C++ at BES

Guang-Pei Chen

May 13, 1999

- Introduction
- C++ under DRUNK
- Testing Results
- Conslusion

Introduction

- Status
  - o C++ is popular in HEP community today
  - o No C++ @ BES
    V101 = 1235 (.f) + 402 (.inc) + 19 (.c)
- Difficulties
  - o Rewrite all BESLIB in C++?
  - o Lack of C++ experts
  - o No field for BESers to use C++
- Insert C++ in DRUNK?
  - o An hybrid
    $\Longrightarrow$ Mix FORTRAN, C, and C++
  - o Keep reconstruction in FORTRAN, but extract `evlst/trklst` in C++
  - o Feasible from view of Computer Language

- Need to
    - Find an approriate C++ compiler
    - Select linker to mix FORTRAN, C, and C++
    - Transfer `evlst/trklst` to C++
    - Construct besEvent class
    - Create interface between FORTRAN and C++

C++ compiler

Work based on HP-UX @IHEP

- FORTRAN compiler: `fort77`
- C compiler: `cc` (`gcc` can be used too)
- C++ compiler: `g++` (`CC` not standard)
- Create own `bcomp.loc` and modify `bcomp`

```
#*******************************************************************************
#  File:     bcomp.loc
#  $Id: bcomp.loc,v 1.2 1999/05/12 02:37:29 chengp Exp $
#  $Author: chengp $
#  Purpose: Personal setup of C++ compiler for bcomp
#  $Date: 1999/05/12 02:37:29 $
#*******************************************************************************
# C++ compiler:
   set CC = "g++ -Dextname"
# C++ extension:
   set CCext = "cc"

#!/u/u01/huxq/bin/csh.9.05 -f
#*******************************************************************************
# BCOMP 1.50
...
foreach ext ($fext c $CCext)
...
if ($src:e == cc    ) set aa = "$CC"      #--C+++
...
```

C++ library

- Common linker: `fort77`
  Much to be modified if g77 used

- Create own `sob_drk.loc` to include C++ library

- Submit DRUNK job as usual
  `drunk -v101 -d test`

```
#****************************************************************************
#   File:     sob_drk.loc
#   $Id: sob_drk.loc,v 1.2 1999/05/12 02:36:34 chengp Exp $
#   $Author: chengp $
#   Purpose: Personal setup of C++ lib for sob_drk
#   $Date: 1999/05/12 02:36:34 $
#****************************************************************************
#--Libraries
   set liblist = "$liblist -L/data/d06/chengp/lib -lstdc++"
   set liblist = "$liblist
       -L/data/d06/chengp/lib/gcc-lib/hppa1.1-hp-hpux10.20/2.8.1 -lgcc"
```

evlst/trklst ⟹ C++

```
// =======================================================================
// File:    evlst.h
// $Id: evlst.h,v 1.2 1999/05/12 02:33:08 chengp Exp $
// Purpose: Transfer evlst data to C++ objects
// $Author: chengp $
// $Date: 1999/05/12 02:33:08 $
// =======================================================================
#ifndef _EVLST_H
#define _EVLST_H

extern "C" struct {
  int evlst[3];
  float ecm;
  int irun, irec, ecode, subcod, evid, ercode;
  int nvtx, ivtx, nuvctk, nchgvc,tottks, nchrg, nneu, nchrgx, nvirt;
  float evlstx[2];
  int npipe, npcor, scvrsn, npizro, nvee;
  int ndctot, ndcunu, nsctot, ntof;
  int vrtsub[15][8], trksub[80][18];
  float evlstz[24];
}evlst_;
#endif

// =======================================================================
// File:    trklst.h
// $Id: trklst.h,v 1.2 1999/05/12 02:33:15 chengp Exp $
// Purpose: Transfer trklst data to C++ objects
// $Author: chengp $
// $Date: 1999/05/12 02:33:15 $
// =======================================================================
#ifndef _TRKLST_H
#define _TRKLST_H

extern "C" struct {
  int trklst[3];
  int nxtsub;
  float trk[10240];
}trklst_;
#endif
```

Class of besEvent

besEvent

```
//============================================================
// File:    besevent.hh
// $Id: besevent.hh,v 1.2 1999/05/12 02:33:29 chengp Exp $
// Purpose: Define BES Event class
// $Author: chengp $
// $Date: 1999/05/12 02:33:29 $
//============================================================
#ifndef _BESEVENT_HH
#define _BESEVENT_HH

// Basic header files

// Own header files
#include "evlst.h"
#include "bestrack.hh"

class besEvent{

public:
  // Constructor
  besEvent(){
    _irun = evlst_.irun;
    _irec = evlst_.irec;
    _ecm = evlst_.ecm;
    _nchrg = evlst_.nchrg;
    _nneu = evlst_.nneu;
    _tottks = evlst_.tottks;

    _track = new besTrack[_tottks];
    for (int i=0; i<_tottks; i++){
      besTrack track(i);
      _track[i] = track;
    }
  };
```

```
  // Destructor
  ~besEvent(){
    delete [] _track;
  };

  // Member functions
  int irun() const;
  int irec() const;
  float ecm() const;
  int nchrg() const;
  int nneu() const;
  int tottks() const;
  besTrack* track(const int &) const;

  bool Bhabha() const;

private:
  int _irun, _irec;
  float _ecm;
  int _nchrg, _nneu, _tottks;
  besTrack* _track;
};

\\ Blah blah ...
inline float besEvent::ecm() const{
  return _ecm;
}

\\ Blah blah ...
inline besTrack* besEvent::track(const int &i) const{
  return &_track[i];
}

bool besEvent::Bhabha() const{
  // DC
  if (_nchrg != 2) return false;

  int sumq = 0;
  float theta[2], phi[2];
  for (int i=0; i<_nchrg; i++) {
    if (_track[i].dc().MFIT() != 2 &&
```

```
_track[i].dc().MFIT() != -19) return false;

    float pe = _track[i].dc().p() / _track[i].sc().E();
    if (pe < 0.87 || pe > 1.13) return false;

    sumq += _track[i].dc().charge();
    theta[i] = _track[i].dc().theta();
    phi[i] = _track[i].dc().phi();
  }

  if (sumq != 0) return false;

  float costheta = cos(theta[0])*cos(theta[1]) + sin(theta[0])*sin(theta[1])*
    cos(phi[0]-phi[1]);
  if (costheta >= -0.998) return false;

  // SC
  float e[2];
  for (int i=0; i<_nchrg; i++) {
    if (_track[i].sc().mod() != 1) return false;

    if (_track[i].sc().phit() <= 15) return false;

    theta[i] = _track[i].sc().theta();
    phi[i] = _track[i].sc().phi();
    costheta = cos(theta[i]);
    if (costheta>0.76 || costheta<-0.76) return false;

    e[i] = _track[i].sc().E();
  }

  costheta = cos(theta[0])*cos(theta[1]) + sin(theta[0])*sin(theta[1])*
    cos(phi[0]-phi[1]);
  if (costheta >= -0.970) return false;

  float invmass = sqrt(e[0]*e[1]*(1-costheta));
  if (invmass <= 0.7) return false;

  return true;

}
#endif
```

besTrack

```
//=============================================================
// File:    bestrack.hh
// $Id: bestrack.hh,v 1.2 1999/05/12 02:34:36 chengp Exp $
// Purpose: Define BES Track class
// $Author: chengp $
// $Date: 1999/05/12 02:34:36 $
//=============================================================
#ifndef _BESTRACK_HH
#define _BESTRACK_HH

// Basic header files

// Own header files
#include "besdctrack.hh"
#include "bestoftrack.hh"
#include "bessctrack.hh"

class besTrack{

public:
  // Constructor
  besTrack(){
    _trkNumber = 0;
  };

  besTrack(const int &trkNumber){
    _trkNumber = trkNumber;
    _dc = besDCTrack(_trkNumber);
    _tof = besTOFTrack(_trkNumber);
    _sc = besSCTrack(_trkNumber);
  };

  besTrack(const besTrack &track){                // Copy constructer
    _trkNumber = track._trkNumber;
    _dc = besDCTrack(_trkNumber);
    _tof = besTOFTrack(_trkNumber);
    _sc = besSCTrack(_trkNumber);
  };
```

```
  // Destructor
  ~besTrack(){};

  // Member functions
  bool charged() const;
  bool neutral() const;
  besDCTrack dc() const;
  besTOFTrack tof() const;
  besSCTrack sc() const;

private:
  // Identification
  int _trkNumber;
  besDCTrack _dc;
  besTOFTrack _tof;
  besSCTrack _sc;
};

#endif
```

besSubTrack, besDCtrack, ...

```
//============================================================
// File:    bessubtrack.hh
// $Id: bessubtrack.hh,v 1.1 1999/05/12 02:34:20 chengp Exp $
// Purpose: Define BES sub-track virtual class
// $Author: chengp $
// $Date: 1999/05/12 02:34:20 $
//============================================================
#ifndef _BESSUBTRACK_HH
#define _BESSUBTRACK_HH

// Basic header files

// Own header files
#include "evlst.h"
#include "trklst.h"

class besSubTrack{
```

```
public:
  // Constructor
  besSubTrack(){
    _trkNumber = -1;
    _isub = -1;
  };

  besSubTrack(const int &trkNumber, const int &isub){
    _trkNumber = trkNumber;
    _isub = isub;
  };

  // Destructor
  ~besSubTrack(){};

  // Member functions
  int itrk(const int &) const;
  float trk(const int &) const;

protected:
  // Identification
  int _trkNumber;
  int _isub;
};

inline int besSubTrack::itrk(const int &i) const{
  return _isub>=0?trklst_.trklst[4+_isub+14]:999;
}

inline float besSubTrack::trk(const int &i) const{
  return _isub>=0?trklst_.trk[_isub+i]:999.0;
}

#endif

//=============================================================
// File:    besdctrk.hh
// $Id: besdctrack.hh,v 1.2 1999/05/12 02:33:23 chengp Exp $
// Purpose: Define BES Track class
// $Author: chengp $
// $Date: 1999/05/12 02:33:23 $
```

```
//============================================================
#ifndef _BESDCTRACK_HH
#define _BESDCTRACK_HH

// Basic header files

// Own header files
#include "evlst.h"
#include "trklst.h"
#include "bessubtrack.hh"

class besDCTrack: public besSubTrack{

public:
  // Constructor
  besDCTrack(){
    _trkNumber = -1;
    _isub = -1;
  };

  besDCTrack(const int &trkNumber){
    _trkNumber = trkNumber;
    _isub = evlst_.trksub[_trkNumber][2] - 1;
  };

  besDCTrack(const besDCTrack &track){                // Copy constructer
    _trkNumber = track._trkNumber;
    _isub = track._isub;
  };

  // Destructor
  ~besDCTrack(){};

  // Member functions:
  int MFIT() const;
  int charge() const;
  float p() const;
  float theta() const;
  float phi() const;
  float x0() const;
  float y0() const;
  float z0() const;
```

```
  float r0() const;
};

inline int besDCTrack::MFIT() const{
  return _isub>=0?itrk(14):999;
}

inline int besDCTrack::charge() const{
  return _isub>=0?itrk(2):999;
}

inline float besDCTrack::p() const{
  return _isub>=0?trk(7):999.0;
}

inline float besDCTrack::theta() const{
  return _isub>=0?trk(8):999.0;
}

inline float besDCTrack::phi() const{
  return _isub>=0?trk(9):999.0;
}

inline float besDCTrack::x0() const{
  return _isub>=0?trk(10):999.0;
}

inline float besDCTrack::y0() const{
  return _isub>=0?trk(11):999.0;
}

inline float besDCTrack::z0() const{
  return _isub>=0?trk(12):999.0;
}

inline float besDCTrack::r0() const{
  return _isub>=0?trk(13):999.0;
}

#endif
```

Interface between FORTRAN and C++

DRUNK card file

- Append C++ file (.cc) in /*LINK card
- Path of CERN cfortran utility

```
/* Blah blah ...
/*LINK test
/*LINK -I/cern/97a/include/cfortran
/* Blah blah ...
```

BESIOR/BESEVN/BESEOR in DRUNK

```
//==============================================================
// File:    test.cc
// $Id: test.cc,v 1.3 1999/05/12 07:20:28 chengp Exp $
// Purpose: Interface of C++ and FORTRAN under DRUNK
// $Author: chengp $
// $Date: 1999/05/12 07:20:28 $
//==============================================================
// Basic header files
#include <iostream.h>

// Own header files
#include "besver.h"
```

```
#include "besevent.hh"
#include "bespaw.hh"
#include "ntuple.hh"

extern "C" seterg_();
extern "C" recon_(int &, int &);
extern "C" scbhabha_();

extern "C" {

  extern void besior_(int *irun){
    besver_.recver = 101;
  }

  extern void besevn_(int *jrec){
    seterg_();

    int istatus, ierror;
    recon_(istatus, ierror);

    besEvent event;
    cout << "irun=" << event.irun() << " irec=" << event.irec()
 << " ecm=" << event.ecm() << " nchrg=" << event.nchrg() << endl;

    for (int i=0; i<event.nchrg(); i++){
      cout << " besTrack p=" << event.track(i)->dc().p()
   << " besTrack MFIT=" << event.track(i)->dc().MFIT() << endl;
    }
  }

  extern void beseor_(int *irun){}
}
```

```
irun=12501 irec=108 ecm=3.77 nchrg=2
 besTrack p=15.8836 besTrack MFIT=2
 besTrack p=7.67718 besTrack MFIT=2
irun=12501 irec=262 ecm=3.77 nchrg=2
 besTrack p=1.86899 besTrack MFIT=2
 besTrack p=2.74078 besTrack MFIT=2
irun=12501 irec=302 ecm=3.77 nchrg=2
 besTrack p=1.69702 besTrack MFIT=2
 besTrack p=1.542 besTrack MFIT=2
```

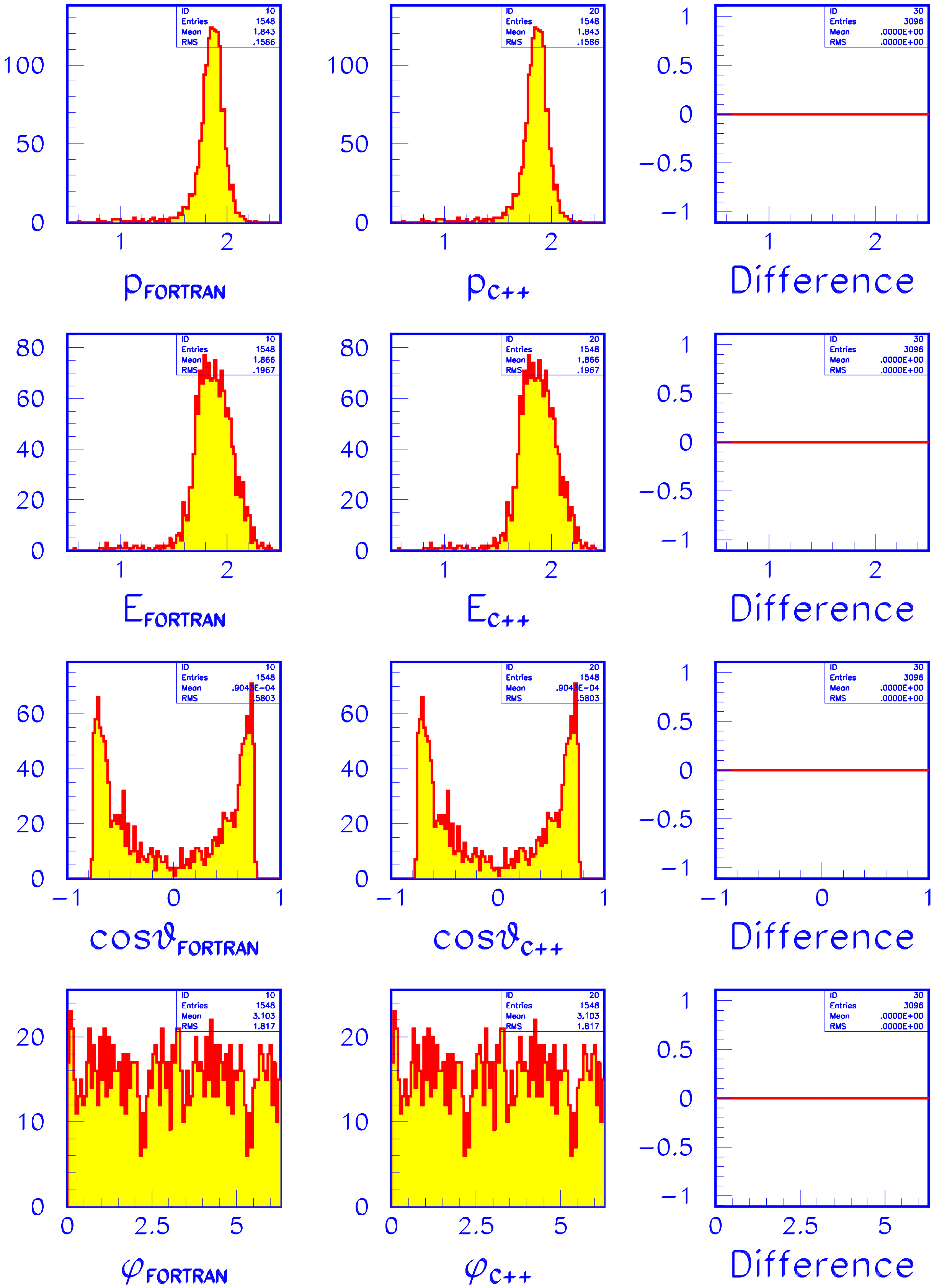
Testing Results

To choose Bhabha events for BSC calibration

- In DC:
  - NCHRG = 2
  - MFIT = 2/-19
  - $\mathbf{0.87 \leq p/E \leq 1.13}$
  - $\mathbf{\Sigma Q = 0}$
  - $\mathbf{\cos\theta_{1-2} < -0.998}$

- In BSC:
  - SCMOD = 1
  - PHIT $> \mathbf{15}$
  - $|\mathbf{\cos\theta}| \leq \mathbf{0.76}$
  - $\mathbf{\cos\theta_{1-2} < -0.970}$
  - Invariant mass $> \mathbf{0.7}$GeV

Ntuple in C++ is managed via CERN cfortran utility


ID 10
Entries 1548
Mean 1.843
RMS .1586
ID 20
Entries 1548
Mean 1.843
RMS .1586
ID 30
Entries 3096
Mean .0000E+00
RMS .0000E+00
p FORTRAN
p C++
Difference
ID 10
Entries 1548
Mean 1.866
RMS .1967
ID 20
Entries 1548
Mean 1.866
RMS .1967
ID 30
Entries 3096
Mean .0000E+00
RMS .0000E+00
E FORTRAN
E C++
Difference
ID 10
Entries 1548
Mean .9043E−04
RMS .5803
ID 20
Entries 1548
Mean .9043E−04
RMS .5803
ID 30
Entries 3096
Mean .0000E+00
RMS .0000E+00
cosϑ FORTRAN
cosϑ C++
Difference
ID 10
Entries 1548
Mean 3.103
RMS 1.817
ID 20
Entries 1548
Mean 3.103
RMS 1.817
ID 30
Entries 3096
Mean .0000E+00
RMS .0000E+00
φ FORTRAN
φ C++
Difference

Conclusion

- C++ in BES physics analysis level available
  - C++ extracts `evlst/trklst` from FORTRAN by
    - `evlst.h`
    - `trklst.h`
  - Analysis extracts information via `besEvent`
  - Histograms/Ntuples created via `cfortran` utility, class might be better
  - FORTRAN, C, and C++ interface created via careful selection of compilers and linker

- Further study need a C++ working grp
  - More bundant `besEvent`: data and member functions
  - Tested by physics analysis