MO601 – Project 1

093311 – Alberto Arruda de Oliveira

Instruction Count

- To count instructions, the Pin Tool *inscount0.cpp* was used
  - Increments a counter for each instruction called

- Several benchmarks have multiple inputs
  - Each execution is named *Workload*
  - The Parameter $Workload can be used to identify different inputs;
  - Checking stdout with full verbosity

- Each output file is named with their benchmark name and workload

Branch Count

- Custom Pin Tool developed for this project;
- Counts the number of instructions that cause branches in the code;
  - Use of Pin's command *INS_isBranch(ins)* to check for branches;
  - Counts also the total number of instructions and the instructions that do NOT cause branches;
- Besides being run with SPEC's benchmarks, another experiment was conducted on simple C++ programs;

Branch Counting Simple C++ Programs

- The idea was to count the branches and instructions of the simplest possible C++ programs;

- Start from the simples thought (not empty) C++ code, then incrementally add lines to check their impact in the instruction and branch count;

- Also check if there is a basic number of instructions present even in the simplest C++ code;

Branch Count – Main and Return

- Is this the simplest C++ code possible?

```
int   main(void) {

   return 0;
}
```

Program	Total Instructions	Branch Instructions	Not Branch Instructions
Main and Return	87710	15501	72209

Branch Count – Main and Return

```
int   main(void) {

    int k = 0;
    return 0;
}
```

Program	Total Instructions	Branch Instructions	Not Branch Instructions
Main and Return	87710	15501	72209
Attribution k = 0	87711	15501	72210

Branch Count – Main and Return

```
int   main(void) {

     int k = 0;

     if(k < 10)

     return 0;
}
```

Program	Total Instructions	Branch Instructions	Not Branch Instructions
Main and Return	87710	15501	72209
Attribution k = 0	87711	15501	72210
Satisfied **IF**	87714	15503	72211

Branch Count – Main and Return

```
int   main(void) {

    int k = 0;

    if(k > 10)

    return 0;
}
```

Program	Total Instructions	Branch Instructions	Not Branch Instructions
Main and Return	87710	15501	72209
Attribution k = 0	87711	15501	72210
Satisfied **IF**	87714	15503	72211
Non-Satisfied **IF**	87713	15502	72211

Branch Count – Main and Return

```
int   main(void) {

    int k = 0;

    if(k > 10);
    else

    return 0;
}
```

Program	Total Instructions	Branch Instructions	Not Branch Instructions
Main and Return	87710	15501	72209
Attribution k = 0	87711	15501	72210
Satisfied **IF**	87714	15503	72211
Non-Satisfied **IF**	87713	15502	72211
IF-ELSE	87714	15503	72211