

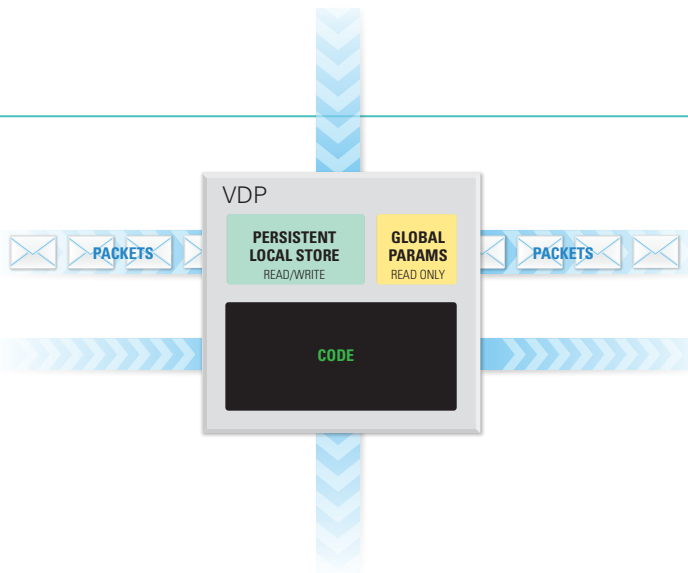
PULSAR

PULSAR (Parallel Ultra Light Systolic Arrays Runtime) offers a clear programming model, based on sound dataflow principles, and expressed through a compact Application Programming Interface (API). PULSAR automates communication and data-driven runtime scheduling. PULSAR's promise is to deliver unprecedented scalability on existing Peta-scale systems and upcoming Exa-scale systems.

BENEFITS OF PULSAR

- ▶ Simple Programming Model
- ▶ Lightweight Runtime System
- ▶ Powerful Virtualization
- ▶ Lightweight Implementation
- ▶ Seamless Multithreading and Message-passing
- ▶ Data-driven Dynamic Runtime Scheduling

PROGRAMMING MODEL



PULSAR offers a straightforward programming model, based on a few simple abstractions. The main abstraction in PULSAR is the Virtual Systolic Array (VSA), which is a connection of Virtual Data Processors (VDPs), identified by tuples, connected with channels and communicating with data packets.

The VDP is the main building block of the VSA, a direct descendant of the Processing Element (PE) in the traditional systolic array nomenclature. The VDP contains executable code, read-only global parameters, read/write, persistent, local store, a set of input channels and a set of output channels. Each VDP is uniquely identified by a tuple (a string of integers), and has a counter defining its life span.

RUNTIME SYSTEM

The VSA is executed by a collection of processes, each process containing a number of worker threads and a thread dedicated to handling inter-node communication, referred to here as the proxy. Usually, one process is mapped to one distributed memory node, one thread is mapped to one core, and the proxy runs on a dedicated core. Other mappings are possible, such as placing one process in each socket of a multi-socket system, or launching more threads than cores (oversubscribing).

