

# Anatomy of a Validator

## Pipelining

The validators make extensive use of an optimization common in CPU design, called `pipelining`. Pipelining is the right tool for the job when there's a stream of input data that needs to be processed by a sequence of steps, and there's different hardware responsible for each. The quintessential example is using a washer and dryer to wash/dry/fold several loads of laundry. Washing must occur before drying and drying before folding, but each of the three operations is performed by a separate unit. To maximize efficiency, one creates a pipeline of stages. We'll call the washer one stage, the dryer another, and the folding process a third. To run the pipeline, one adds a second load of laundry to the washer just after the first load is added to the dryer. Likewise, the third load is added to the washer after the second is in the dryer and the first is being folded. In this way, one can make progress on three loads of laundry simultaneously. Given infinite loads, the pipeline will consistently complete a load at the rate of the slowest stage in the pipeline.

## Pipelining in the Validator

The validator contains two pipelined processes, one used in leader mode called the TPU and one used in validator mode called the TVU. In both cases, the hardware being pipelined is the same, the network input, the GPU cards, the CPU cores, writes to disk, and the network output. What it does with that hardware is different. The TPU exists to create ledger entries whereas the TVU exists to validate them. [Previous Native ZK Token Proof Program](#)[Next Validator's Transaction Processing Unit \(TPU\)](#)