

Project:

Write a Pintool in Probe mode called **project.so**.

The pintool should be based on the pintool from exercise 4 that counts the number of times each basic block and each target jump was executed.

Use the provided pintool source code “**btranslate-for-project.cpp**” in the moodle.

The pintool spawns a thread that, based on the gathered profiling, creates a 2nd Translation Cache called TC2, where execution continues from there.

The **project.so** pintool receives 2 knobs: “**-create_tc2**” for generating TC2 and “**-prof_time**” that sets the number of seconds for the profiling duration (default is set to 2 seconds).

For example, the following command line:

```
<pindir>/pin -t project.so -create_tc2 -prof_time 4 -- ./cc1 expr.i
```

creates the TC2 cache after 4 seconds of running the code in TC containing the profiling code that counts the execution frequency of every basic block and every target jump.

The objective of the project is to reduce the overhead time of collecting the profiling data and the overhead of running the code in TC and TC2 by:

- Minimizing the number of profiling instructions added to each BBL.
For example, use **killed** regs to hold temporary values of RAX, RBX or RCX, instead of saving and restoring them from memory.
- Reduce the transition overhead during TC2 execution.
Use the collected profiling data for each indirect jump + collect it for each indirect call as well (collected during TC execution) to apply the **de-virtualization** optimization which replaces the most frequent jump targets by direct jumps/calls.
- **Bonus:** applying additional optimizations that can improve performance during TC2 execution, e.g., code reordering, loop unrolling or function inlining.

The total elapsed time of running the pintool on **cc1** with **expr.i** input, **minus** the elapsed time that it takes to create TC at load time, must be only up to **15%** slower than the elapsed time of running **cc1** on **expr.i** without pin.

Measuring the elapsed time of running the cc1:

```
time ./cc1 expr.i
```

(Note: the above command generates an assembly file called **expr.s**).

Measuring total elapsed time for running the project pintool on cc1:

```
time ../../pin -t obj-intel64/project.so -create_tc2 -- ~/workdir/gcc/cc1 expr.i
```

The elapsed time of creating TC can be found in the cerr printouts as follows:

```
“create_tc took: <TC creation elapsed time> seconds”
```

You can deduct the above TC creation elapsed time from the overall elapsed time of running the pintool project.

Programming Tips:

Consider using the following available debug functions in **btranslate-for-project.cpp**:

1. **dump_instr_from_xedd()**
2. **dump_instr_from_mem()**
3. **dump_entire_instr_map()**
4. **dump_tc()** for TC and TC2, using the knobs: **-dump_tc** and **-dump_tc2** correspondingly.

Submission requirements:

The submission of this project is **in pairs only**.

Submit 1 compressed file called "**project.zip**" into the moodle project [link](#) containing the following files:

1. The binary of your pintool **project.so** (compiled, and tested by you that it runs and gives the result).
2. A directory called: 'src' containing all the sources of your pintool along with the make files and a README.txt file that includes the following:
 - a. names + id numbers
 - b. compilation command
 - c. how to run the tool.
 - d. Describe the threshold used to distinguish between frequently and rarely executed instructions.

Submission deadline: midnight Sunday October 26, 2025.

Project defense:

Before the submission date, we will publish several dates and time slots for each pair to choose from, in order to defend their project.

There is an option to defend the project via zoom.

During defense, **each pair must present foils** describing:

1. The problem that you are trying to solve
2. The solution which includes presentable walk thru on the code implementation.
3. Measurement results in the form of a **graph**.

Foils need to include a presentable walk thru on the code implementation.

Good Luck!