VortexRNG

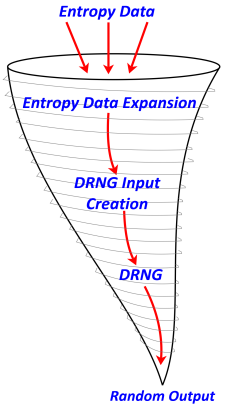
VortexRNG Server ™

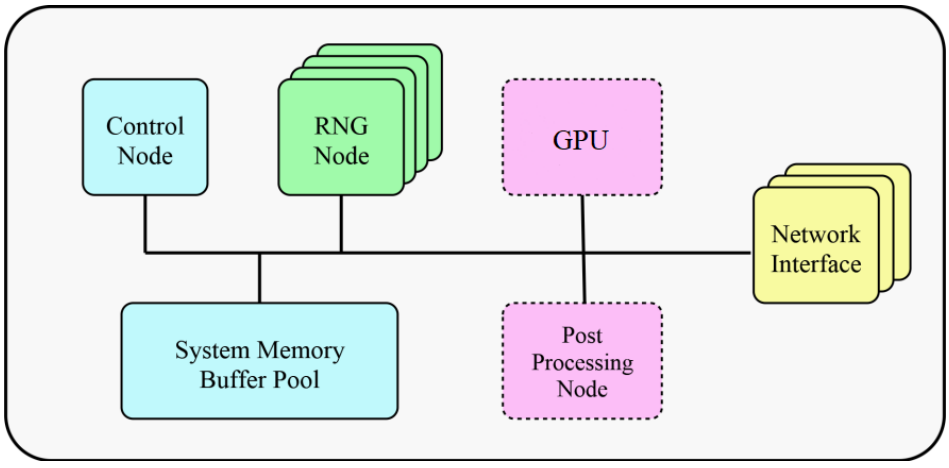
The VortexRNG Server is an innovative software package for researchers and scientists who require a large volume of high quality, deterministic random numbers in their simulation work. The VortexRNG Server off-loads random number generation from the compute resources running simulations for faster simulations and quicker results. The VortexRNG Server turns a 64-bit X86 based Linux computer into a high-performance, high quality random number generator that can be shared between many users.

The VortexRNG Server utilizes multiple CPU cores as RNG nodes as shown in the block diagram below. As well as being highly scalable (it can run on computers from laptop to supercomputer), the VortexRNG Server is also highly configurable. RNG Nodes can be reserved for high priority requests or grouped to act as a single RNG for parallel generation, increasing the generation rates for a single client. When run on a dedicated system, non-RNG functions can be assigned to dedicated cores to help maximize throughput. When run on a multi-purpose system where resources must be shared, these functions can run on the Control Node to free up resources for other applications.

Two RNG algorithms are supported by the VortexRNG Server, the commonly used Mersenne Twister and the proprietary VortexRNG algorithm. The Mersenne Twister algorithm is widely used in simulations and incorporating it in the VortexRNG Server provides compatibility with previous simulation work and with others that don't use the VortexRNG Server.

The unique architecture of the VortexRNG algorithm makes the data flow through the algorithm similar to a swirling vortex. Synthesized entropy data is passed into the algorithm at a relatively slow rate where it is expanded into a larger set of entropy data. Multiple expanded entropy data values are combined to feed into the DRNG. The DRNG is repeatedly cycled to produce the random number output sequence. Each stage of the data flow is cycled faster and more times than the previous stage, resulting in dozens of random numbers being produced for each set of input entropy data values. The amount of state data makes the probability of generating a repeating pattern on the order of one in 10^{300} numbers, thus effectively true random number generation comparable to a hardware true RNG but with deterministic operation. The VortexRNG algorithm was designed with acceleration in mind so it runs very efficiently on NVIDIA GPUs.





VortexRNG

VortexRNG Server

- Offloads random number generation from primary compute resources so simulations run faster.
- Multi-threaded architecture takes advantage of multi-core processors and multiple processors in a system.
- A C source code driver is provided for client use and easy implementation on a variety of host microprocessors. Only slight modifications are required to work with Fortran applications. The server side software will work with any client side operating system, programming language or processor architecture.
- Interacts with client through “streams”, each stream is a virtual, independent DRNG.
- Supports buffered pre-generation to minimize latency when the client requests a block of numbers.
- Multiple options for increasing number generation rates (without compromising the quality of generated numbers) such as grouping streams for parallel generation.
- Supports multiple priority levels on a per-stream basis so lower priority development work doesn’t impact generation rates for high priority “production” simulation runs.
- Supports multiple NICs and any TCP/IP based network or IPoIB. Other high performance interfaces and protocols such as RDMA will be supported in future releases.
- Post processing options for floating point numberss and specific distributions.
- Expected to run on any Linux distribution (tested on Ubuntu and CentOS, others to follow).
- Secure operation ensured by users only being able to interact with streams they create.
- VortexRNG Server Manager provides load balancing and license management for multiple VortexRNG Server instances running on separate computers.
- Logging and administrative options facilitate usage monitoring and billing for usage. Future versions will support per-user permissions for using high-performance options.
- Multiple product offerings suitable for running on a wide range of hardware from a desktop computer for individual or small departmental use to supercomputer class systems with hundreds to thousands of cores.

RNG Features

- Easy selection option to use the Mersenne Twister DRNG or the VortexRNG DRNG.
- Supports the Mersenne Twister DRNG for compatibility with previous simulation work and with others not using the VortexRNG Server. GPU acceleration is not available for the Mersenne Twister.
- The VortexRNG DRNG is a very fast and strong proprietary DRNG based on multiple enhanced linear-feedback shift registers. Generation rates in excess of 20 million numbers per-second per-core can be achieved with commodity-level 64-bit X86 processors.
- Performance and the quality of numbers generated with the VortexRNG DRNG are comparable to commonly used DRNGs for simulations such as Mersenne Twister, the GSL Tausworthe RNG and others.
- Comparable results on common tests of randomness as for commonly used DRNGs. This includes NIST-22 and FIPS 140 tests for cryptographically secure applications.
- Easy client selection for deterministic or random mode with similar performance and number quality.
- Seeding algorithm generates quality seeds from the user provided seed. This provides for high quality number generation from the very first numbers generated after seeding.

GPU Acceleration

- The VortexRNG DRNG can take advantage of an NVIDIA GPU for significantly higher performance (over one billion numbers per second with a mid-range graphics oriented GPU)
- Core section of the DRNG algorithm runs on parallel GPU threads and optionally on parallel CUDA blocks to achieve even higher generation rates.
- Supports NVIDIA architectures starting with Pascal (compute capabilities 6.0 or higher). Even the older architectures provide a significant performance boost compared to running on a X86 processor.
- Configurable usage of GPU resources allows for optimizing for maximum performance or for multiple users to efficiently share an expensive resource.
- Supports multiple GPUs and a mix of GPU compute capabilities, each individually configurable.
- Uses a GPU “white list” to prevent specific GPUs in the system from being used for number generation.