TLA+ Project 2024

Lorenzo Migliari

June 14, 2024

1 Overview

This specification defines the interactions between a **CPU (Data Broker)** and **GPU kernels** within a simplified system where the **CPU** provides data to the kernels. The primary components include the **CPU (Data Broker)**, **GPU Kernels (KGpus)**, **streamlets**, **logical memory segments**, and a **push channel**[1]. To better understand the specification, we will first analyze the actions taken by each component and then examine a simplified general case involving just one kernel.

2 GPU Kernels

GPU Kernels are the core of this specification. A **GPU kernel** is initialized with a random number of missing data values, which is the data it needs to complete its computation. This specification's objective is to fulfill each **GPU kernel**'s request.

The kernels can communicate with the **CPU** through a *push channel*, where they can push their data requests. In the specification, the *push channel* is a one-way message channel from **GPU kernels** to the **CPU**. The **CPU** will process these requests in order.

Another important component to understanding **GPU kernel behavior** is the **logical segments**. Each **GPU kernel** has its own logical segment. Each logical segment represents a portion of the **Application Cache** that is consumed by a single **GPU kernel** and filled by the **StreamLet**.

DATA	NULL	NULL

Figure 1: Example of a Logical Segment with only one data value.

[1] For a more detailed component description, refer to the TLA specification.

2.1 Behavior

We can summarize the **GPU Kernels'** behavior as follows:

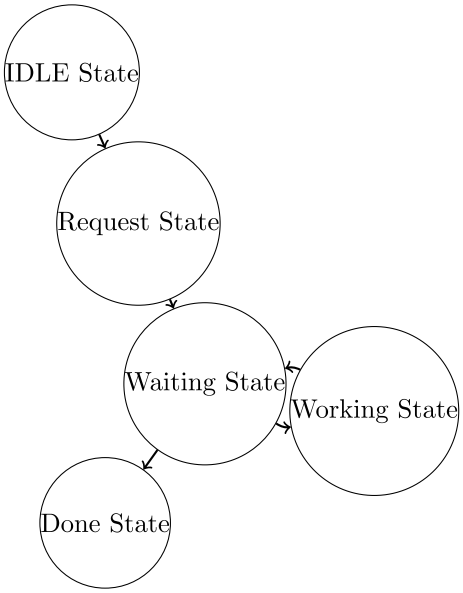

Figure 2: State Transition of the GPU Kernels.

1. All kernels start in the *IDLE State*.
2. If a kernel requires some data (data_needed > 0), it sends a request to the **CPU** through the **Push Channel** and transitions to the *Request State*.
3. When the **CPU** processes the request, it sends an *ACK* to the kernel, which then transitions to the *Waiting State*.
4. In the *Waiting State*, the kernel monitors its logical segment. When data arrives in the logical segment (as shown in Figure 1), it transitions to the *Working State* and stores the data in its own memory.
5. The kernel transitions between the *Working State* and the *Waiting State* until it has obtained all necessary data.
6. Once all data is received, the kernel transitions to the *Done State* and remains there.

3 CPU - Data Broker

The **CPU** component is responsible for managing the requests from the kernels. It finds the location of the data needed inside the physical memory and then provides the offset of that data to the **streamlets**. The **streamlets** will then stream the data into the addresses within the logical segment of the kernel that requires the data.

3.1 Behavior

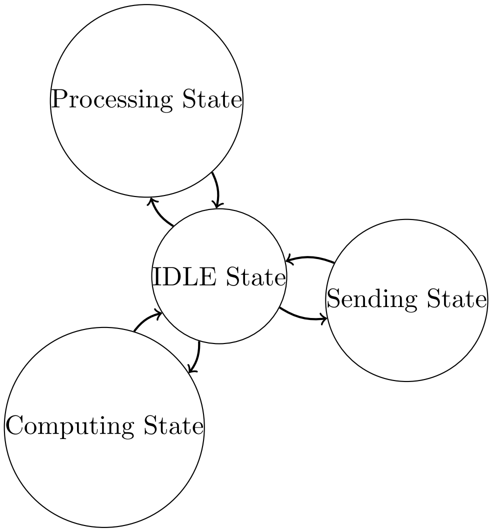

Figure 3: State Transition of the CPU (Data Broker).

We can summarize the **CPU** behavior as follows:

- The **CPU** starts in the *IDLE State*. After each state change, it returns to the *IDLE State* to decide which state to transition to next.
- If the **push channel** is not empty (i.e., a **GPU kernel** sent a request), it transitions to the *Processing State*, where it pops the request from the **push channel** and stores it in its own subscription list, which is a list of pending data requests not yet completed.
- If the subscription list is not empty, it transitions to the *Computing State*. In this state, the **CPU** computes the location in the physical memory of the data requested by the kernel.
- If a memory address is computed, the **CPU** transitions to the *Sending State*, where it initializes a **streamlet** to provide all the necessary data to the kernel to fulfill the request. To initialize a **streamlet**, it sets the offset of the data required by specifying the begin address and the end address of the data. The **streamlet** then provides the data to the logical segment of the kernel.

4 System Behavior

The graph below shows a simplified behavior of the spec for just one **GPU** that requests one data value.

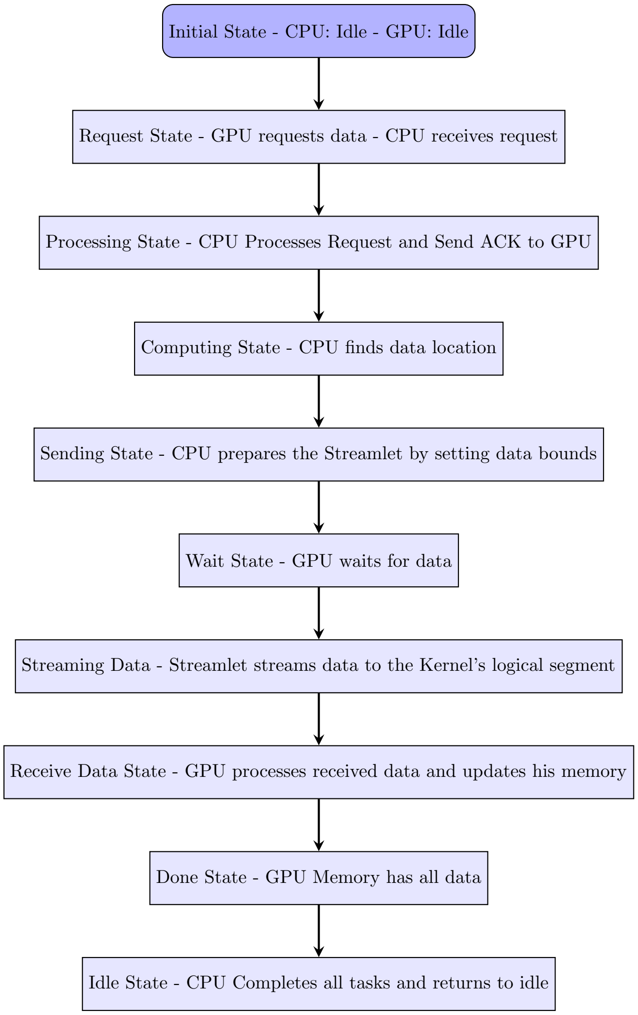
Initial State - CPU: Idle - GPU: Idle
Request State - GPU requests data - CPU receives request
Processing State - CPU Processes Request and Send ACK to GPU
Computing State - CPU finds data location
Sending State - CPU prepares the Streamlet by setting data bounds
Wait State - GPU waits for data
Streaming Data - Streamlet streams data to the Kernel's logical segment
Receive Data State - GPU processes received data and updates his memory
Done State - GPU Memory has all data
Idle State - CPU Completes all tasks and returns to idle

Figure 4: State Transition of the entire system.