydro
  - fusion plasmas
  - advanced accelerator design

The C++/Fortran Battle:

- Performance
  - a loop is a loop by any other name
  - pointer aliasing
  - language complexity
  - paradigms (particles vs. species)
- Legacy Base
  - extending
  - heterogeneous source
  - tools
- Expertise
- Market Presence (F90 vs. C++ : ride the wave!!!)

High Performance

- C++
  - standard compliance
  - Mature Compilers
    - Aggressive In-lining
    - Small object opt.
  - Restrict Keyword
  - Template programming
- Java
  - Compile as another language
  - intrinsic concurrency
    - threads (how light?)
    - RMI

Mixed model - message passing/RMI/threading
Memory/thread affinity

Language Design/Features

The performance will get there:

The long pole in the tent will be the language features which limit how simulations developers can represent their physics.

- overloaded operators
- ***generic programming with parameterized classes***
  - ***generic libraries (STL)***
  - ***expression templates***
  - ***compile-time polymorphism***
  - ***meta-programming***