ArcHPC

PRODUCT BRIEF

Optimizing Your GPU Infrastructure

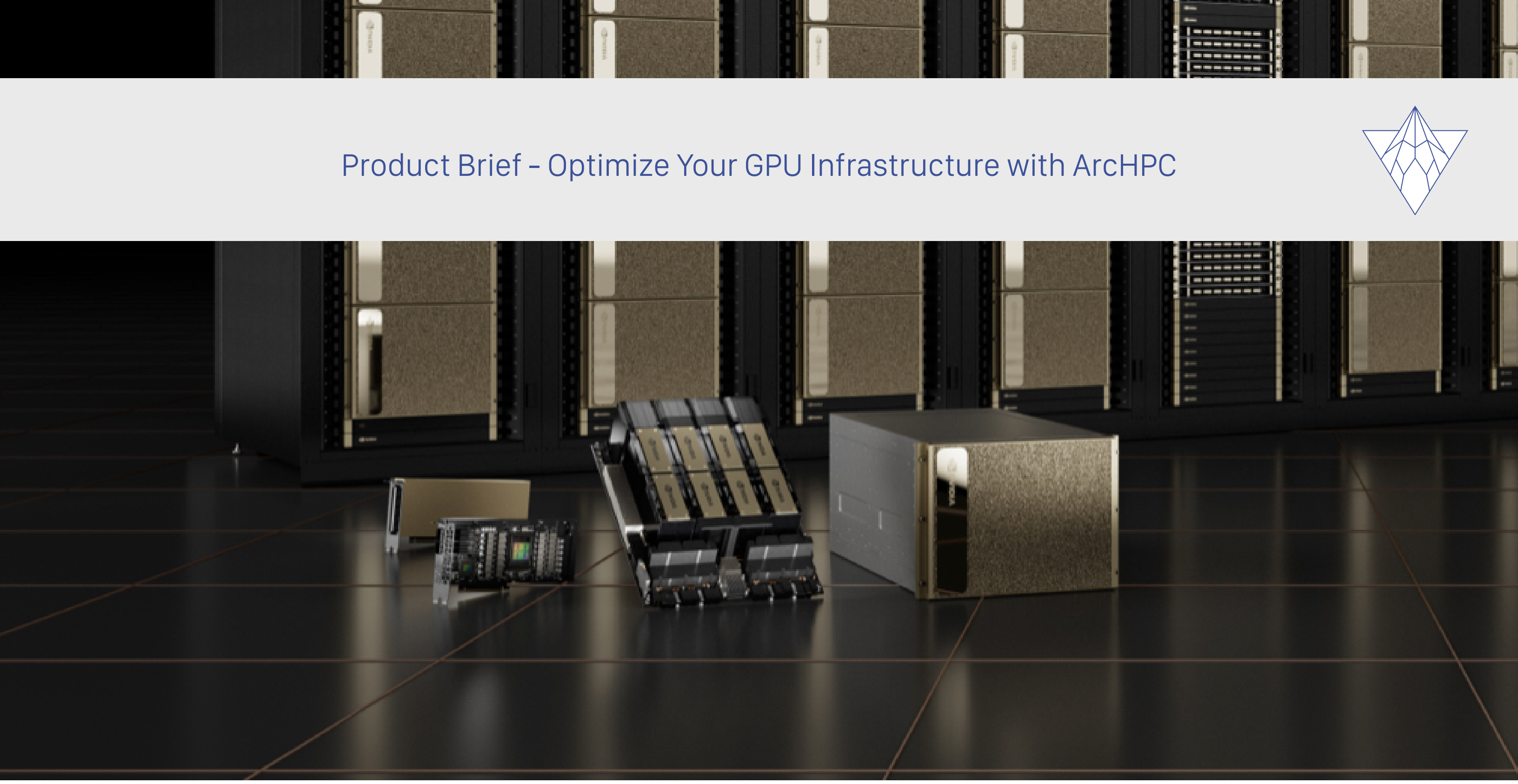
ARC COMPUTE

Table of Contents

About Arc Compute

Arc Compute is a research and development company building optimization software for accelerated hardware, revolutionizing the HPC space. We aim to fully optimize the performance and utilization of the most prominent accelerated hardware technologies. Our impact on the world will involve reducing carbon footprints for organizations and individuals looking to use accelerated hardware-powered solutions in their day-to-day operations by maximizing the capabilities of the underlying hardware infrastructure. Our research and development have increased performance by enabling the complete utilization of computing resources. We expect to expand on this research as we continue mapping the capabilities of accelerated hardware at low-level utilization and optimization points. Arc Compute specializes in developing software solutions that maximize the throughput of accelerated hardware through mapping low-level utilization/optimization points, scheduling, and maximizing task compute and input/output thresholds.

ARC COMPUTE



Overview of ArcHPC Suite

ArcHPC is an advanced software suite designed to improve efficiency in task execution within computing environments. It does this by strategically scheduling additional arithmetic operations during periods of memory operations. This innovative approach results in an increased density of tasks and enhanced performance on accelerated hardware, effectively addressing the limitations and barriers inherent in traditional GPU hypervisors and management solutions. ArcHPC allows for the interlacing of complementary tasks, by increasing user/task density executing concurrently, that have distinct resource needs in each GPU clock cycle. This capability ensures the shared use of GPU resources without any loss in performance. Additionally, by optimally managing low-level compute resources, ArcHPC is able to achieve 100% utilization. This leads to a notable increase in performance, with boosts ranging from 39% to 308%, all while utilizing only half of the resources normally required.

How We Do It (1/2)

1. Maintain processor up time by memory-level parallelism

Maintaining processor uptime through memory-level parallelism is a critical consideration in optimizing overall system performance. By efficiently overlapping or interleaving memory access operations with computational tasks, memory-level parallelism helps minimize idle cycles and ensures that the processor is consistently engaged in productive work.

Memory type	CPI (cycles)
Global memory	**290**
L2 cache	**200**
L1 cache	**33**
Shared Memory (1d/st)	**(23/19)**

*Arithmetic operations take 1 to 11 cycles, which makes it possible to fit 19x to 200x more operations in that time frame increasing the GPUs performance.

2. Fine-tuning in the GPU task environment for minimum and maximum compute times at intersection points

This involves optimizing the execution of tasks by adjusting parameters to achieve the best possible performance within the given constraints. Careful tuning allows for the efficient allocation of resources, enhancing the overall efficiency of GPU-based computations.

How We Do It (2/2)

3. Maximizing the optimal thread arrangement

Maximizing the optimal arrangement of threads mitigating divergence. This entails organizing parallel SIMD threads in a manner that fully leverages the capabilities of the GPU’s Streaming Multiprocessors (SMs) and CUDA cores. Achieving the optimal thread arrangement is essential for parallel processing and can significantly impact the overall throughput, efficiency and minimizing latency of GPU computations.

4. Optimizing warp scheduling.

Maximizing optimal warp scheduling involves efficiently organizing and scheduling groups of threads, known as warps, to fully utilize the available computational resources. By optimizing warp scheduling, developers can enhance the parallelism and concurrency of GPU operations, leading to improved overall performance in various computational workloads.

Key Benefits of ArcHPC

Better Performance

ArcHPC increases performance reducing compute time by maintaining processor up time using memory-level parallelism, GPU task environment tuning and maximizing optimal thread arrangement and warp scheduling. These features
enable multiple compute-intensive and memory-intensive jobs, such as AI training and inference, to run simultaneously on the same GPUs while increasing workload performance and expediting completion times. ArcHPC's method of layering jobs is faster than legacy methods like bare metal and MIG.

Increased Utilization

ArcHPC increases utilization by enabling the saturation of pipelines by compute-compatible workloads. These workloads can be layered in the same architecture while reallocating idle and underutilized resources where needed during runtime and fine tuning task compute environments based on the exact resources they need at a given moment.

Reduced Hardware Requirements

ArcHPC's ability to speed up different types of workloads running on the same infrastructure simultaneously enables organizations to drastically reduce their infrastructure requirements and carbon footprint while getting the most out of their GPUs.

Ease-of-Use

ArcHPC is a simple-to-use application that integrates with other HPC tools and is easy to manage.

Impact of ArcHPC

Common Issues Addressed

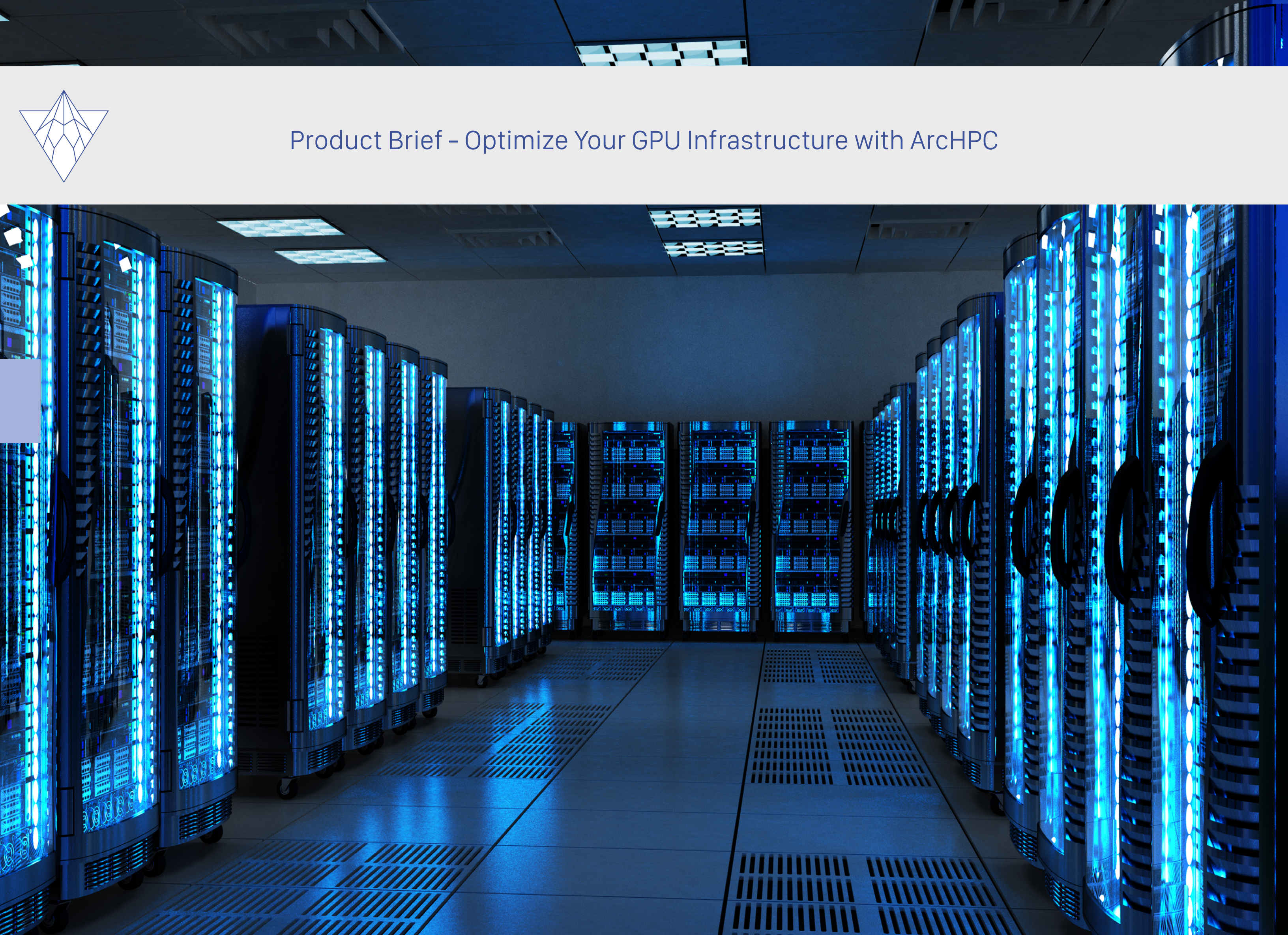
- Lower operational carbon footprint
- Code inefficiencies
- Performance increases, reduced time to market/expedited project completion
- Utilization/full value of investment realization
- Hardware-constrained growth

Relevant Industries

- Pharmaceuticals
- Oil and Gas/Energy
- Information Technology
- Aerospace
- Aviation
- Gaming and Entertainment

Applicable Workflows

- AI/ML
- Drug discovery
- Fluid dynamics
- GEOINT
- Autonomous applications/Autonomy
- Protein folding
- Scientific and engineering simulations
- Gaming-as-a-Service
- Data science
- Analytics
- Rendering



Product Summary

ArcHPC is a kernel-based application that sits just above bare metal in the HPC tool and software tech stack that integrates with job scheduler and orchestration tools. It directly manages and optimizes GPU SM cores and other computing resources to achieve 100% utilization business objectives and higher GPU performance targets. ArcHPC reduces computing times by mitigating infrastructure-use inefficiencies, inefficient/unoptimized code, and human administration task management inefficiencies.

Key Features

User Space Vs. Kernal Space Applications

User Space Applications - Job Schedulers (i.e. Slurm)

- Address data and code inputs
- Provides an environment for queuing inputs
- Determines the inputs deployed to **Kernel Space Applications**
- Can only manage when/where inputs are deployed.

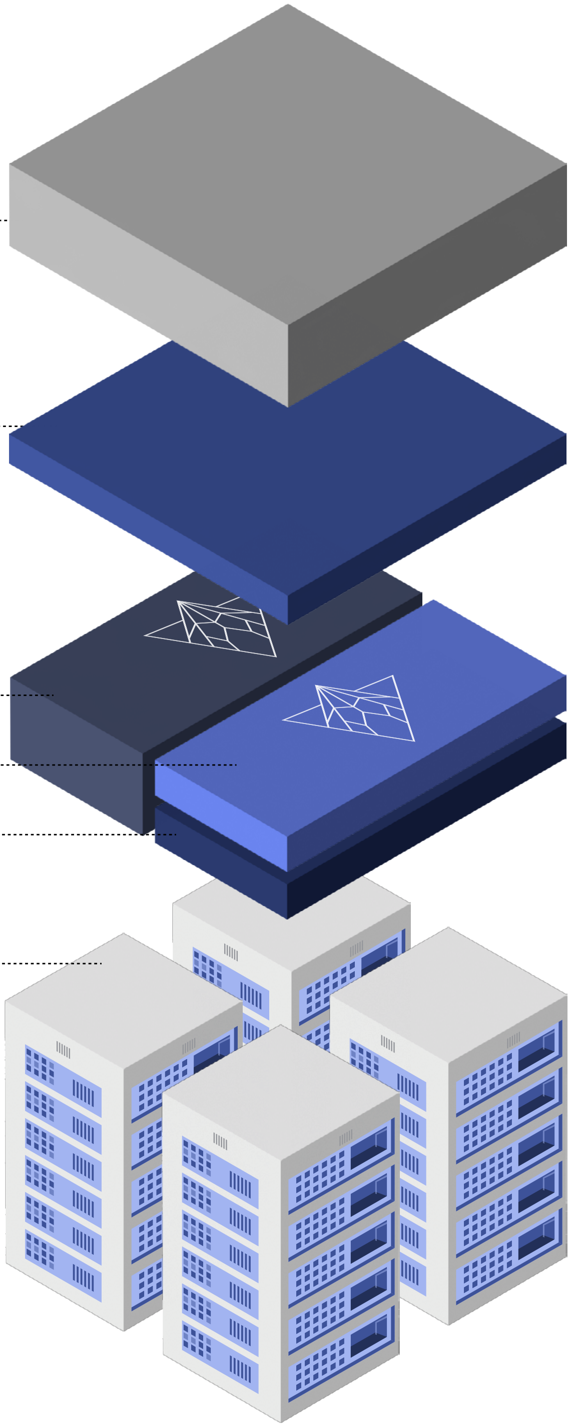
Kernel Space Applications - Virtualizers (ArcHPC)

- Addresses how the hardware operates.
- Ability to manage where, when, and how resources apply to the inputs of **User Space Applications**

Key Features List

- Easy to use browser-based interface.
- Supports NVIDIA, Intel, and AMD GPUs
- Kernal Space Application
- Unlimited partitioning capabilities
- Cluster management using job scheduler (i.e. Slurm)
- Organizational-Level Provisioning
- Shared Roles

Data Center Tech Stack



ArcHPC sits within the management layer of data center nodes offering lower level GPU/CPU optimizations than any job scheduler can.

Questions about ArcHPC?

Contact us!

✉ archpc@arccompute.io

🌐 www.arccompute.io