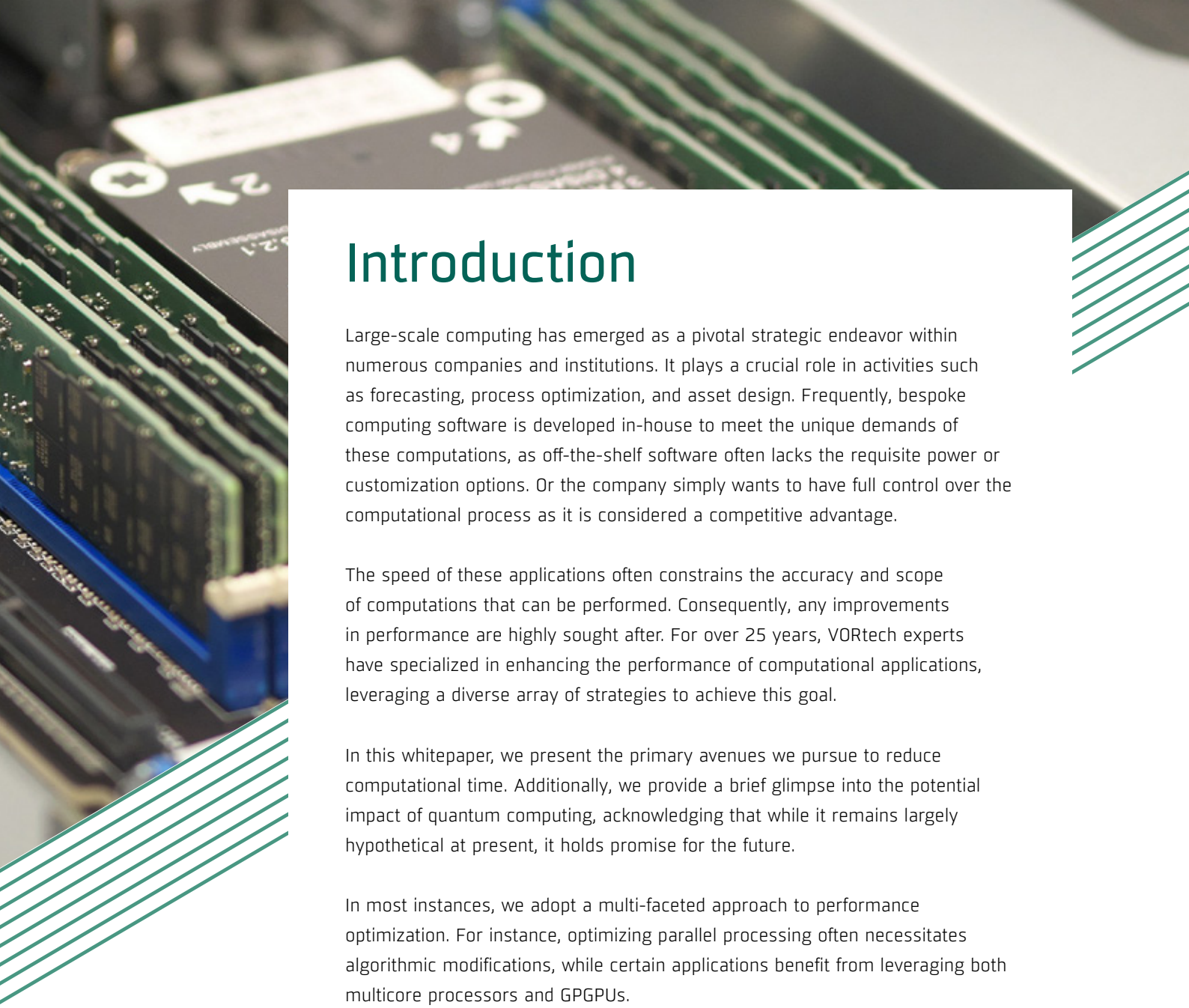
VORTECH

WHITEPAPER

Eight ways to speed up your computing software.

Compute more in less time.

Content



Introduction

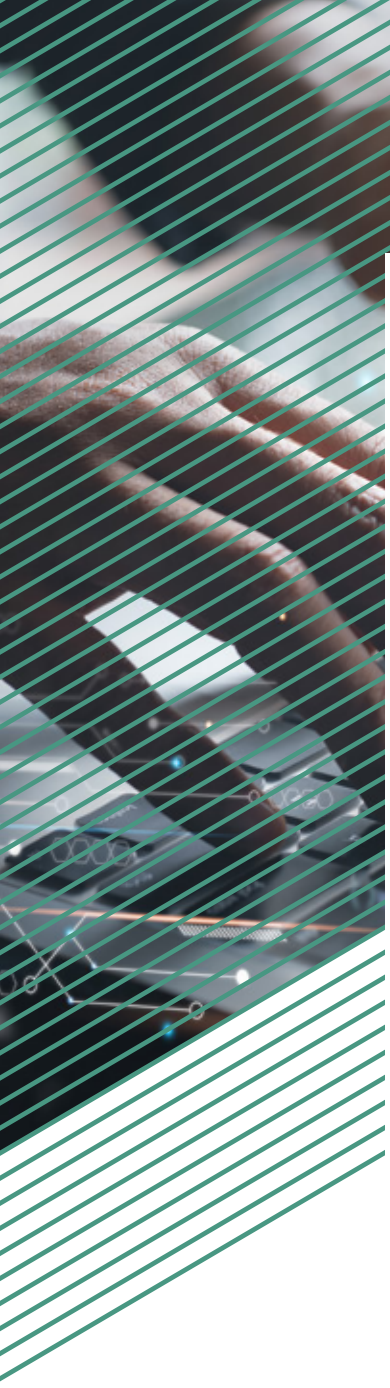
Large-scale computing has emerged as a pivotal strategic endeavor within numerous companies and institutions. It plays a crucial role in activities such as forecasting, process optimization, and asset design. Frequently, bespoke computing software is developed in-house to meet the unique demands of these computations, as off-the-shelf software often lacks the requisite power or customization options. Or the company simply wants to have full control over the computational process as it is considered a competitive advantage.

The speed of these applications often constrains the accuracy and scope of computations that can be performed. Consequently, any improvements in performance are highly sought after. For over 25 years, VORtech experts have specialized in enhancing the performance of computational applications, leveraging a diverse array of strategies to achieve this goal.

In this whitepaper, we present the primary avenues we pursue to reduce computational time. Additionally, we provide a brief glimpse into the potential impact of quantum computing, acknowledging that while it remains largely hypothetical at present, it holds promise for the future.

In most instances, we adopt a multi-faceted approach to performance optimization. For instance, optimizing parallel processing often necessitates algorithmic modifications, while certain applications benefit from leveraging both multicore processors and GPGPUs.

If you seek guidance on the most effective path forward for your application, we invite you to engage us for a model scan. This comprehensive assessment will furnish you with actionable insights to guide your developers in optimizing the application effectively.



1. Performance-aware programming

How does it work?

Naïve programming practices can significantly impair performance, often resulting in costly inefficiencies. However, implementing even a modicum of programming discipline can yield notable performance enhancements. Strategies such as minimizing function calls, eliminating unnecessary memory allocations, reordering loops, and judiciously deciding whether to reuse or recalculate results are well-established techniques for mitigating computing time.

Potential speedup

The speedup obviously depends on how good the original programming already was, but a speedup by a factor of 2 or more is not unusual if the original program was not yet optimized.

Advantages

- Any developer with an understanding of program execution can implement these optimizations.
- Typically, the code modifications required are minor, often resulting in improved code quality.
- Adhering to programming guidelines can foster performance-aware coding practices.
- Performance analyzers are readily accessible tools for identifying and resolving performance issues.

Disadvantages

- It necessitates developers who grasp the performance implications of their decisions and understand both the compiler and the hardware.
- Striving for maximum hardware performance can present a challenge in balancing performance optimization with code readability.

2. Use parallelism in standard hardware

How does it work?

The processing unit in most computers comprises multiple processors, each equipped with multiple computing cores, facilitating parallel execution of application tasks. Core counts per processor typically range from 4 to 8, though top-tier models can boast up to 128 cores. Systems typically feature 2 processors.

Potential speedup

The maximum speedup achievable is heavily influenced by both the total number of cores in the hardware and the characteristics of the application. Generally, application performance shows a linear increase with the addition of more cores. However, beyond a certain threshold, parallelization becomes less efficient, resulting in diminishing returns. For instance, while an application may experience a 3x speedup with 4 cores, the speedup may only reach 7x when utilizing 16 cores.

Advantages

- No specialized hardware is required; it optimizes the use of standard available hardware.
- Substantial performance improvements can be achieved through straightforward code adjustments that are easily comprehensible.
- Extensive software restructuring is frequently unnecessary.
- Cloud providers commonly furnish access to cutting-edge multicore processors, granting easy entry to high-performance computing capabilities.

Disadvantages

- Performance enhancements are constrained by the CPU resources of a single server.
- Achieving optimal hardware utilization may require substantial code refactoring. The goal is to maximize application parallelism while mitigating potential bottlenecks that arise when multiple processes access memory via the same bus.

3. Use GPGPUs

How does it work?

General Purpose Graphical Processing Units (GPGPUs) feature numerous small computing cores operating synchronously to execute operations on multiple data items concurrently. If your application involves repetitive computations across a vast dataset, leveraging a GPGPU can yield substantial performance gains. High-end GPGPUs boast several thousand computing cores.

Potential speedup

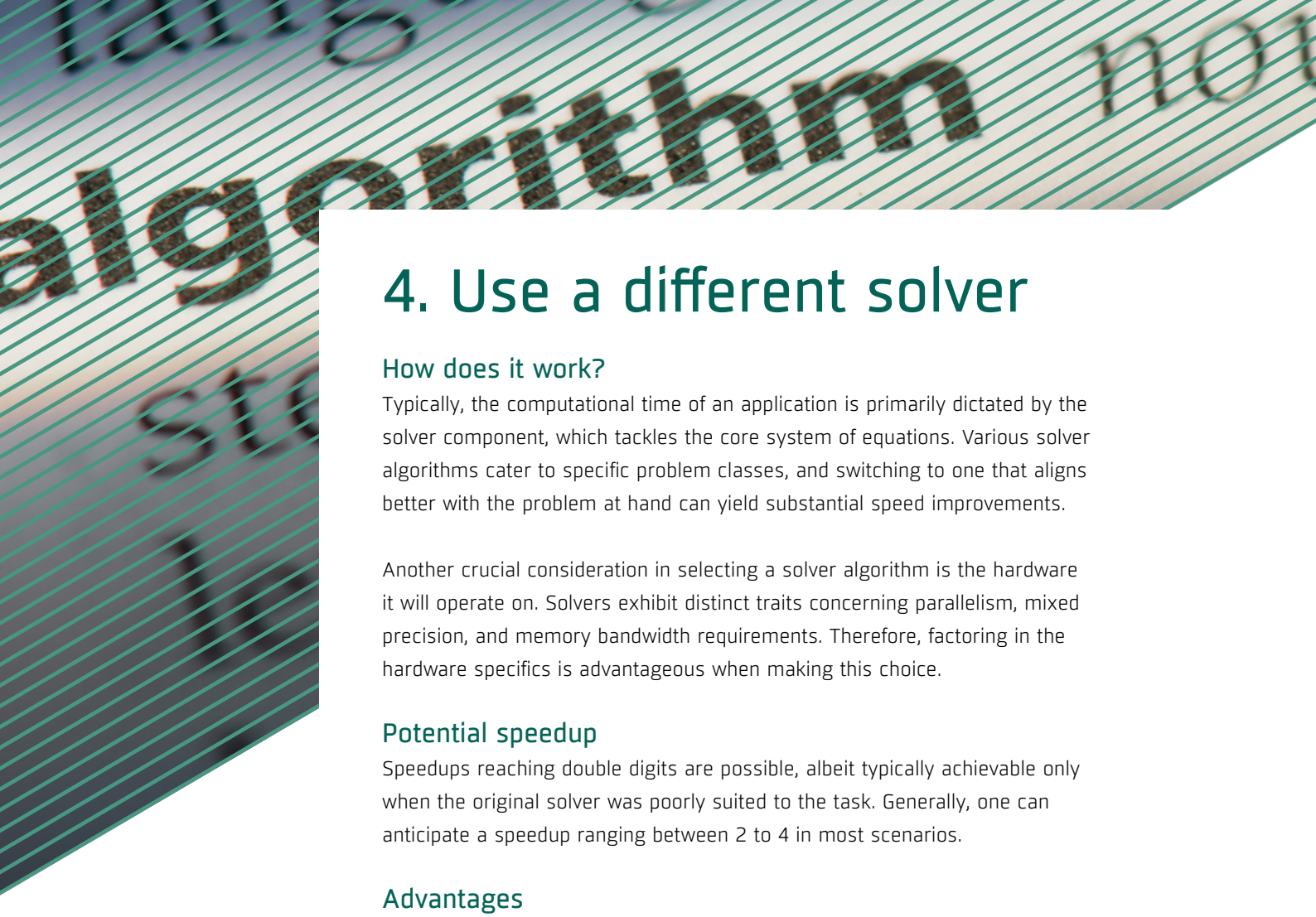
The speedup compared to CPU computation can go up to 10x, although this largely hinges on the level of parallelism within the application. Typically, the speedup falls within the range of about 4x.

Advantages

- Offers the potential for significant speedup, albeit primarily for specific applications.
- Energy-efficient: GPGPUs consume less energy per computation compared to standard processors.
- Further performance and energy efficiency gains are achievable using lower precision arithmetic.
- GPGPUs are readily accessible and straightforward to integrate into your system.
- GPGPUs are accessible on public cloud platforms.

Disadvantages

- Speedup potential is restricted in most applications.
- Leveraging GPGPUs typically demands substantial code alterations and introduces constructs less familiar to many developers.
- Significant performance gains in certain applications are only realized through porting most of the application completely to the GPU, primarily due to the relatively slow data transfer between CPU and GPU.



4. Use a different solver

How does it work?

Typically, the computational time of an application is primarily dictated by the solver component, which tackles the core system of equations. Various solver algorithms cater to specific problem classes, and switching to one that aligns better with the problem at hand can yield substantial speed improvements.

Another crucial consideration in selecting a solver algorithm is the hardware it will operate on. Solvers exhibit distinct traits concerning parallelism, mixed precision, and memory bandwidth requirements. Therefore, factoring in the hardware specifics is advantageous when making this choice.

Potential speedup

Speedups reaching double digits are possible, albeit typically achievable only when the original solver was poorly suited to the task. Generally, one can anticipate a speedup ranging between 2 to 4 in most scenarios.

Advantages

- Decreases in the total number of computations result in a more energy-efficient application.
- Depending on the application's structure, merely invoking a different solver routine may suffice, thereby necessitating only minor code modifications.
- In certain scenarios, adopting a more suitable solver can also enhance robustness and accuracy.

Disadvantages

- Implementing an improved solver demands specialized expertise and occasionally entails research efforts.
- Substantial refactoring may be essential depending on the quality and architecture of the existing code.

5. Reducing model complexity

How does it work?

Numerous computing codes rely on a mathematical model that is then discretized employing techniques like finite elements or finite differences. Achieving high accuracy typically demands a large number of discretization points. However, high precision isn't always necessary. In such instances, a reduced-order model with fewer degrees of freedom can be developed. Rigorous mathematical techniques guarantee that this reduced model closely approximates the full model.

The reduced-order model finds application, for example, in optimization processes for getting close to the optimal point. Subsequently, the full model takes charge during the final optimization phase to attain the exact optimum.

Potential speedup

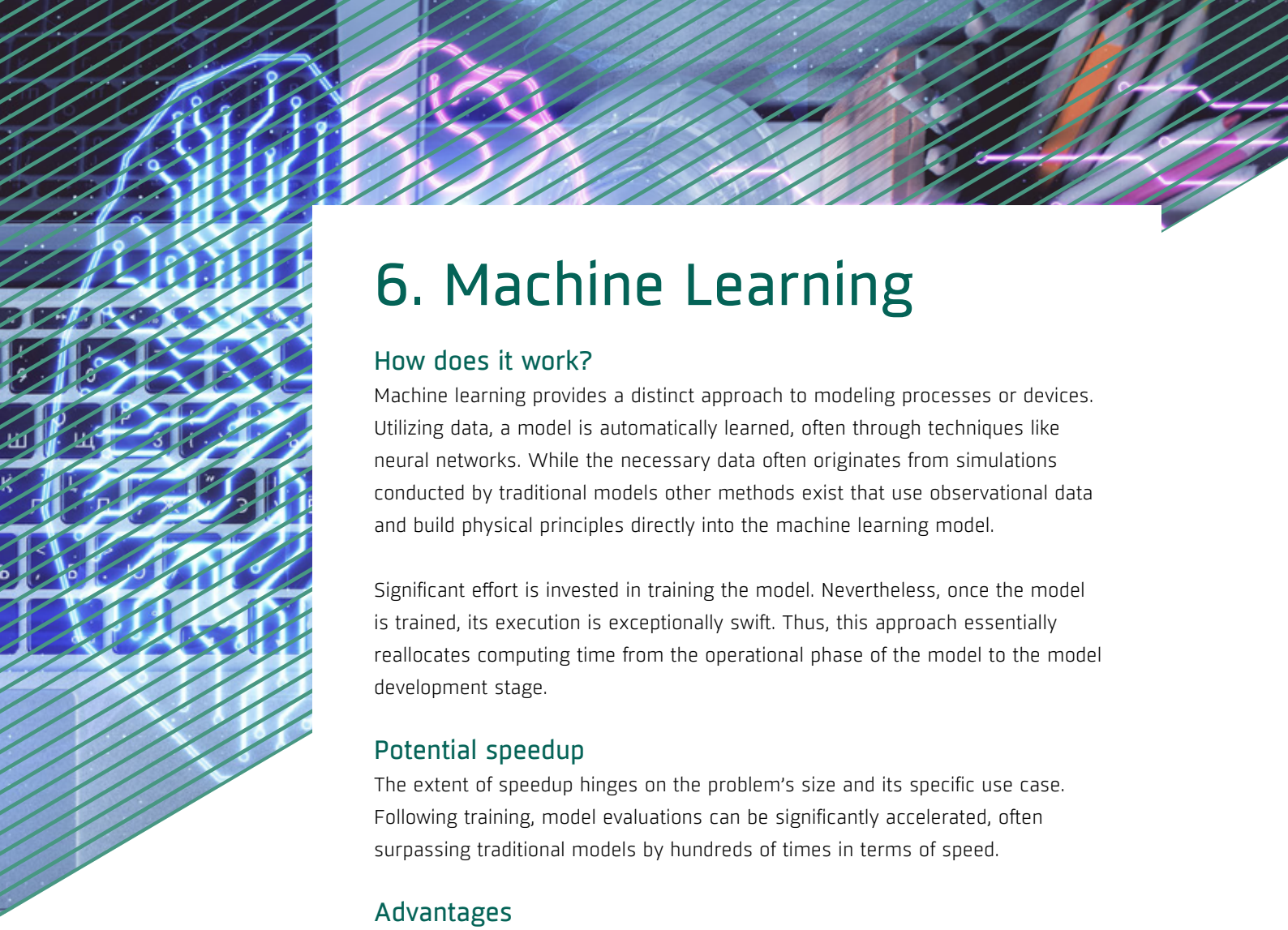
The speedup depends on the size of the problem. Typical speedup is somewhere between 4 and 20.

Advantages

- Reducing the number of computations not only enhances speed but also lowers energy consumption.
- The precision of the reduced-order model is adjustable, enabling informed decision-making regarding its accuracy.

Disadvantages

- Constructing a reduced-order model is frequently intricate. The simple methods that exist often entail numerous initial computations to gather essential information for the reduced model.
- The reduced model's decreased accuracy may not always meet acceptable standards.
- The need to manage both the original and reduced complexity models complicates software maintenance.



6. Machine Learning

How does it work?

Machine learning provides a distinct approach to modeling processes or devices. Utilizing data, a model is automatically learned, often through techniques like neural networks. While the necessary data often originates from simulations conducted by traditional models other methods exist that use observational data and build physical principles directly into the machine learning model.

Significant effort is invested in training the model. Nevertheless, once the model is trained, its execution is exceptionally swift. Thus, this approach essentially reallocates computing time from the operational phase of the model to the model development stage.

Potential speedup

The extent of speedup hinges on the problem's size and its specific use case. Following training, model evaluations can be significantly accelerated, often surpassing traditional models by hundreds of times in terms of speed.

Advantages

- The speedups during the use of the model can be spectacular.

Disadvantages

- Building a proper model necessitates extensive knowledge of machine learning.
- The reliability of the model may diminish, as verifying its correctness can prove challenging.
- The selection of training data warrants careful consideration, as it profoundly impacts the quality of the trained model.
- When using simulation data to train the model numerous computations of the original model are still needed.

7. High-Performance Computing

How does it work?

The term high-performance computing (HPC) denotes the utilization of systems equipped with extensive processor arrays. HPC systems typically feature dozens to thousands of processors, often incorporating GPU nodes. What sets them apart from conventional compute systems is the speed of their interconnect.

To run effectively on this kind of hardware, applications must consist of separate processes that communicate through explicit messages. Adapting an application not initially designed for HPC entails significant effort. However, when techniques like model order reduction or machine learning prove ineffective, simply using a lot of hardware is a very valid solution.

Potential speedup

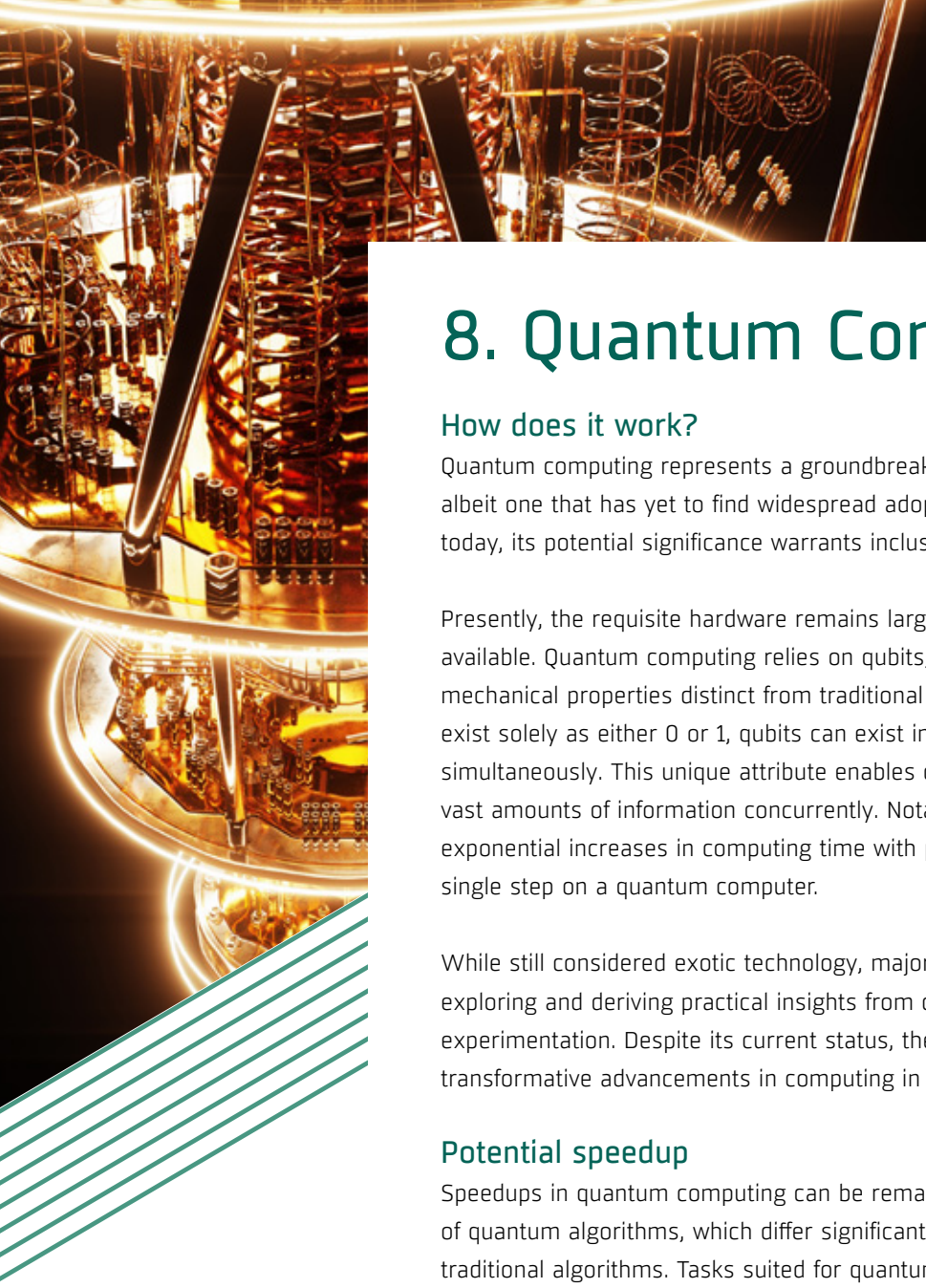
The maximum attainable speedup is inherently constrained by both the number of processors within the system and the degree of parallelism inherent in the application. A meticulously crafted application can effectively scale up to encompass hundreds of processors, achieving a speedup that constitutes a noteworthy fraction of the total number of processors employed.

Advantages

- Significant speedups are achievable.
- Large datasets can be effectively managed.
- Well-designed applications possess the flexibility to scale seamlessly from utilizing a few processors for small tasks to harnessing numerous processors for larger workloads.
- Access to high-performance computing systems is facilitated through HPC providers, eliminating the need for proprietary hardware in most cases.

Disadvantages

- Effective utilization of the hardware necessitates specially designed or adapted applications.
- HPC systems are typically shared among multiple users, potentially resulting in larger compute jobs needing to wait for an available timeslot.
- The utilization of high-performance computing systems often incurs significant expenses.
- Handling large volumes of data when using a remote system can be cumbersome due to the need for uploading and downloading.



8. Quantum Computing

How does it work?

Quantum computing represents a groundbreaking paradigm shift in computation, albeit one that has yet to find widespread adoption. Even if we are not using it today, its potential significance warrants inclusion in this overview.

Presently, the requisite hardware remains largely experimental and scarcely available. Quantum computing relies on qubits, which possess quantum mechanical properties distinct from traditional bits. Unlike classical bits, which exist solely as either 0 or 1, qubits can exist in a superposition of both states simultaneously. This unique attribute enables quantum computers to process vast amounts of information concurrently. Notably, algorithms burdened by exponential increases in computing time with problem size can be executed in a single step on a quantum computer.

While still considered exotic technology, major corporations are actively exploring and deriving practical insights from quantum computing experimentation. Despite its current status, the field holds promise for transformative advancements in computing in the years to come.

Potential speedup

Speedups in quantum computing can be remarkable due to the unique scalability of quantum algorithms, which differ significantly from the scaling patterns of traditional algorithms. Tasks suited for quantum computing can potentially achieve virtually unlimited speedup, presenting exciting prospects for specific computing endeavors.

Advantages

- Quantum computing enables computations of unprecedented complexity that surpass the capabilities of traditional computers.

Disadvantages

- Hardware availability remains highly limited, with unreliable performance.
- Programming quantum computers demands a completely different skill set compared to traditional computing.
- Proficiency in these specialized skills is scarce.

VORTECH

About VORtech

VORtech comprises a team of scientific software engineers renowned for their expertise in crafting high-tech computational applications. Our professionals blend a fervent dedication to software development with profound insights into engineering and mathematics. We thrive on close collaboration with our clients' domain experts, leveraging their specialized domain knowledge alongside our technical acumen, together delivering the best possible solutions.

At VORtech, we offer a diverse array of expertise necessary for crafting top-tier computational software. Our foundation lies in mastery of software development across a broad spectrum of programming languages, including Fortran, C, C#, Python, and Matlab. Furthermore, our team includes specialists in high-performance computing, data-model fusion, machine learning, and web development.

Our typical engagement commences with a meticulous model scan, wherein we analyze the client's software and provide insights into prevailing issues. Subsequently, we embark on short-term projects to showcase our capabilities by resolving a specific challenge. This often leads to a strategic partnership, wherein VORtech becomes a valued collaborator, tailoring our involvement to match the evolving needs of the client.

Our clients are primarily large companies and research institutes. Our role is that of bringing software from a low TRL-level, often coming from the research department, to the high TRL-levels that are needed for operational use.