Assignment3: Bottleneck analysis using Tejas

Part A

Get source code of the Tejas architectural simulator from here.

How to set up Tejas:

- Configuration file set-up:
  Start with the configuration file (`config_tigerLake.xml`) and NOC configuration file (`NocConfig.txt`) given in the folder `tejas_resources` in the CS810_resources repository. Make copies of these files and modify them as required.
    - Set `EmulatorType` to `none`
    - Set `CommunicationType` to `file`
    - Set `StoreExecutionTraceInAFile` to `false`
    - Set `subsetSimSize` to the number of instructions that you want to simulate
    - Set `ToCompute` (under `Criticality`) to `false`
    - Set `ReadingFromLeanTrace` to `false`
    - Set `NocConfigFile` to the full path to the `NocConfig.txt` file
- Go to the folder `src/simulator/pipeline/branchpredictor/TAGESCL/`, and run `make`. If the make succeeds, you can use the state-of-the-art branch predictor TAGE-SC-L by (1) setting the `Predictor_Mode` (under the `BranchPredictor` tag in the configuration file) to `TAGE-SC-L`, and (2) setting the `TAGESCLLibDirectory` to the full path to the folder where you just ran make. If the make fails, and you are not able to debug it, then you may use a slightly older branch predictor by setting the `Predictor_Mode` to `TAGE`. The `TAGESCLLibDirectory` tag need not be updated in this case.
- Run `ant make-jar` in the root folder (the folder that contains the file `build.xml`). This will create the file `tejas.jar` in the folder `jars/`.
- To run Tejas, run `java -jar <path-to-jar-file> <path-to-config-file> <path-to-stat-file-to-be-created> <path-to-trace-file>`
    - Trace files are available in the folder `CPU2017_benchmarks/tejas_traces/` in the CS810_resources repository.
    - Example: `java -jar /xyz/tejas.jar /xyz/config_tigerLake.xml /xyz/gcc.stat /xyz/gcc`
    - This command simulates the execution of (a portion) of the *gcc* benchmark on a processor that is described in `config_tigerLake.xml` and records the statistics in the file `gcc.stat` .

- The above example command assumes that `tejas.jar`, `config_tigerLake.xml`, `gcc_0.gz`, and `gcc_1.gz` are found in the folder `/xyz/`.
- Note that in the command, we should not write `_0.gz` or `_1.gz` while mentioning the trace file.

Part B

Play around with the configuration file and identify the “bottleneck” for each SPEC CPU2017 benchmark application. That is, attempt to identify which components of the architecture play the greatest role in determining the application’s performance. Feel free to explore even unrealistic scenarios like perfect caches and perfect branch predictors to help you in your investigation.

Your report must include detailed analysis of each benchmark, with supporting plots as evidence for your claims.

Deadline: 23:55, 20/09/2023 (3 weeks)
Marks: 7.5