755 Phoenix Drive, Ann Arbor MI, 48108, USA
Phone: 734 668-2930 • Fax: 734 668-2877 • Email: info@carsim.com carsim.com

High Performance Computing (HPC) and VehicleSim

This document describes how VehicleSim products can be used within a high-performance computing environment and the requirements for such.

Definition of High-Performance Computing (HPC)

HPC generally refers to one of two critical items:

1. A hardware system in which multiple computers are connected through a software layer to provide a single, virtualized, computing platform. This platform, sometimes referred to as a cluster, draws upon the resources of the underlying hardware. The software layer is used to allocate and distribute these resources to executing applications.
2. The design, configuration, and deployment of software written to specifically target an existing HPC cluster and its underlying hardware.

This document is predominantly concerned with the second, software centered, component.

This document will briefly cover many topics regarding HPC computing. A basic understanding of the concepts is assumed. Additional familiarity with the common terms and use of a VehicleSim product (CarSim, TruckSim, BikeSim, or SuspensionSim) is also assumed. Definitions and explanations of the referenced VehicleSim features can be found elsewhere in documents bundled within the related VehicleSim product installation.

Hardware and Software Requirements

There are several requirements when deploying VehicleSim to an HPC environment.

The implementation of these requirements, generally called “a stack”, may impose important restrictions or considerations when designing an HPC application.

HPC Cluster Hardware

This is the computer hardware that sits at the very bottom of an HPC stack and can potentially determine much of the subsequent options when building a HPC environment.

There are two commonly used options available to meet this requirement:

1. Buy and manage it internally
2. Lease it from someone else

Buying and managing it internally generally has a larger upfront cost compared to leasing. Over time, however, the savings can be massive if the HPC workload is large enough. Hardware leasing is generally abstracted, and this abstraction enables the option to avoid the costs of the Virtualized Hardware Provisioner.

Designing a functional HPC hardware cluster to be hosted and managed internally can be a complex task. Design and implementation recommendations of this hardware is outside the scope of this document.

Virtualized Hardware Provisioner

The Virtualized Hardware Provisioner is the stack component that takes the HPC cluster's aggregated computing resources and distributes them to the application runtime environments.

If the HPC hardware is leased the Virtualized Hardware Provisioner component is already provided. Companies like Amazon AWS, Google Cloud Platform, Microsoft Azure, etc., can provide this component as part of their high-performance computing offering

If the HPC Cluster is internally hosted, a hardware virtualization platform like VMWare VSphere or Hyper-V will be needed to allocate physical computational resources. Alternatively, though very rarely done, allocation of the entire cluster to a single application deployment is possible. In this case the virtualized hardware provisioner is not needed.

Application Deployment Orchestrator

Once there is a virtual cluster, or a full internally hosted cluster, available through a virtual hardware provisioner, managing application deployments is the next requirement.

Managing those applications is the responsibility of an Application Deployment Orchestrator. It manages not only applications intended to be executed on the HPC system - but also data persistence, availability, load balancing, external access, and communication between components within the application.

The application deployment orchestrator can take the form of a Kubernetes installation, an abstracted service like AWS Fargate, or a local container orchestrator like Docker Swarm or Docker Compose.

Application Runtime

When the orchestrator performs an application deployment, it will need a runtime to execute an application. The choice of orchestration, and application availability needs, may impose restrictions on the available options.

These virtualized runtime options include but are not limited to containerized runtimes like Docker or Containerd, or even VMs running Linux or Windows.

Workload Manager

Finally, before an application can be designed, it will need a workload management solution. This component is what maintains the workload within the cluster, connects to application instances, and distributes work amongst them. This component is necessary to ensure proper completion of all units of work, and successful allocation of results. It is responsible for managing excess work that needs to be queued for consumption by the application instances.

Options here include AMQP message brokers like RabbitMQ or IBM MQ, or data streaming applications like Apache Kafka.

HPC Application Development

Licensing

Mechanical Simulation has a HPC Licensing option which can be purchased that specifically meets the needs of an HPC deployment. The HPC license enables the use of a different operating mode of the license server. This mode provides efficient and rapid license check operations which enable the license server to keep up with the far faster request rate for licenses by HPC applications. The HPC license effectively removes the overhead of license management

VS SDK

When writing VehicleSim based applications, the Mechanical Simulation VS SDK is the best place to start. It provides a set of tools, libraries, code examples and documentation to assist development of said applications.

VS API

VS API is a set of functions and utilities for interacting directly with the VS Solver. It is generally used for writing wrapper applications that can modify or extend the solver in a variety of ways. A short list of some of its features include:

- Adding equations at runtime, through the employment of VS Commands.
- Exchange of internal and external variables into the simulation.
- Direct access of variables within the VS Math Model. Operations performed on these variables within the wrapper program can then be integrated with the calculations performed within the VS Math Model.
- Saving and restoring VS Math Model state.
- Applying callbacks to the VS Math Model.

VS Connect API

This is a library which provides a series of functions that assist in communicating between VehicleSim and external software. It can be used to interact with software like MATLAB or Simulink. easier.

VS Output and Table APIs

These two APIs are used to interact with outputs from the VS Solver. Either output channels or output tables. Using this, you can prune results for DoE-like applications that are generally deployed on HPC systems.

VS Vehicle Module

This library is a high-level wrapper for the VS API, which makes creating runs significantly easier. It is especially useful when creating simulations that have multiple vehicle instances (using either one or more solver) in the same environment.

VS Terrain Library

This library is a standalone DLL that can generate VS Terrain files for use in simulations. The geometry generated by this library can be used to attach friction and rolling resistance to each face of the resulting terrain.