

GPUTEjas Instruction Manual v2

August 5, 2024

1 Introduction

GPUTEjas is a new Java-based parallel GPGPU simulator built by the Srishti research group, IIT Delhi. It is a fast trace driven simulator, which uses relaxed synchronization, and non-blocking data structures to derive its speedups. Secondly, it introduces a novel scheduling and partitioning scheme for parallelizing a GPU simulator.

The following document provides all the steps necessary to install and run GPUTEjas.

2 Installation:

1. Download GPUTEjas from the link below:

```
http://www.cse.iitd.ac.in/tejas/gputejas/home\_files/gputejas\_installation\_kit.zip
```

2. Check that your system meets the requirements:

- GPU with SM compute capability: ≥ 3.5 and ≤ 8.6
- OS: Linux
- Java: version $\geq 8.x$
- GCC version: $\geq 5.3.0$
- CUDA version: ≥ 8.0 and $\leq 11.x$
- nvcc version for tool compilation ≥ 10.2

3. Make sure you have CUDA toolkit installed on your system and *nvcc* is in your PATH.
4. Extract the contents and go into the main gputejas directory.

```
unzip gputejas_installation_kit.zip
cd gputejas
```

5. To install the NVBit instrumentation library please clone the following Github:

```
git clone https://github.com/CoffeeBeforeArch/nvbit_tools.git
```

6. We created a tracer tool on top of *NVBit*. To install it, follow the steps given below.

```
mkdir temp
cd temp
git clone https://github.com/Yashashwee/nvbit_tools.git
cd ..
```

7. Copy the files to the original NVBit directory:

```
rm -r nvbit_tools/core nvbit_tools/test-apps/ nvbit_tools/tools
cp -r temp/nvbit_tools/* nvbit_tools/
rm -rf temp
```

8. Follow the steps below to install GPUtejas:

```
chmod +x setup_gputejas.sh
./setup_gputejas.sh setup
```

This will install GPUtejas and all required components with it.

9. Change the config.xml file in the config directory to match your desired configuration and set the maximum number of simulation threads you want. The config can be found at the following location inside the gputejas directory:

```
gputejas/src/simulator/config/config.txt
```

For a primer on the parameters in the config file refer to Section 3.

3 Config Parameters:

The number of parameters in the config file may seem overwhelming to a new user. The three main sections for running simulations are:

- **Simulation:** Contains the parameters:

1. *MaxNumJavaThreads*: To set the number of Java threads running the simulation.
 2. *MaxNumBlocksPerJavaThread*: To set the maximum number of simulation blocks to run on each thread.
 3. *GPUType*: Choose between {Tesla, Fermi, Maxwell, Kepler, Pascal, Ampere}.
 4. *ThreadsPerCTA*: To set the number of threads to run per CTA, depending on the type scheduling you want.
- **System**: Contains parameters of the GPU being simulated, like number of Texture Processor Clusters, Streaming Multiprocessors and their details.
 - **Library**: Contains the collection of simulation elements like Caches and their details.

Apart from these there are two other sections *OperationLatency* and *OperationEnergy* which contain operation wise latency and energy requirements.

4 Generating Traces:

1. Compile your benchmark using the CUDA Compiler. For instance we include the a simple vector addition benchmark with the installation kit. Compile it using the following command.

```
./setup_gputejas.sh make-example
```

or manually make with:

```
cd nvbit_tools/test-apps/vectoradd/
make clean
make
cd ../../..
```

2. Make the required changes to the config.xml file
3. Go into the main GPUtejas directory and run the following command:

```
./setup_gputejas.sh genTrace
```

4. Follow the instruction that appear on the screen.
5. At the end you must have the traces divided into simulation threads and further divided among kernel. With this you also have the kernel-wise hashed instructions.

5 Run Simulation

1. Re-build GPUtejas every time you make changes in the code:

```
./setup_gpuतेjas.sh make-jar
```

2. Run the simulator:

```
./setup_gpuतेjas.sh run
```