CMPEN 431 Project 1 - Design Space Exploration

Video Link: Project Overview

Sethu Jose
Teaching Assistant
Spring 2022

Disclaimer

1. This presentation is just an overview. It is missing a lot of details. Project guide should always be your primary source of information on what to implement and what are the deliverables.
2. If in doubt about the project framework, refer the source code. Understand how it works.

Agenda

- Project Overview
- Connecting to the server and setting up project
- Code walkthrough:
  - runprojectsuite.sh
  - 431project.cpp/h
  - 431projectUtils.cpp
  - YOURCODEHERE.cpp
  - Outputs

Project Overview

Design Space Exploration (DSE)

- DSE is the process of finding the best solution to a problem under given constraints.
- In this project, DSE refers to finding the best settings for a range of design parameters of a computer architecture so that it maximizes either of the following:
  - Performance (measured in terms of execution time)
  - Energy-Delay Product (EDP)
    - A metric that combines both performance and energy efficiency.
    - EDP = energy consumed * execution time
- Two types of DSE explorations:
  - Exhaustive: If the number of possible solutions is large, DSE may explore all of them individually
  - Intelligent heuristic: prunes down the design space to prioritize evaluation nof more reasonable design points first

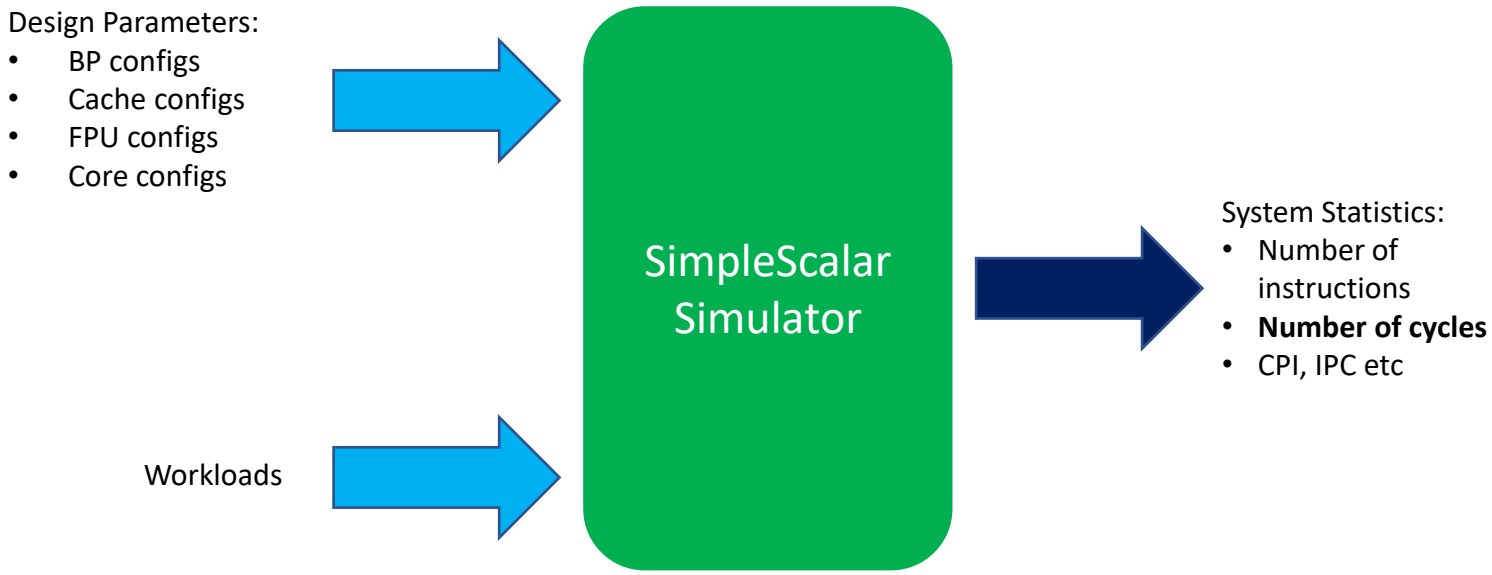
DSE space

- The space or set of parameters over which DSE exploration is done.
- In this project, we have the following groups of design space to explore:
  1. Branch Predictor (BP)
     - Branch settings, RAS, btb etc
  2. Cache Configurations
     - {l1, ul2} block, {dl1, il1, ul2} sets, {dl1, il1, ul2} assoc, associated latencies etc.
  3. Core Configurations
     - Width, scheduling etc.
  4. Floating Point Unit (FPU)
     - Fpwidth
- In total, our Design space has a total of 18 parameters:
  - 15 independent parameters
  - 3 dependent parameters

SimpleScalar

- Architecture Simulator:
  - program that simulates the execution of a computer architecture
  - Allows to modify architecture parameters in code without making an hardware changes
  - SimpleScalar simulates architecture called PISA. Its similar to MIPS.
- Workflow:
  1. Accepts a set of design parametres and an executable (workload) as input
  2. Adjusts the architecture with the parameters specified and runs the executable on this modified architecture
  3. Generates a wide range of system statistics like number of instructions simulated, execution time, number of cycles etc.

Project 1 overview



Full list of design parameters:

["width", "scheduling", "l1block", "dl1sets", "dl1assoc", "il1sets", "il1assoc", "ul2sets", "ul2block", "ul2assoc", "replacepolicy", "fpwidth", "branchsettings", "ras", "btb", "dl1lat", "il1lat", "ul2lat"]

Our Heuristic

1. Design space dimensions can be labelled as either explored and unexplored.
2. Initially all dimensions are unexplored.
3. Choose an unexplored dimension, exit if all dimensions are explored.
    1. Evaluate all possible design points by changing the value of this dimension only.
    2. Fix value of this dimension by selecting the best design so far (consider DSE goal).
    3. Mark this dimension as explored
4. Go to step 3.

How to explore?

["width", "scheduling", "l1block", "dl1sets", "dl1assoc", "il1sets", "il1assoc", "ul2sets", "ul2block", "ul2assoc", "replacepolicy", "fpwidth", "branchsettings", "ras", "btb", "dl1lat", "il1lat", "ul2lat"]

- *Formulate your dimension array using the heuristic.*

How to explore?

0	0	0	5	0	5	0	2	2	2	0	1	0	1	2	2	2	5

["width", "scheduling", "l1block", "dl1sets", "dl1assoc", "il1sets", "il1assoc", "ul2sets", "ul2block", "ul2assoc", "replacepolicy", "fpwidth", "branchsettings", "ras", "btb", "dl1lat", "il1lat", "ul2lat"]

- *Use the baseline configuration as the default value for all unexplored dimensions*

How to explore?

["width", "scheduling", "l1block", "dl1sets", "dl1assoc", "il1sets", "il1assoc", "ul2sets", "ul2block", "ul2assoc", "replacepolicy", "fpwidth", "branchsettings", "ras", "btb", "dl1lat", "il1lat", "ul2lat"]

- *Possible values of width=("1" "2" "4" "8")*
- *Value in the array is the index into this array of possible width. So, a 0 means width value of 1 is chosen.*

- *Similarly, 4th dimension is dl1sets. Its possible value array is dl1sets=("32" "64" "128" "256" "512" "1024" "2048" "4096" "8192"). A value of 5 indicates dl1sets is set to 1024.*

How to explore?

["width", "scheduling", "l1block", "dl1sets", "dl1assoc", "il1sets", "il1assoc", "ul2sets", "ul2block", "ul2assoc", "replacepolicy", "fpwidth", "branchsettings", "ras", "btb", "dl1lat", "il1lat", "ul2lat"]

- Try out all possible settings for each dimension while keeping the rest constant.
- Pick the one with the least execution time if the DSE performance is done. Pick one with least EDP if DSE energy is run.
- For example, lets assume width of 4 gives the best execution time in this scenario
  - If so, set width to a value of 2.
  - Then explore the next dimension

How to explore?

["width", "scheduling", "l1block", "dl1sets", "dl1assoc", "il1sets", "il1assoc", "ul2sets", "ul2block", "ul2assoc", "replacepolicy", "fpwidth", "branchsettings", "ras", "btb", "dl1lat", "il1lat", "ul2lat"]

How many possible values are there for each dimension?

- *Given in the dimension cardinality array*

How to explore?

["width", "scheduling", "l1block", "dl1sets", "dl1assoc", "il1sets", "il1assoc", "ul2sets", "ul2block", "ul2assoc", "replacepolicy", "fpwidth", "branchsettings", "ras", "btb", "dl1lat", "il1lat", "ul2lat"]

- *While exploring a new dimension, make sure you assign the already explored dimensions to their "best" settings.*
- *Currently explored dimension is assigned a new unexplored value*
- *Unexplored dimensions are set to the baseline value*

How to explore?

["width", "scheduling", "l1block", "dl1sets", "dl1assoc", "il1sets", "il1assoc", "ul2sets", "ul2block", "ul2assoc", "replacepolicy", "fpwidth", "branchsettings", "ras", "btb", "dl1lat", "il1lat", "ul2lat"]

- *Whenever you change any cache settings, you need to re-calculate the cache latency parameters according to the constraints given in section 8.3 of the project guide*
- *Assign these re-calculated latencies to the last three dependent dimensions*

How to explore?

["width", "scheduling", "l1block", "dl1sets", "dl1assoc", "il1sets", "il1assoc", "ul2sets", "ul2block", "ul2assoc", "replacepolicy", "fpwidth", "branchsettings", "ras", "btb", "dl1lat", "il1lat", "ul2lat"]

- *While starting to explore a new dimension, you may have already explored the new dimension's baseline value.*
  - *Can you skip re-running this baseline value?*

Connecting to Server

Code Walk-through

Deliverables