Building and Running ESESC Tutorial

Speaker: Daphne Gorman

Department of Computer Engineering,
University of California, Santa Cruz
http://masc.soe.ucsc.edu

Building and Running

- You will learn:
  - To compile ESESC
  - High level view of code structure
  - Run a simple application
  - Overview of `esesc.conf`
  - Simple analysis of results

Outline

- Obtaining and Building ESESC
- ESESC Code Structure
- Running ESESC
- Getting ESESC Output

Getting ESESC

Repo:

- https://github.com/masc-ucsc/esesc

Online tutorials:

- http://masc.soe.ucsc.edu/esesc

Building

- Getting the code
  - `git clone https://github.com/masc-ucsc/esesc.git`
- Directory structure
  - ~/projs/esesc – source directory
  - `ls ~/projs/esesc`
- Create Build directory
  - `mkdir -p ~/build/debug`
  - `mkdir -p ~/build/release`

Building

- Two modes
  - Debug
    - Slower, more information
  - Release
    - Faster, less information

Building

- Build
  ```
  cd ~/build/release
  cmake ~/projs/esesc
  make
  ```

Setup Run Directory

- Create a run directory

```
cd ~/build/release
mkdir run
cd run
```

- Copy configuration files

```
cp ~/projs/esesc/conf/* .
```

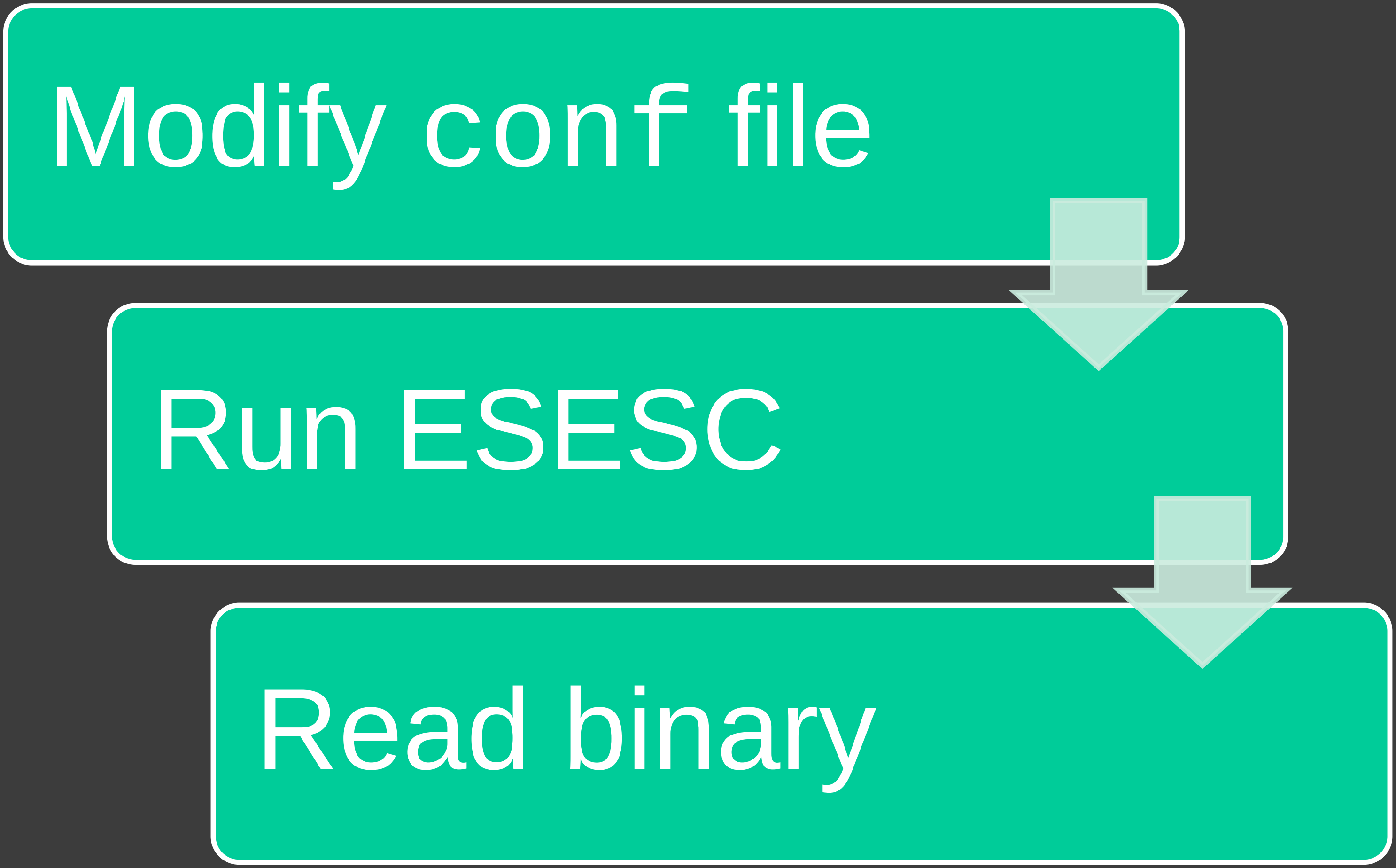
- Copy binaries to simulate

```
cp ~/projs/esesc/bins/* .
```

Outline

- Obtaining and Building ESESC
- ESESC Code Structure
- Running ESESC
- Getting ESESC Output

ESESC Flow



Top level configuration file: esesc.conf

- Overview
- benchName parameter:
  - Point to a static ARMv7 binary
  - Pass arguments

```
benchName  = "myProgram  myArguments"
```

Launcher

- Supports running multiple (different) benchmarks simultaneously
  - Spawns each benchmark as a thread
  - Supports SPEC Rate type runs
- Suites
  - CPU 2000/2006
  - PARSEC
  - SPLASH
- Usage:

$\texttt{launcher [-- rloop] [-- stdin <file>] -- }\underbrace{\texttt{<benchname> [args]}}_{\text{One or more times}}$

Outline

- Obtaining and Building ESESC
- ESESC Code Structure
- Running ESESC
- Getting ESESC Output

Run Release Mode

- From the release build directory, run:

```
~/build/release/main/esesc
```

- Check results:

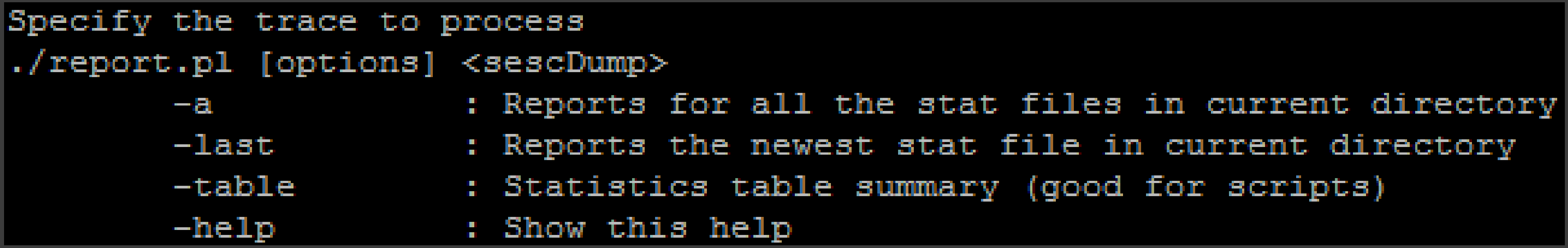
```
~/projs/esesc/conf/scripts/report.pl -a
```

Outline

- Obtaining and Building ESESC
- ESESC Code Structure
- Running ESESC
- Getting ESESC Output

report.pl

- `report.pl` is executable script for displaying stats from the ESESC run, using a dump



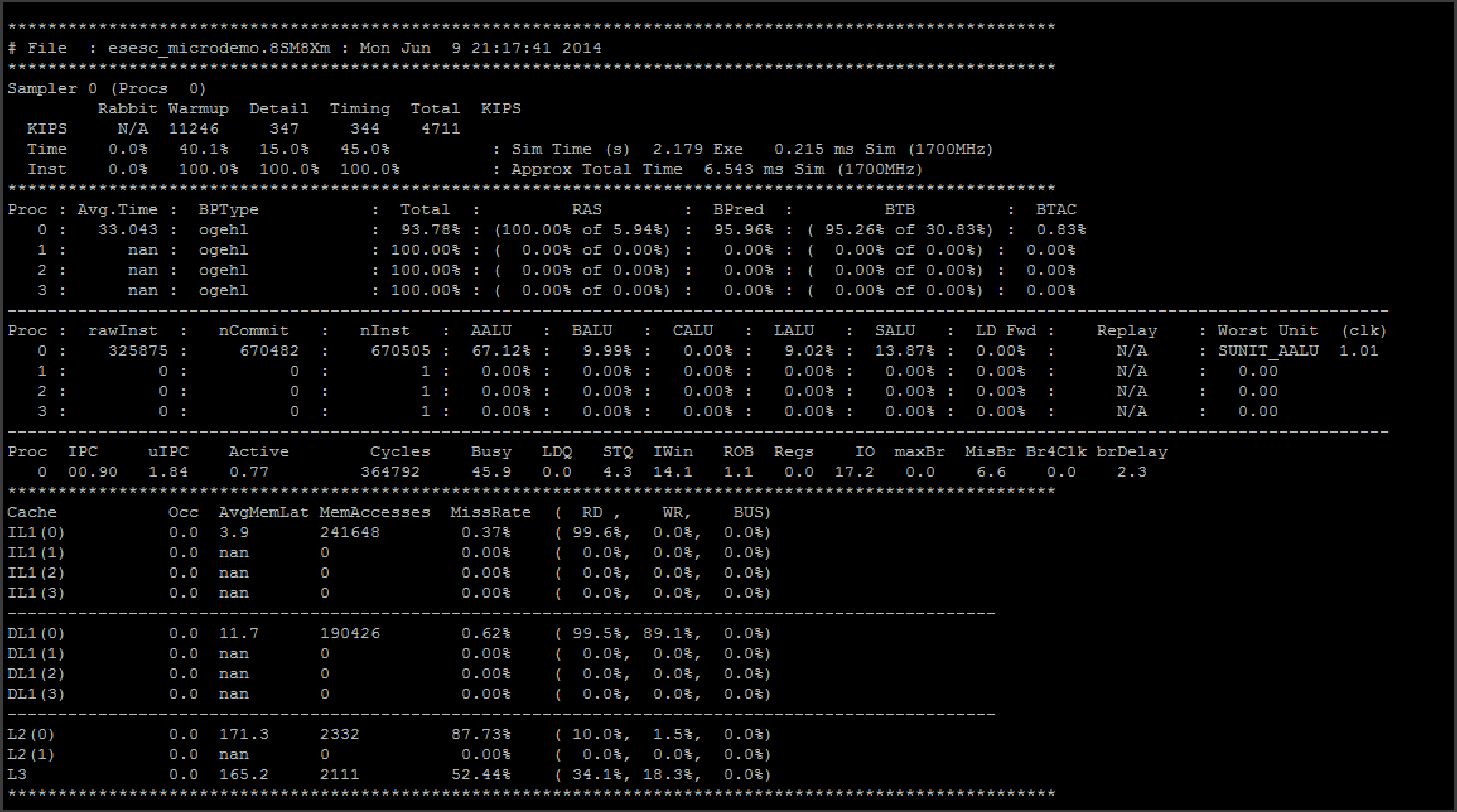
```
Specify the trace to process
./report.pl [options] <sescDump>
        -a              : Reports for all the stat files in current directory
        -last           : Reports the newest stat file in current directory
        -table          : Statistics table summary (good for scripts)
        -help           : Show this help
```

- The “./report.pl -a” or “./report.pl -last” commands most common to use

report.pl

- Memory Read/Writes, Caches, IPC, Instruction counts, Cycles
- Note: All time units are in **cycles**
- What various fields mean
  - AALU: Arithmetic, Logic (execute stage)
  - BALU: Branching
  - CALU: Complex Unit
  - LALU: Loads
  - SALU: Stores
  - B*, br*, or *Br*: Branch-related statistics

Stats from report.pl



```
********************************************************************************************************
# File  : esesc_microdemo.8SM8Xm : Mon Jun  9 21:17:41 2014
********************************************************************************************************
Sampler 0 (Procs  0)
         Rabbit Warmup  Detail  Timing  Total  KIPS
  KIPS     N/A  11246     347     344    4711
  Time    0.0%   40.1%   15.0%   45.0%          : Sim Time (s)  2.179 Exe   0.215 ms Sim (1700MHz)
  Inst    0.0%   100.0%  100.0%  100.0%         : Approx Total Time  6.543 ms Sim (1700MHz)
********************************************************************************************************
Proc : Avg.Time :  BPType           :  Total   :          RAS          :  BPred  :          BTB          :  BTAC
   0 :   33.043 :  ogehl            :  93.78% : (100.00% of 5.94%) :  95.96% : ( 95.26% of 30.83%) :  0.83%
   1 :      nan :  ogehl            : 100.00% : (  0.00% of 0.00%) :   0.00% : (  0.00% of 0.00%) :  0.00%
   2 :      nan :  ogehl            : 100.00% : (  0.00% of 0.00%) :   0.00% : (  0.00% of 0.00%) :  0.00%
   3 :      nan :  ogehl            : 100.00% : (  0.00% of 0.00%) :   0.00% : (  0.00% of 0.00%) :  0.00%
-----------------------------------------------------------------------------------------------------------------------------------------
Proc :  rawInst  :   nCommit   :   nInst   :  AALU   :  BALU   :  CALU   :  LALU   :  SALU   :  LD Fwd :    Replay    : Worst Unit  (clk)
   0 :   325875 :    670482   :    670505 : 67.12% :   9.99% :   0.00% :   9.02% :  13.87% :  0.00%  :      N/A     : SUNIT_AALU  1.01
   1 :        0 :         0   :         1 :   0.00% :  0.00% :   0.00% :   0.00% :   0.00% :  0.00%  :      N/A     :   0.00
   2 :        0 :         0   :         1 :   0.00% :  0.00% :   0.00% :   0.00% :   0.00% :  0.00%  :      N/A     :   0.00
   3 :        0 :         0   :         1 :   0.00% :  0.00% :   0.00% :   0.00% :   0.00% :  0.00%  :      N/A     :   0.00
-----------------------------------------------------------------------------------------------------------------------------------------
Proc  IPC     uIPC    Active        Cycles    Busy   LDQ   STQ  IWin   ROB  Regs    IO  maxBr  MisBr Br4Clk brDelay
   0  00.90   1.84    0.77         364792    45.9   0.0   4.3  14.1   1.1   0.0  17.2   0.0    6.6    0.0    2.3
********************************************************************************************************
Cache           Occ  AvgMemLat MemAccesses  MissRate  (  RD ,    WR,     BUS)
IL1(0)          0.0  3.9       241648        0.37%    ( 99.6%,  0.0%,  0.0%)
IL1(1)          0.0  nan       0             0.00%    (  0.0%,  0.0%,  0.0%)
IL1(2)          0.0  nan       0             0.00%    (  0.0%,  0.0%,  0.0%)
IL1(3)          0.0  nan       0             0.00%    (  0.0%,  0.0%,  0.0%)
--------------------------------------------------------------------------------------------------
DL1(0)          0.0  11.7      190426        0.62%    ( 99.5%, 89.1%,  0.0%)
DL1(1)          0.0  nan       0             0.00%    (  0.0%,  0.0%,  0.0%)
DL1(2)          0.0  nan       0             0.00%    (  0.0%,  0.0%,  0.0%)
DL1(3)          0.0  nan       0             0.00%    (  0.0%,  0.0%,  0.0%)
--------------------------------------------------------------------------------------------------
L2(0)           0.0  171.3     2332         87.73%    ( 10.0%,  1.5%,  0.0%)
L2(1)           0.0  nan       0             0.00%    (  0.0%,  0.0%,  0.0%)
L3              0.0  165.2     2111         52.44%    ( 34.1%, 18.3%,  0.0%)
********************************************************************************************************
```

Demo: Build Run ESESC

- Build ESESC in Release mode
- Run a simple benchmark in release mode.
- Check results
- Build ESESC in Debug mode

Summary

- Ran ESESC for the first time
- Gather some statistics
- A high level idea of the code structure