Measurements For A Tailored Number System

Andrew Shewmaker
What to measure for a custom number system?

- **Dynamic Range**
  - avoid wasting bits on needless range

- **Density**
  - devote more bits where they matter

- **Literals and Mathematical Constants**
  - enable natural expressions despite limitations of a given base (0.1, pi, tau, etc.)

- **Operations, Functions, and Patterns**
  - focus implementation on most critical functionality and optimizations
How to measure?

- Application output
- Compiler assisted
- Emulation
- Binary instrumentation
Jet mixfrac

Frequency of Numbers in jet mixfrac

Count

Value

0.00
0.25
0.50
0.75
1.00

3e+06
2e+06
1e+06
0e+00
Frequency of Numbers in snap.out
Configurable Runtime Analysis for Floating-point Tuning

- [http://blog.freearrow.com/software/craft](http://blog.freearrow.com/software/craft)
- Detects numerical cancellation
- Automates search for mixed precision replacement
- Performs generalized precision analysis
- Tracks ranges of numbers used in each instruction
  - floatwatch made histograms [http://www.ashleybrown.co.uk/index.php/academia/phd/](http://www.ashleybrown.co.uk/index.php/academia/phd/)
- Histogram analysis prototype available at [https://github.com/agshew/craft](https://github.com/agshew/craft)
  - bins of various widths across the real number line
  - points of interest
  - global and per instruction
  - input and output operands
  - timeseries
CLAMR - adaptive mesh refinement mini-app

Counts of Numbers over Time CLAMR-timeseries-in.dat
CLAMR - adaptive mesh refinement mini-app

Counts of Numbers in CLAMR-poi-in.dat

Counts of Numbers in CLAMR-poi-out.dat
NuT - Monte Carlo code for Neutrino Transport

Counts of Numbers in NuT-in.dat

Value

Count
NuT - Monte Carlo code for Neutrino Transport

Counts of Numbers in NuT-poi-in.dat

Counts of Numbers in NuT-poi-out.dat
VPIC - particle-in-cell modeling kinetic plasmas

Counts of Numbers in VPIC-poi-in.dat

Counts of Numbers in VPIC-poi-out.dat

Count (log10)
Livermore Unstructured Lagrangian Explicit Shock Hydrodynamics
Livermore Unstructured Lagrangian Explicit Shock Hydrodynamics

Counts of Numbers in LULESH-poi-in.dat

Value

Count (log10)

0 0.3 0.6 0.9 1

1e+01 1e+03 1e+05
Next Steps

- Fix segfault when running LULESH
- Estimate cardinality with Hyperloglog++
- Track heavy hitters with Count-min sketch
- Improve on Dynamic Average-Deviation Optimal (DADO) histograms
  - split when number of unique values exceeds threshold
  - split on median instead of mean
  - separate heavy hitters?
- Bursty statistical sampling
- Submit pull request to upstream
- Make CRAFT easier to deploy (packages, containers, etc.)
- Analyze real codes and inputs
Build CRAFT

- Install dependencies according to README.md and locations set up environment
  - may have to build libdwarf with ./configure --enable-shared
  
  CRAFT_DIR=$HOME/src/craft
  export PLATFORM=x86_64-unknown-linux2.4
  export PATH=$CRAFT_DIR/$PLATFORM:$CRAFT_DIR/scripts:$PATH
  export LD_LIBRARY_PATH=:.$CRAFT_DIR/$PLATFORM:/usr/local/lib:$LD_LIBRARY_PATH
  export DYNINST_ROOT=$HOME/src
  export DYNINSTAPI_RT_LIB=/usr/local/lib/libdyninstAPI_RT.so

- cd $HOME/src; git clone https://github.com/agshew/craft.git/; cd craft; git checkout histograms

- Edit Makefile
  XED_KIT=$(HOME)/pin-2.14-71313-gcc.4.4.7-linux/extras/xed-intel64
  LOCAL_INC_DIRS = -I$(HOME)/src/dyninst-9.1.0
  LOCAL_LIB_DIRS = -L$(HOME)/src/dwarf-20120410/libdwarf

- make -j$(nproc)
Using CRAFT to perform histogram analysis

cd $HOME/src/craft
cd demo/sum2pi_x
make
fpinst --thistogram ./sum2pi_x

plot_poi hist-${PID}-poi-out
plot_poi hist-${PID}-poi-in
plot_hist_timeseries hist-${PID}-all-out-timeseries
plot_hist_timeseries hist-${PID}-all-in-timeseries
plot_hist hist-${PID}-all-in
plot_hist hist-${PID}-all-out

# view hist-${PID}-*png