Performance

EnergyPlus C++

Stuart G. Mentzer
Objexx Engineering, Inc.

What Does C++ Do Slower?

Stream i/o is very slow and ObjexxFCL layer slows it more

Compiler must assume aliasing: Limits optimization

Multi-dim array lookups: Fortran can compile as linear
  FArray linear access allows us to tune this away

Fstring heap use when Fortran may use smart allocator
  Tune C++ by eliminating unnec. object construction

x**3 is compiled as x*x*x but F2C++ gives us pow(x,3)

A modest tuning effort can usually get to par with Fortran

“Keyhole” Performance Issues

Repeat computation inside loops

Repeats of the same `A(i).B(j).C(k).D`

`any( A ⊙ B )` creates temp. array to query

Array(i,j,k) lookups in hot spots

Algorithmic Performance Issues

Don’t know much yet

Deep object access kills cache performance

Conditionals in hot spots prevent vectorization and hurt pipelining

Profiling: gprof

Doesn’t capture system library calls

Fortran and C++ are reasonably similar

Hot spots are not typical array/loop code

Sample case used for initial tuning

gprof time was brought to par with Fortran

Lib call overhead (mostly heap) was quickly tuned: C++ is 29% slower so more to do

Tuning Trial: RefrigeratedWarehouse

Build for profiling:

```
source bin/Linux/GCC/64/p/setProject
cd src/EnergyPlus
mak -j8
```

Run case: Get a `gmon.out` file

Generate the profile:

```
gprof /path/bin/Linux/GCC/64/p/EnergyPlus >profile
```

Fortran Profile

% time	cumulative seconds	self seconds	calls	name
38.26	293.98	293.98	4260981678	__fluidproperties_MOD_findarrayindex
8.26	357.42	63.44	291703665	__refrigeratedcase_MOD_calculatecompressors
5.57	400.22	42.80	1328632241	__curvemanager_MOD_curvevalue
5.13	439.65	39.43	1844931093	__fluidproperties_MOD_getinterpolatedsatprop
4.22	472.04	32.40	1062173460	__fluidproperties_MOD_getsupheatdensityrefrig
3.69	500.42	28.38	3813395	__refrigeratedcase_MOD_simulatedetailedrefrigerationsystems
3.20	524.98	24.56	291703665	__refrigeratedcase_MOD_calculatecondensers
3.13	549.01	24.03	316511785	__refrigeratedcase_MOD_calculatewalkin
2.30	566.69	17.68	908924092	__psychrometrics_MOD_psyrhoairfnpbtdbw
2.27	584.17	17.48	3852292	__refrigeratedcase_MOD_initrefrigeration
…				
0.00	768.41	0.00	1	updatemeterreporting_

C++ Profile

```
 %   cumulative  self
time  seconds seconds         calls name
27.81 278.65  278.65 4253482878 EnergyPlus::FluidProperties::FindArrayIndex(...)
 8.17 360.57   81.92  291061989 EnergyPlus::RefrigeratedCase::CalculateCompressors(int)
 5.64 417.09   56.52 1060777950 EnergyPlus::FluidProperties::GetSupHeatDensityRefrig(...)
 4.74 464.54   47.45 1840864989 EnergyPlus::FluidProperties::GetInterpolatedSatProp(...)
 3.11 495.73   31.20   92214636 ObjexxFCL::MArray1<FArray1<WarehouseCoilData>, double>::operator=(...)
 2.50 520.79   25.06 7959029579 ObjexxFCL::Fstring::~Fstring()
 2.33 544.12   23.33  291061989 EnergyPlus::RefrigeratedCase::CalculateCondensers(int)
 2.21 566.29   22.17 1065963489 EnergyPlus::FluidProperties::GetSatEnthalpyRefrig(...)
 2.20 588.39   22.10  315688425 EnergyPlus::RefrigeratedCase::CalculateCoil(int, double)
 2.10 609.44   21.05 1322923380 EnergyPlus::CurveManager::PerformanceCurveObject(...)
 2.09 630.35   20.91 7161413107 ObjexxFCL::Fstring::Fstring(char const*)
 1.85 648.86   18.51   3803475 EnergyPlus::RefrigeratedCase::SimulateDetailedRefrigerationSystems()
 1.59 664.79   15.94  907019067 EnergyPlus::Psychrometrics::PsyRhoAirFnPbTdbW(...)
 1.45 679.37   14.58  774901500 EnergyPlus::FluidProperties::GetSatSpecificHeatRefrig(...)
 1.42 693.64   14.27 1326584818 EnergyPlus::CurveManager::CurveValue(...)
 1.31 706.74   13.10    180140 EnergyPlus::UpdateDataandReport(int)
 ...
 0.0 1002.10    0.00         1 EnergyPlus::...::NeededComfortControlTypes()
```

Start at the Top: FindArrayIndex

Dominant call count and time for lookups =>

- Consider data structure
- Consider caching

For now we are just keyhole optimizing:

- Can we make it faster?
- Should it be inlined?

FindArrayIndex: Core

```
IF (Value < Array(start)) THEN
  FindArrayIndex = 0
ELSE IF (Value >Array(finish)) THEN
  FindArrayIndex = finish
ELSE ! binary search
  DO WHILE ((finish - start) > 1)
    middle = (finish + start) / 2
    IF (Value > Array(middle)) THEN
      start = middle
    ELSE
      finish = middle
    END IF
  END DO
  FindArrayIndex = start
END IF
```

```
if ( Value < Array( start ) ) {
    FindArrayIndex = 0;
} else if ( Value > Array( finish ) ) {
    FindArrayIndex = finish;
} else { // binary search
    while ( ( finish - start ) > 1 ) {
     middle = ( finish + start ) / 2;
     if ( Value > Array( middle ) ) {
         start = middle;
     } else {
         finish = middle;
     }
    }
    FindArrayIndex = start;
}
```

FindArrayIndex: Rewrite: 2+x Faster

```
if ( Value < Array( start ) ) {
    FindArrayIndex = 0;
} else if ( Value > Array( finish ) ) {
    FindArrayIndex = finish;
} else { // binary search
    while ( ( finish - start ) > 1 ) {
      middle = ( finish + start ) / 2;
      if ( Value > Array( middle ) ) {
           start = middle;
      } else {
           finish = middle;
      }
    }
    FindArrayIndex = start;
}
```

```
if ( Value < Array[ 0 ] ) {
    return 0;
} else {
    size_type end( Array.size() - 1u );
    if ( Value > Array[ end ] ) {
      return Array.u();
    } else { // Binary search
      size_type beg( 0 ), mid;
      while ( beg + 1 < end ) {
           mid = ( ( beg + end ) >> 1 ); // Bit shift
           ( Value > Array[ mid ] ? beg : end ) = mid;
      }
      return Array.l() + beg;
    }
}
```

C++ Profile: New FindArrayIndex

% time	cumulative seconds	self seconds	calls	name
15.24	123.33	**123.33**	4253482878	EnergyPlus::FluidProperties::FindArrayIndex(...)
8.31	190.61	67.28	291061989	EnergyPlus::RefrigeratedCase::CalculateCompressors(int)
5.32	233.70	43.09	**1060777950**	EnergyPlus::FluidProperties::GetSupHeatDensityRefrig(...)
4.24	267.98	34.28	1840864989	EnergyPlus::FluidProperties::GetInterpolatedSatProp(...)
4.04	300.70	32.72	92214636	ObjexxFCL::MArray1<FArray1<WarehouseCoilData>, double>::operator=(...)
3.09	325.67	24.97	1322923380	EnergyPlus::CurveManager::PerformanceCurveObject(...)
2.89	349.03	23.37	291061989	EnergyPlus::RefrigeratedCase::CalculateCondensers(int)
2.73	371.12	22.09	6898251629	ObjexxFCL::Fstring::~Fstring()
2.57	391.90	20.79	315688425	EnergyPlus::RefrigeratedCase::CalculateCoil(int, double)
2.39	411.26	19.36	3803475	EnergyPlus::RefrigeratedCase::SimulateDetailedRefrigerationSystems()
1.99	427.36	16.10	6100635158	ObjexxFCL::Fstring::Fstring(char const*)
1.95	443.11	15.75	907019067	EnergyPlus::Psychrometrics::PsyRhoAirFnPbTdbW(...)
1.90	458.50	15.39	180140	EnergyPlus::UpdateDataandReport(int)
…				
0.00	**809.23**	0.00	1	EnergyPlus::...::NeededComfortControlTypes()

CalculateCompressors

```
for ( CompIndex = 1; CompIndex <= NumComps; ++CompIndex ) {
  …
    if ( SELECT_CASE_var == RatedSubcooling ) {
      if ( System_SysNum.NumStages == 1 ) { // Single-stage system
        HCaseInRated = System_SysNum.HSatLiqCond - System_SysNum.CpSatLiqCond *
Compressor_CompID.RatedSubcool;
      } else if ( System_SysNum.NumStages == 2 && StageIndex == 1 ) { // Two-stage …
        HCaseInRated = GetSatEnthalpyRefrig( System_SysNum.RefrigerantName, System_SysNum.
TIntercooler, 0.0, System_SysNum.RefIndex, "RefrigeratedCase:CalcCompressors" ) -
System_SysNum.CpSatLiqCond * Compressor_CompID.RatedSubcool;
      } else if ( System_SysNum.NumStages == 2 && StageIndex == 2 ) { // Two-stage …
        HCaseInRated = System_SysNum.HSatLiqCond - System_SysNum.CpSatLiqCond *
Compressor_CompID.RatedSubcool;
      } // NumStages
  …
}
```

Top-of-Profile Functions

Similar to CalculateCompressors

Not much easy tuning

Did hoisting out of loops, comon subexprs, …

Why is C++ slower @ lookup, add, multiply, = ?

Performance “Bugs”

`any|all|count( A ⊙ B )` Temp boolean array!

EnergyPlus does this a lot! Even in loops!

Replaced them all:

- `any( A == B )` -> `any_eq( A, B )`
- 5% speedup

C++ Profile: Current w/ More Tuning

```
 %   cumulative  self
time  seconds seconds       calls name
14.14   98.99   98.99 3783737520 EnergyPlus::FluidProperties::FindArrayIndex(...)
 8.04 155.26    56.27  291830985 EnergyPlus::RefrigeratedCase::CalculateCompressors(int)
 4.69 188.08    32.82   92489580 ObjexxFCL::MArray1<FArray1<WarehouseCoilData>, double>::operator=(...)
 4.57 220.07    31.99  823085619 EnergyPlus::FluidProperties::GetSupHeatDensityRefrig(...)
 3.91 247.44    27.37 1845735297 EnergyPlus::FluidProperties::GetInterpolatedSatProp(...)
 3.49 271.88    24.44 1325423694 EnergyPlus::CurveManager::PerformanceCurveObject(...)
 3.43 295.90    24.02  291830985 EnergyPlus::RefrigeratedCase::CalculateCondensers(int)
 3.28 318.89    22.99  316631305 EnergyPlus::RefrigeratedCase::CalculateCoil(int, double)
 2.18 334.15    15.26 1329096472 EnergyPlus::CurveManager::CurveValue(...)
 2.16 349.24    15.09     759080 EnergyPlus::HeatBalanceIntRadExchange::CalcInteriorRadExchange(...)
 2.13 364.13    14.89    3814835 EnergyPlus::RefrigeratedCase::SimulateDetailedRefrigerationSystems()
 1.79 376.63    12.50     180078 EnergyPlus::UpdateDataandReport(int)
 ...
 0.00 700.13     0.00          1 EnergyPlus::...::NeededComfortControlTypes()
```

Fstring ctor/dtor pushed off top (below 1%) by making const local strings static and replacing literals in calls
GetSupHeatDensityRefrig call hoisted out of loop cut its and FindArrayIndex call count

Are We There Yet?

gprof says we beat the Fortran (700s/768s)

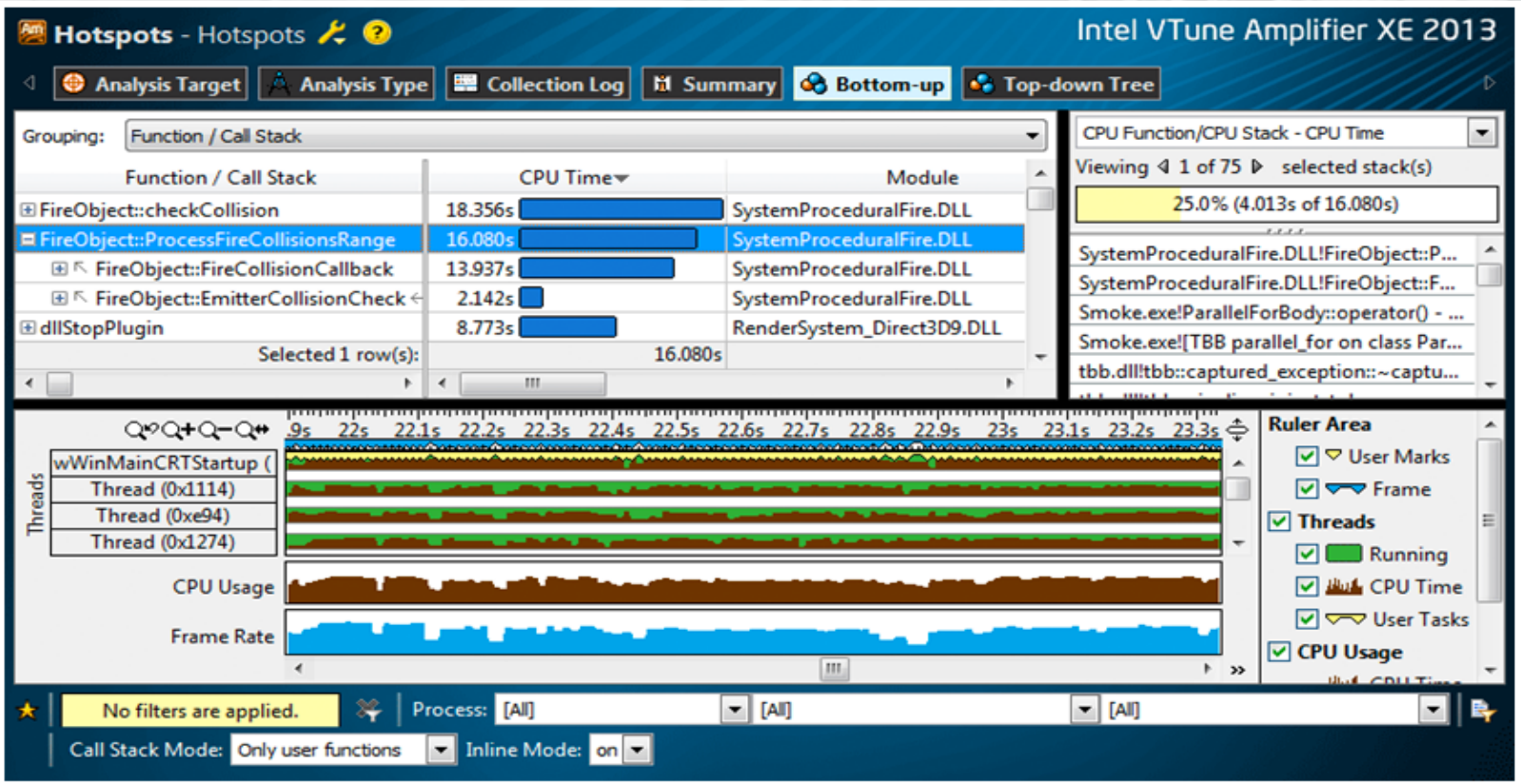
gprof doesn't measure system calls

C++ was still ~50% slower overall Why?

We need a different tool for that:

- perf Seems to be the modern/preferred Linux tool
- oprofile Similar to perf / May not work reliably
- gperftools
- Sysprof, VTune (Windows too), Zoom, …
- valgrind --tool=callgrind (slow) + KCachegrind (GUI)

VTune is Pretty



perf Usage

Build release with symbols (no -s) and (GCC)

-fno-omit-frame-pointer -ggdb

- This is the s build type in GNU Make setup

perf record -g -- /Projects/EnergyPlus/epc/bin/Linux/GCC/64/s/EnergyPlus

Already preprocessed so just give it executable

Runs case with the “s” build: Get perf.data file

perf report -g

perf Report: curses (Not Pretty)

```
Samples: 779K of event 'cycles', Event count (approx.): 535736804379
+  10.78%  EnergyPlus  EnergyPlus          [.] EnergyPlus::FluidProperties::FindArrayIndex(double, ObjexxFCL::FArray1
+   5.79%  EnergyPlus  EnergyPlus          [.] EnergyPlus::RefrigeratedCase::CalculateCompressors(int)
+   5.22%  EnergyPlus  libc-2.18.so        [.] _int_malloc
+   4.20%  EnergyPlus  libc-2.18.so        [.] _int_free
+   3.39%  EnergyPlus  EnergyPlus          [.] EnergyPlus::FluidProperties::GetInterpolatedSatProp(double, ObjexxFCL:
+   3.38%  EnergyPlus  EnergyPlus          [.] EnergyPlus::FluidProperties::GetSupHeatDensityRefrig(ObjexxFCL::Fstrir
+   3.30%  EnergyPlus  libc-2.18.so        [.] __memcpy_sse2_unaligned
+   2.95%  EnergyPlus  EnergyPlus          [.] EnergyPlus::CurveManager::PerformanceCurveObject(int, double, ObjexxF(
+   2.60%  EnergyPlus  libc-2.18.so        [.] malloc
+   2.56%  EnergyPlus  EnergyPlus          [.] ObjexxFCL::MArray1<ObjexxFCL::FArray1<EnergyPlus::RefrigeratedCase::Wa
+   2.39%  EnergyPlus  EnergyPlus          [.] EnergyPlus::RefrigeratedCase::CalculateCondensers(int)
+   2.31%  EnergyPlus  libm-2.18.so        [.] __ieee754_pow_sse2
+   2.20%  EnergyPlus  EnergyPlus          [.] EnergyPlus::RefrigeratedCase::CalculateCoil(int, double)
+   1.88%  EnergyPlus  EnergyPlus          [.] EnergyPlus::CurveManager::CurveValue(int, double, ObjexxFCL::Optional<
+   1.80%  EnergyPlus  EnergyPlus          [.] ObjexxFCL::Fstring::len_trim() const
+   1.76%  EnergyPlus  EnergyPlus          [.] ObjexxFCL::FArray1D<double>::l() const
+   1.73%  EnergyPlus  EnergyPlus          [.] EnergyPlus::RefrigeratedCase::SimulateDetailedRefrigerationSystems()
+   1.50%  EnergyPlus  libc-2.18.so        [.] strlen
+   1.36%  EnergyPlus  libm-2.18.so        [.] __ieee754_exp_sse2
+   1.23%  EnergyPlus  libm-2.18.so        [.] __exp1
+   1.18%  EnergyPlus  EnergyPlus          [.] ObjexxFCL::FArray1D<EnergyPlus::RefrigeratedCase::WarehouseCoilData>::
+   1.18%  EnergyPlus  EnergyPlus          [.] EnergyPlus::Psychrometrics::PsyRhoAirFnPbTdbW(double, double, double,
+   1.17%  EnergyPlus  EnergyPlus          [.] EnergyPlus::CalcHeatBalanceInsideSurf(ObjexxFCL::Optional<int const, v
+   1.11%  EnergyPlus  EnergyPlus          [.] EnergyPlus::HeatBalanceIntRadExchange::CalcInteriorRadExchange(ObjexxF
+   1.03%  EnergyPlus  libc-2.18.so        [.] free
```

perf Drill Down

```
Samples: 779K of event 'cycles', Event count (approx.): 535736804379
-  10.78%  EnergyPlus  EnergyPlus          [.] EnergyPlus::FluidProperties::FindArrayIndex(double, ObjexxFCL::FArray1
   - EnergyPlus::FluidProperties::FindArrayIndex(double, ObjexxFCL::FArray1<double> const&, int, int)
      + 50.16% EnergyPlus::FluidProperties::GetInterpolatedSatProp(double, ObjexxFCL::FArray1<double> const&, ObjexxF(
      + 39.42% EnergyPlus::FluidProperties::GetSupHeatDensityRefrig(ObjexxFCL::Fstring const&, double, double, int&, (
      + 7.16% EnergyPlus::FluidProperties::GetSatPressureRefrig(ObjexxFCL::Fstring const&, double, int&, ObjexxFCL::Fs
      + 1.85% EnergyPlus::RefrigeratedCase::CalculateCompressors(int)
      + 0.80% EnergyPlus::FluidProperties::GetSatEnthalpyRefrig(ObjexxFCL::Fstring const&, double, double, int&, Objex
      + 0.61% EnergyPlus::FluidProperties::GetSatSpecificHeatRefrig(ObjexxFCL::Fstring const&, double, double, int&, (
+    5.79%  EnergyPlus  EnergyPlus          [.] EnergyPlus::RefrigeratedCase::CalculateCompressors(int)
-    5.22%  EnergyPlus  libc-2.18.so        [.] _int_malloc
     _int_malloc
+    4.20%  EnergyPlus  libc-2.18.so        [.] _int_free
+    3.39%  EnergyPlus  EnergyPlus          [.] EnergyPlus::FluidProperties::GetInterpolatedSatProp(double, ObjexxFCL:
+    3.38%  EnergyPlus  EnergyPlus          [.] EnergyPlus::FluidProperties::GetSupHeatDensityRefrig(ObjexxFCL::Fstrir
+    3.30%  EnergyPlus  libc-2.18.so        [.] __memcpy_sse2_unaligned
+    2.95%  EnergyPlus  EnergyPlus          [.] EnergyPlus::CurveManager::PerformanceCurveObject(int, double, ObjexxF(
-    2.60%  EnergyPlus  libc-2.18.so        [.] malloc
   - malloc
      + 13.90% EnergyPlus::RefrigeratedCase::CalculateCompressors(int)
      + 10.59% EnergyPlus::RefrigeratedCase::SimulateDetailedRefrigerationSystems()
      + 8.51% EnergyPlus::Psychrometrics::PsyTsatFnHPb(double, double, ObjexxFCL::Optional<ObjexxFCL::Fstring const, \
      + 7.19% EnergyPlus::Psychrometrics::PsyWFnTdbTwbPb(double, double, double, ObjexxFCL::Optional<ObjexxFCL::Fstrir
      + 5.64% EnergyPlus::Psychrometrics::PsyHFnTdbRhPb(double, double, double, ObjexxFCL::Optional<ObjexxFCL::Fstring
      + 5.55% EnergyPlus::Psychrometrics::PsyWFnTdbRhPb(double, double, double, ObjexxFCL::Optional<ObjexxFCL::Fstring
      + 4.08% EnergyPlus::ZoneEquipmentManager::SetZoneEquipSimOrder(int, int)
```

perf Observations

The better we do the more diffuse the hot spots!

malloc/free are still high up: Need more love

- Pool allocators are a possibility but sig. effort

Lack of parents for _int_malloc:

- perf record -gdwarf may fix this

pow and exp: Hoisting and square/cube

Does this make gprof obsolete?

perf Conclusions & Action

Heap allocs were big and corr. with call count

First step was to make const local strings static

- Fortran is being smart about avoiding heap

This & tuning other hot spots found by perf got C++ down to ~30% slower in a few hours

Then focus shifted to 8.2

Not sure how far we'll get it down in this phase

Tuning Plan

Profile and tune a few representative cases to get as close to Fortran performance as we can

Make notes about the process for team use

Deeper tuning and performance investigation

Safeguarding Performance

Performance benchmark “test” suite?

Catch performance regressions while “fresh”

Wiki page on profiling for developers

Concepts for Very High Performance

Reduce conditionals in p-c code
Change conditionals to booleans
Small, local state variables
Tick-tock design: synchronous state chg/upd
Cache state or drilled refs to state inputs
Pipelining & branch prediction
Elim chains of refs/ptrs for state checks
Parallelize: Decouple by zones/subsystems

Questions