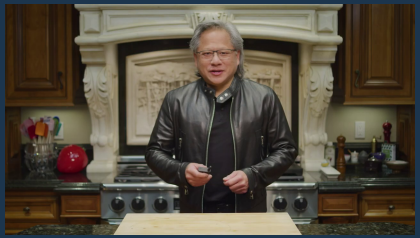
Retour GTC 2021 : GPU Technical Conference

Pierre Aubert

Introduction

GTC 2021 `https://gtc21.event.nvidia.com/`

This is not possible to summarize **1600** talks !



Keynotes video link

Only focus on :

- **Hardwares** evolution
- **Compilers** evolution
- **Programming Languages** evolution
- **Machine Learning** / **Deep Learning** evolution and use
- **Job Communication** / **Job Placement**
- **Cybersecurity**
- How all (will) work together (with a bit of **Forum ORAP**)

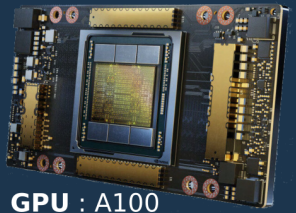
NVidia news

NVidia news

GTC 2020 :
- **NVidia** bought **Mellanox**

NVidia news

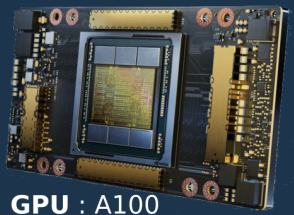
GTC 2020 :
- **NVidia** bought **Mellanox**



GPU : A100
40 GB and **80** GB

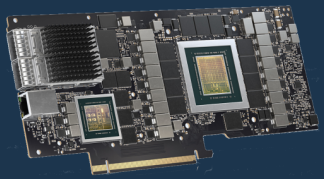
NVidia news

GTC 2020 :
- **NVidia** bought **Mellanox**



GPU : A100
40 GB and **80** GB

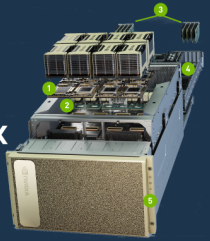
DPU : Bluefield 2

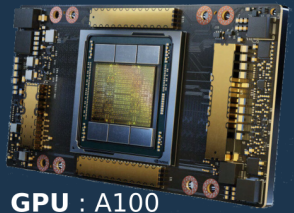


NVidia news

GTC 2020 :
- **NVidia** bought **Mellanox**

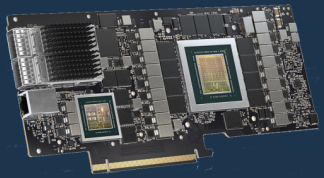
DGX





GPU : A100
40 GB and **80** GB

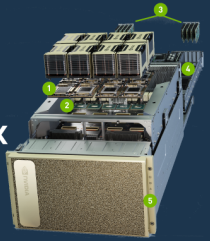
DPU : Bluefield 2

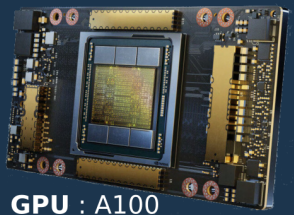


NVidia news

GTC 2020 :
- **NVidia** bought **Mellanox**

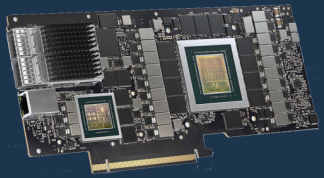
DGX





GPU : A100
40 GB and **80** GB

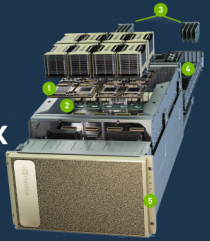
DPU : Bluefield 2



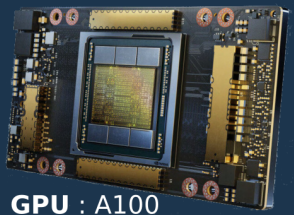
GTC 2021 :
- **NVidia** bought **ARM**

NVidia news

GTC 2020 :
- **NVidia** bought **Mellanox**

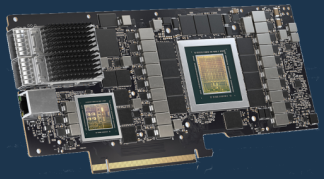


DGX

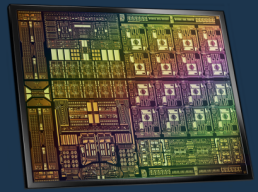


GPU : A100
40 GB and **80** GB

DPU : Bluefield 2



DPU : Bluefield 3

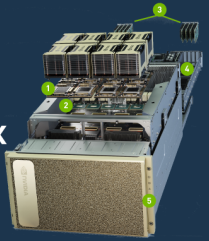


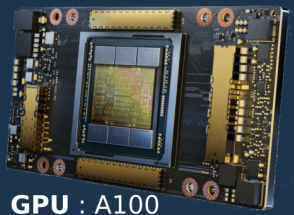
GTC 2021 :
- **NVidia** bought **ARM**

NVidia news

GTC 2020 :
- **NVidia** bought **Mellanox**

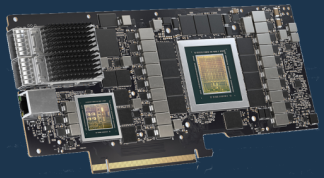
DGX





GPU : A100
40 GB and **80** GB

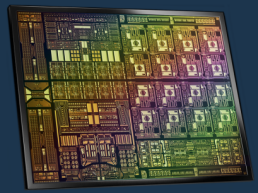
DPU : Bluefield 2



GTC 2021 :
- **NVidia** bought **ARM**

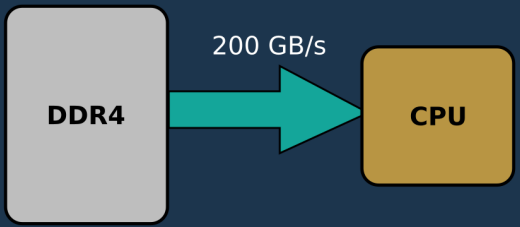
DGX

DPU : Bluefield 3

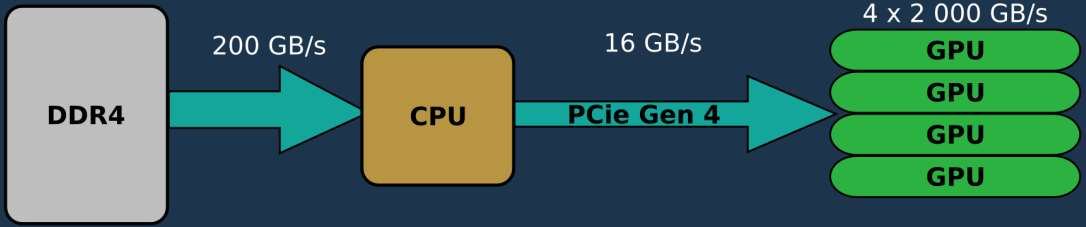


Hardware evolution (NVidia)

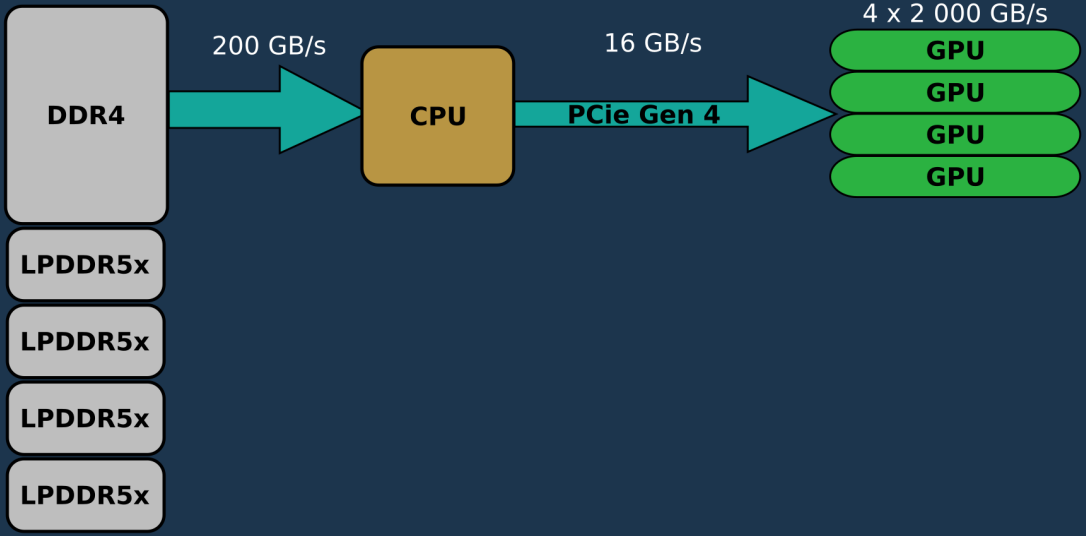
Hardware evolution (NVidia)



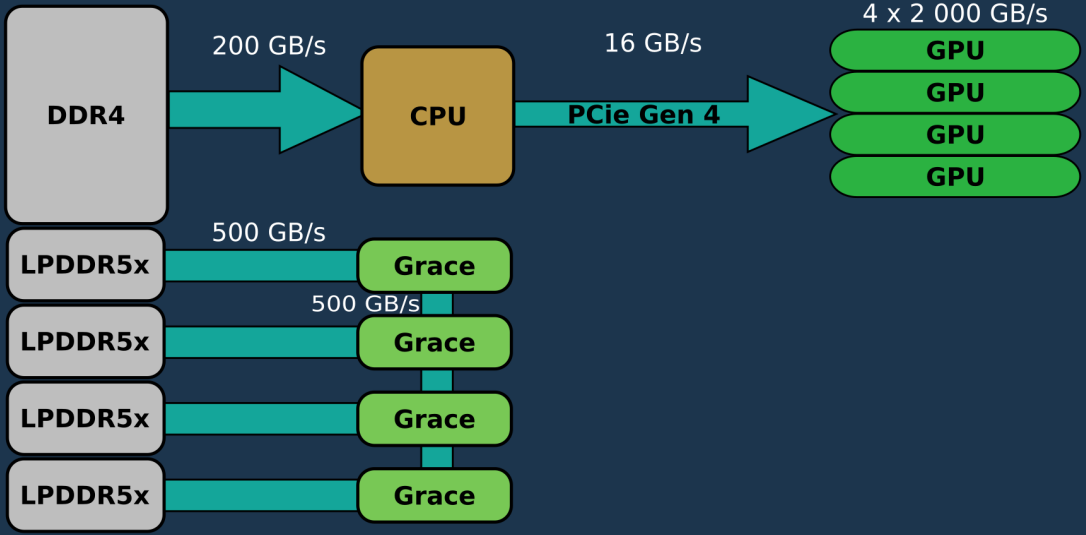
Hardware evolution (NVidia)



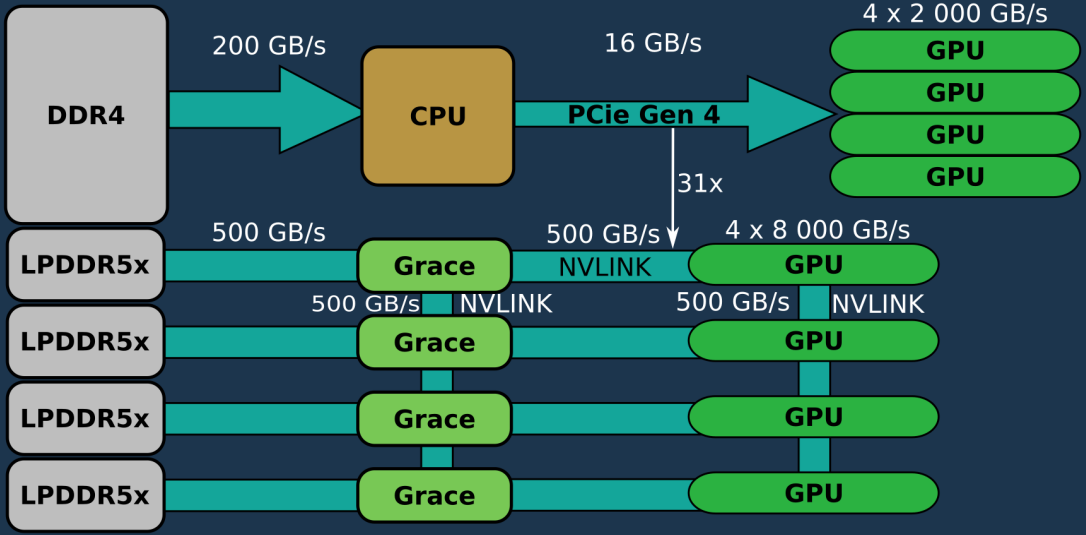
Hardware evolution (NVidia)



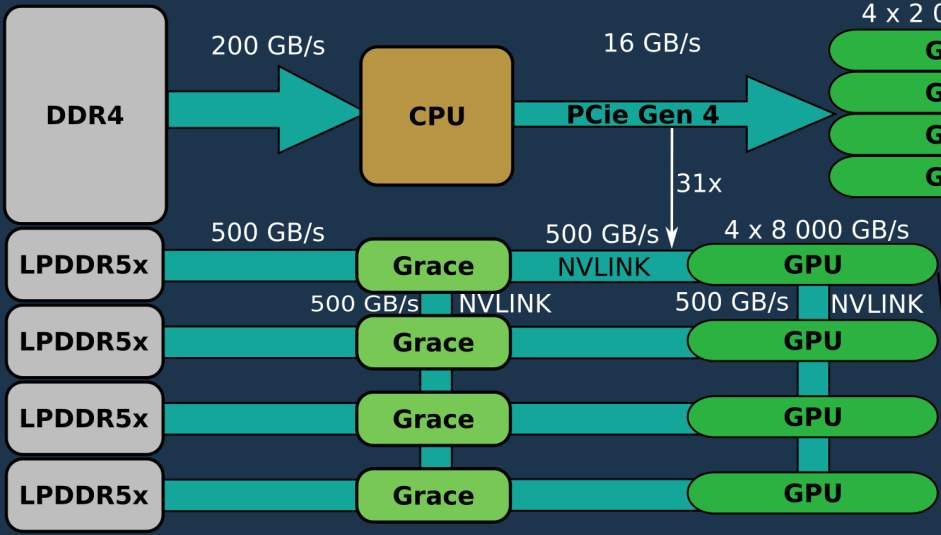
Hardware evolution (NVidia)

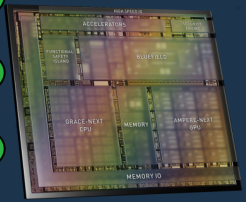


Hardware evolution (NVidia)



Hardware evolution (NVidia)

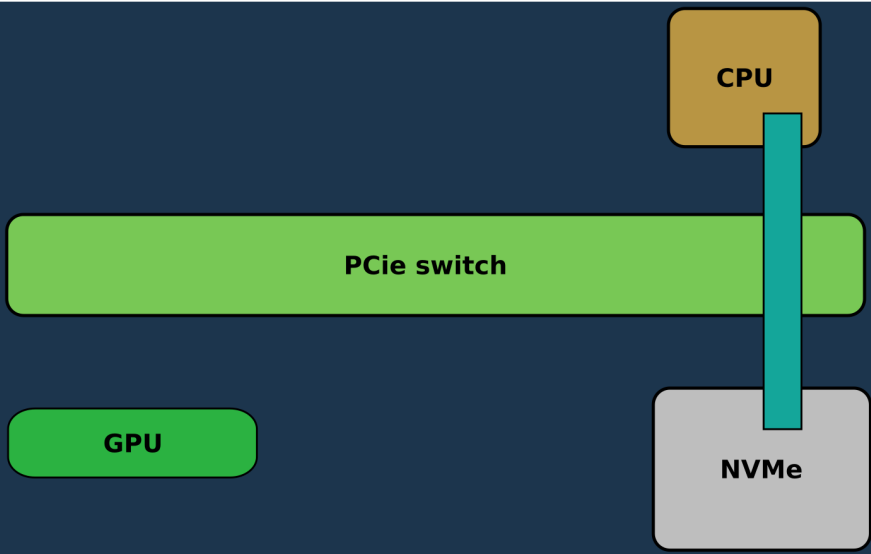




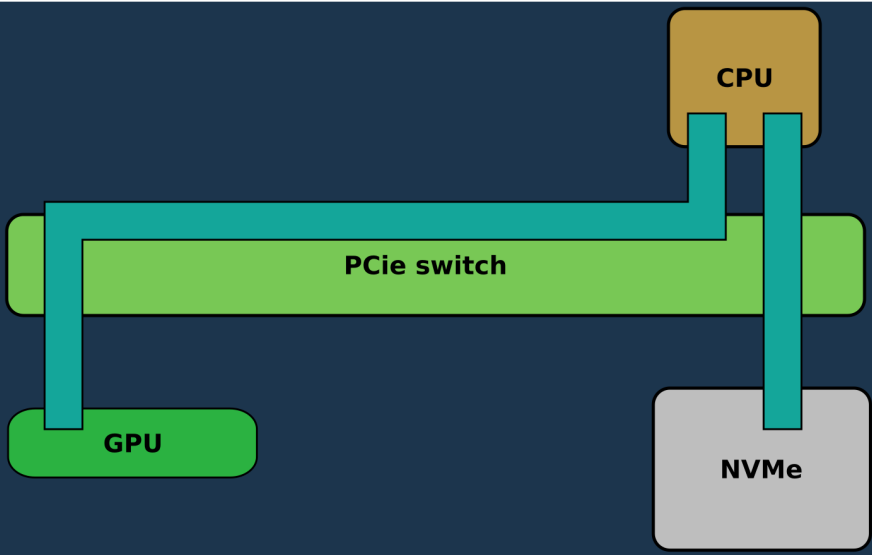
Altan

Hardware evolution (NVidia)

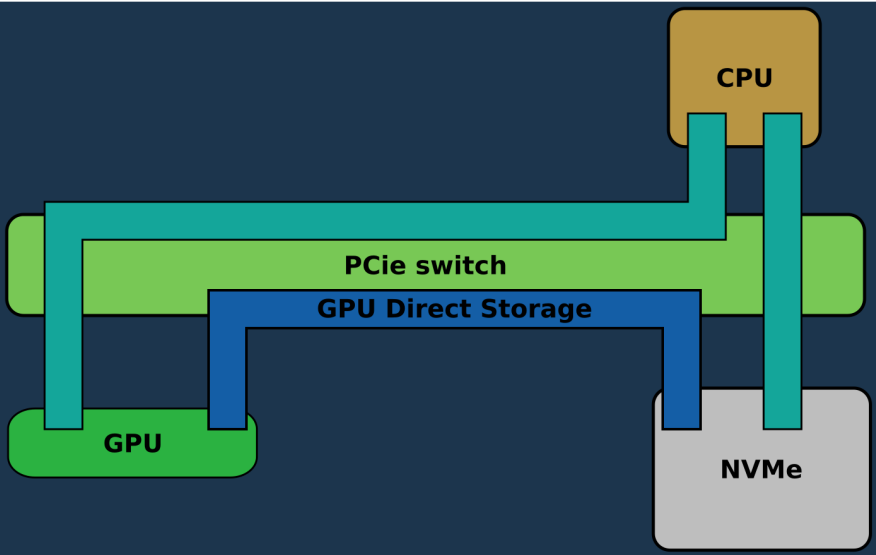
Hardware evolution (NVidia)



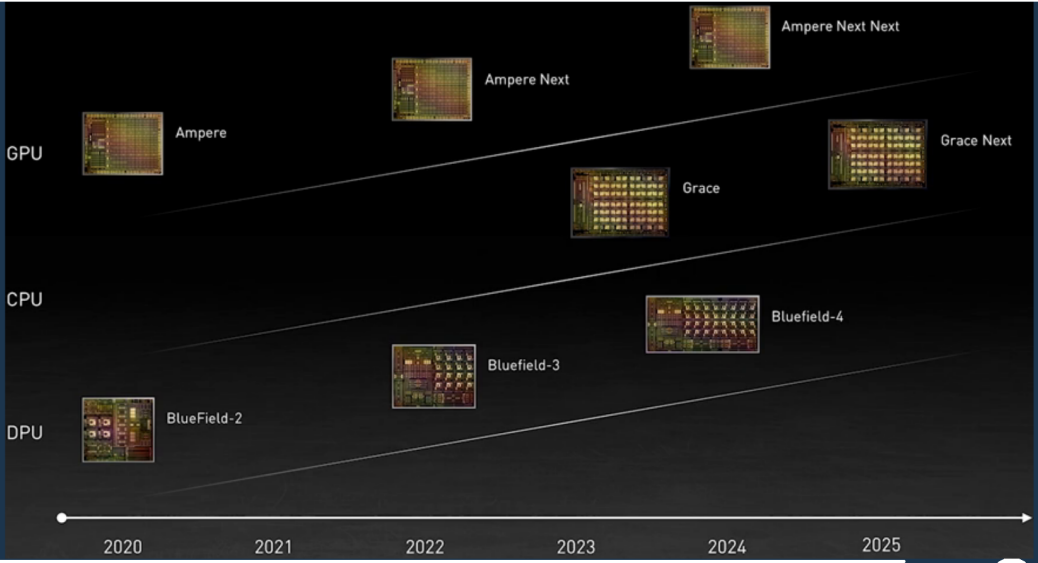
Hardware evolution (NVidia)



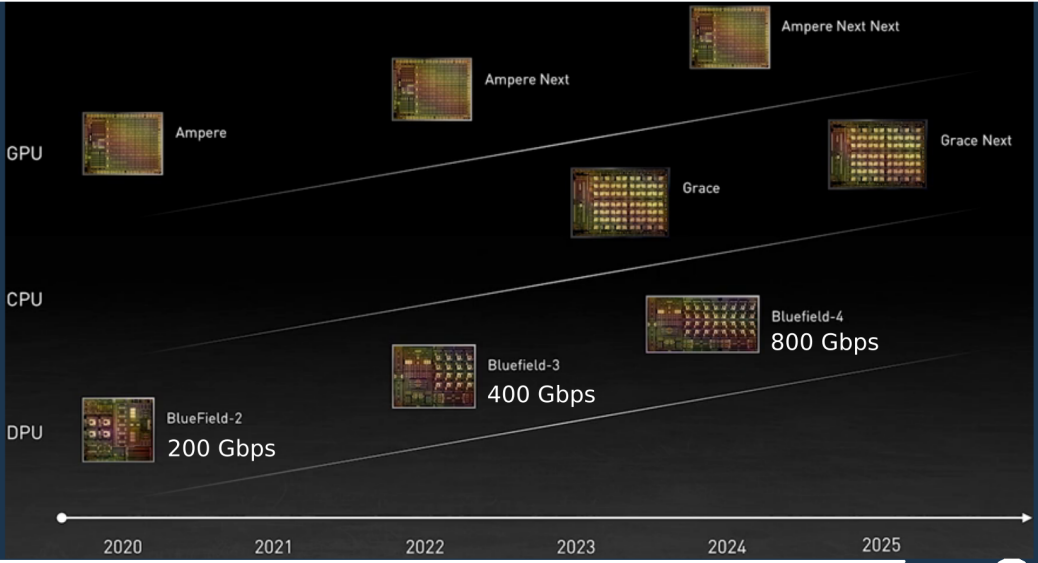
Hardware evolution (NVidia)



Hardware evolution (NVidia)



Hardware evolution (NVidia)



NVCC, NVC++ evolution (NVidia)

NVCC, NVC++ evolution (NVidia)

Today

NVCC, NVC++ evolution (NVidia)

Today

nvc++

C++17 : **std::transform**
Fortran : **do concurent**

NVCC, NVC++ evolution (NVidia)

Today

nvc++

-stdpart
target GPU

C++17 : **std::transform**
Fortran : **do concurent**

NVCC, NVC++ evolution (NVidia)

Today

nvc++

-stdpart
target GPU

C++17 : **std::transform**
Fortran : **do concurent**

Compute Capabilities :
- Ampere : 8
- Turing : 7
- Volta : 7
- Pascal : 6

NVCC, NVC++ evolution (NVidia)

Today

nvc++

-stdpart
target GPU

C++17 : **std::transform**
Fortran : **do concurent**

Compute Capabilities :
- Ampere : 8
- Turing : 7
- Volta : 7
- Pascal : 6

nvcc

CUDA_ARCH

__host__
__device__
__global__

NVCC, NVC++ evolution (NVidia)

Today

Future (2022-2023)

nvc++

-stdpar target GPU

C++17 : **std::transform**
Fortran : **do concurent**

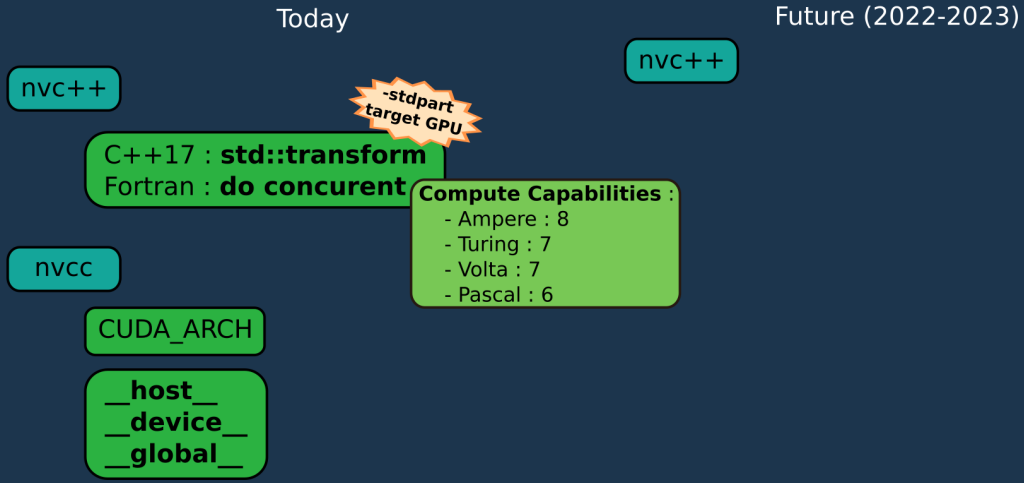
Compute Capabilities :
- Ampere : 8
- Turing : 7
- Volta : 7
- Pascal : 6

nvcc

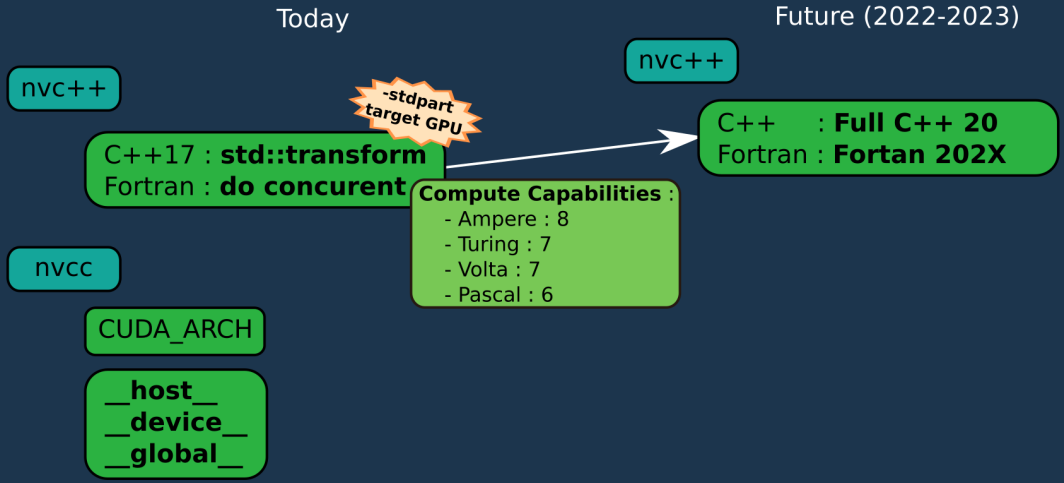
CUDA_ARCH

__host__
__device__
__global__

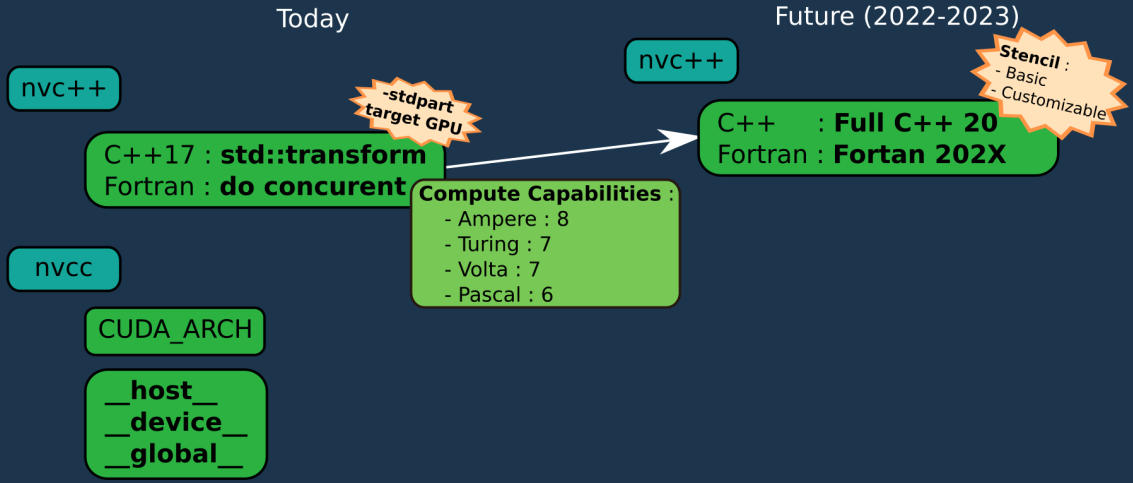
NVCC, NVC++ evolution (NVidia)



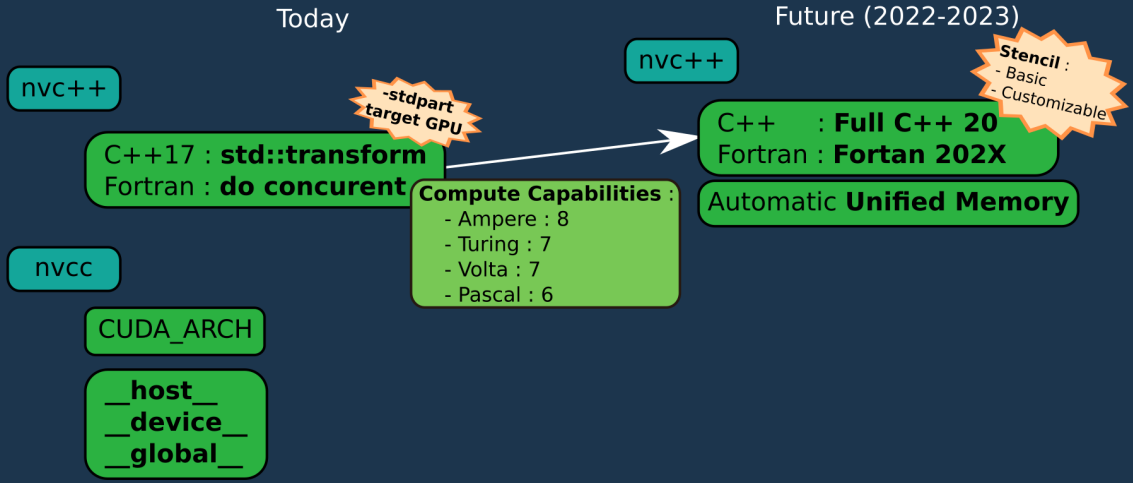
NVCC, NVC++ evolution (NVidia)



NVCC, NVC++ evolution (NVidia)



NVCC, NVC++ evolution (NVidia)



NVCC, NVC++ evolution (NVidia)

Today

nvc++

C++17 : **std::transform**
Fortran : **do concurent**

-stdpart target GPU

Compute Capabilities :
- Ampere : 8
- Turing : 7
- Volta : 7
- Pascal : 6

nvcc

CUDA_ARCH

__host__
__device__
__global__

Future (2022-2023)

nvc++

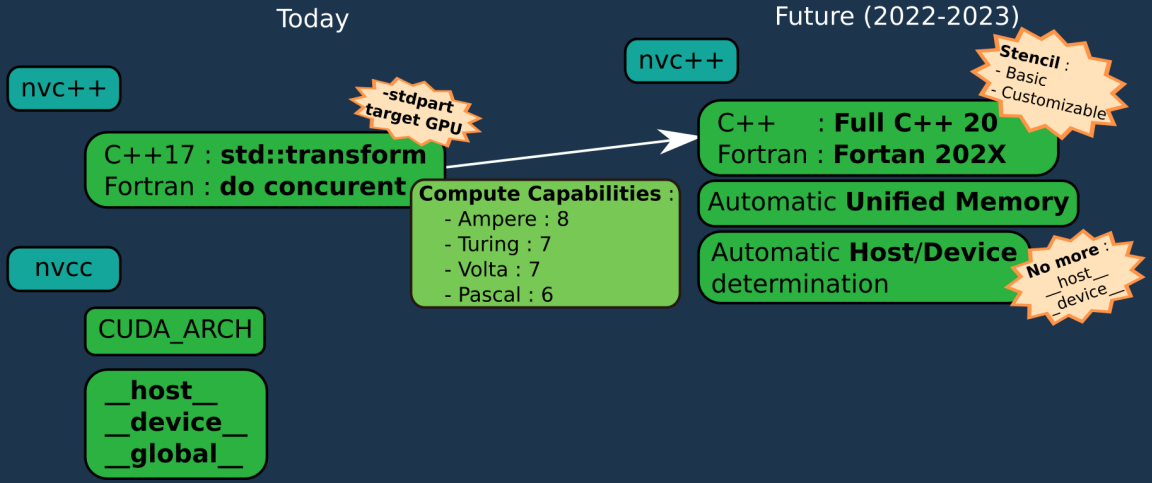
C++ : **Full C++ 20**
Fortran : **Fortan 202X**

Automatic **Unified Memory**

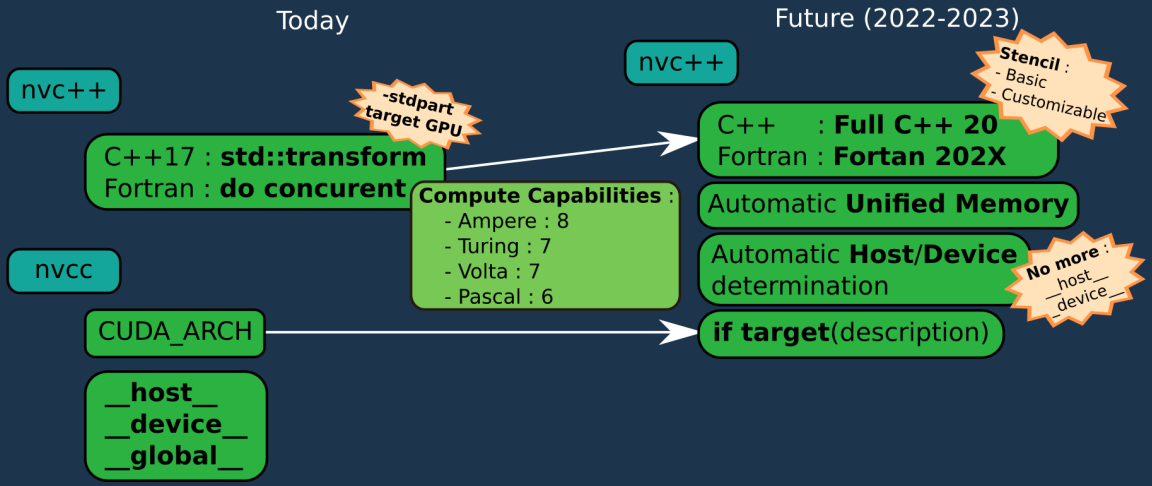
Automatic **Host/Device** determination

Stencil :
- Basic
- Customizable

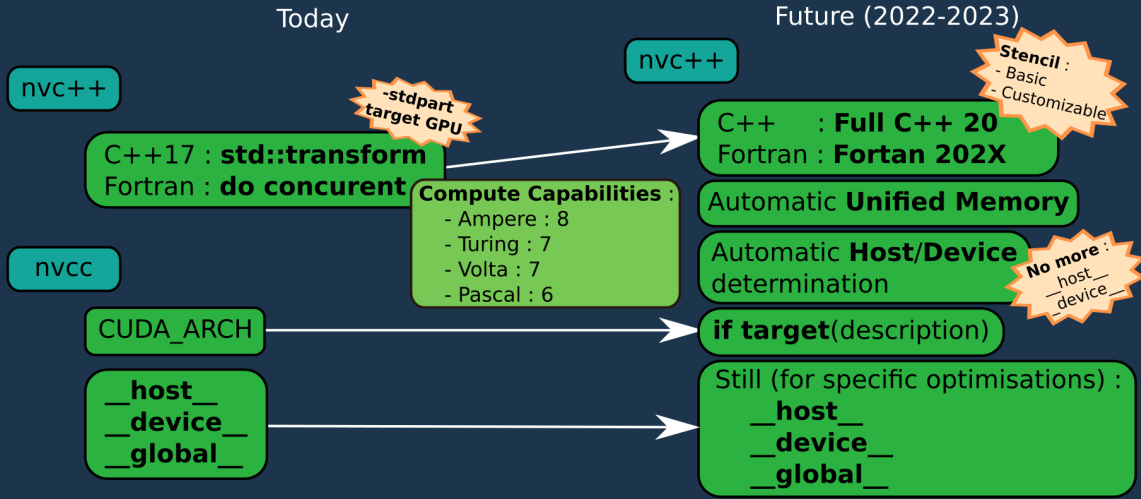
NVCC, NVC++ evolution (NVidia)



NVCC, NVC++ evolution (NVidia)



NVCC, NVC++ evolution (NVidia)



GPU and FORTRAN (NVFortran)

```
d = 2.5 * ceil(transpose(a)) + 3.0 * abs(transpose(b))
d = 2.5 * ceil(transpose(a)) + 3.0 * abs(b)
d = reshape(a,shape=[ni,nj,nk])
d = reshape(a,shape=[ni,nk,nj])
d = 2.5 * sqrt(reshape(a,shape=[ni,nk,nj],order=[1,3,2]))
d = alpha * conjg(reshape(a,shape=[ni,nk,nj],order=[1,3,2]))
d = reshape(a,shape=[ni,nk,nj],order=[1,3,2])
d = reshape(a,shape=[nk,ni,nj],order=[2,3,1])
d = reshape(a,shape=[ni*nj,nk])
d = reshape(a,shape=[nk,ni*nj],order=[2,1])
d = reshape(a,shape=[64,2,16,16,64],order=[5,2,3,4,1])
d = abs(reshape(a,shape=[64,2,16,16,64],order=[5,2,3,4,1]))
c = matmul(a,b)
c = matmul(transpose(a),b)
c = matmul(reshape(a,shape=[m,k],order=[2,1]),b)
c = matmul(a,transpose(b))
c = matmul(a,reshape(b,shape=[k,n],order=[2,1]))
```

```
c = matmul(transpose(a),transpose(b))
c = matmul(transpose(a),reshape(b,shape=[k,n],order=[2,1]))
d = spread(a,dim=3,ncopies=nk)
d = spread(a,dim=1,ncopies=ni)
d = spread(a,dim=2,ncopies=nx)
d = alpha * abs(spread(a,dim=2,ncopies=nx))
d = alpha * spread(a,dim=2,ncopies=nx)
d = abs(spread(a,dim=2,ncopies=nx))
d = transpose(a)
d = alpha * transpose(a)
d = alpha * ceil(transpose(a))
d = alpha * conjg(transpose(a))
c = c + matmul(a,b)
c = c - matmul(a,b)
c = c + alpha * matmul(a,b)
d = alpha * matmul(a,b) + c
d = alpha * matmul(a,b) + beta * c
```

Paradigms evolution

Current default situation

New default situation

Paradigms evolution

Current default situation

Scalar

New default situation

Paradigms evolution

Current default situation

Scalar

Single-threaded

New default situation

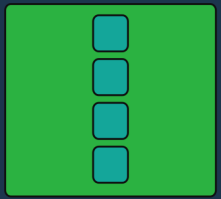
Paradigms evolution

Current default situation

Scalar

Single-threaded

Execution **in order**



New default situation

Paradigms evolution

Current default situation

Scalar

Single-threaded

Execution **in order**

Recent machine :
- 64 cores
- 16 SIMD units

uses < 0.1% of computing power

New default situation

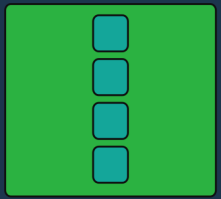
Paradigms evolution

Current default situation

Scalar

Single-threaded

Execution **in order**



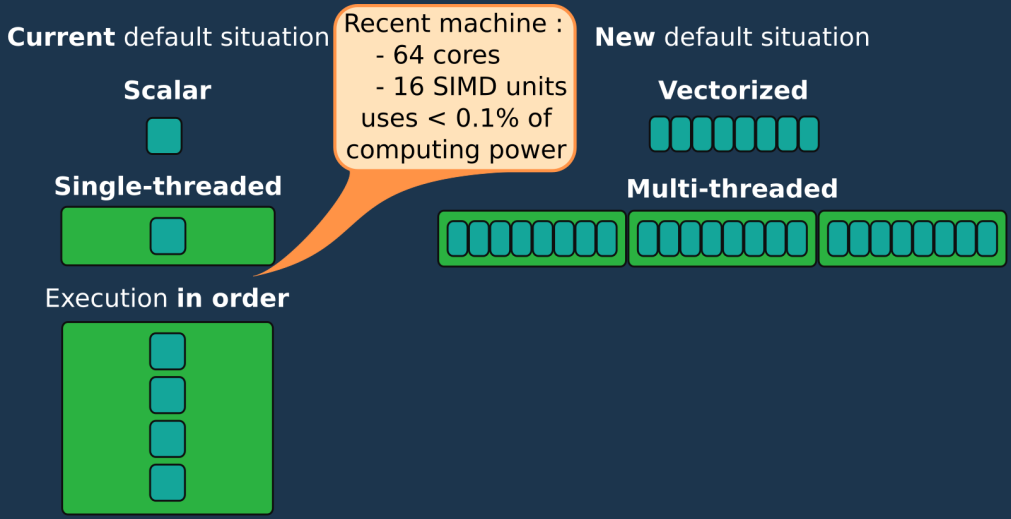
Recent machine :
- 64 cores
- 16 SIMD units

uses < 0.1% of computing power

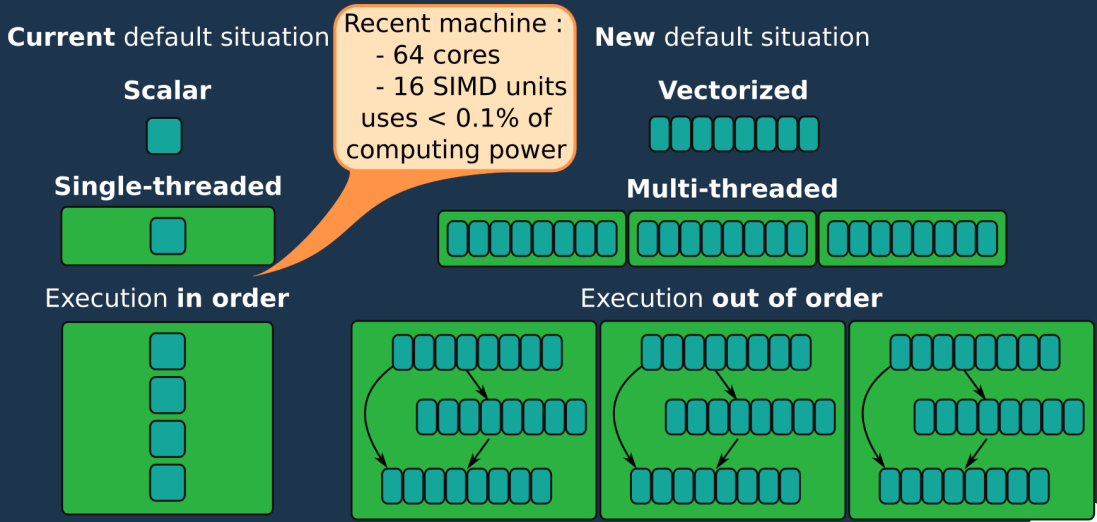
New default situation

Vectorized

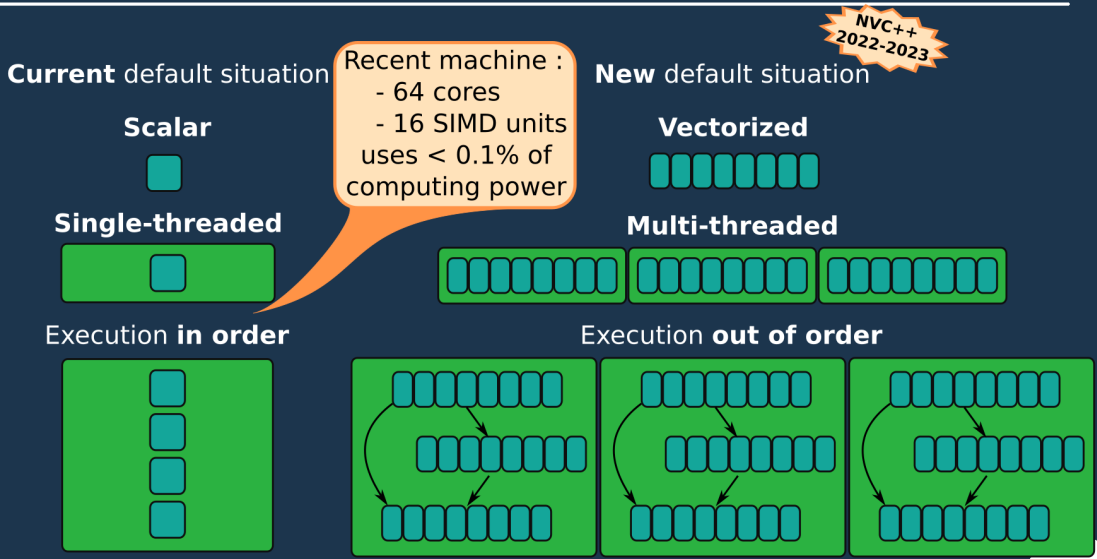
Paradigms evolution



Paradigms evolution



Paradigms evolution



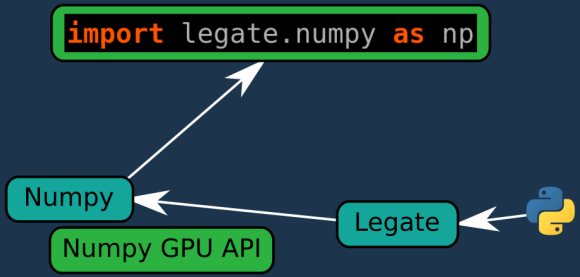
Python (ecosystem) evolution (NVidia)

Python (ecosystem) evolution (NVidia)

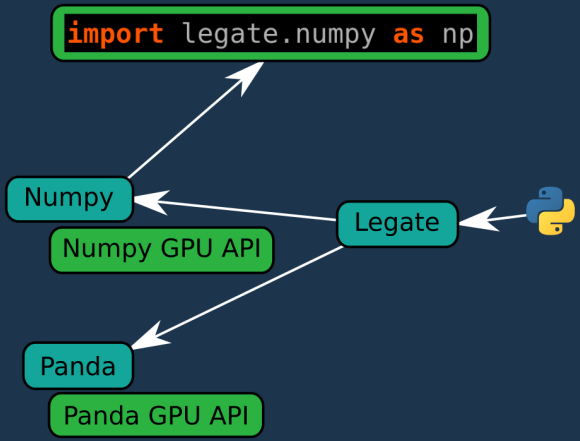
Python (ecosystem) evolution (NVidia)

Python (ecosystem) evolution (NVidia)

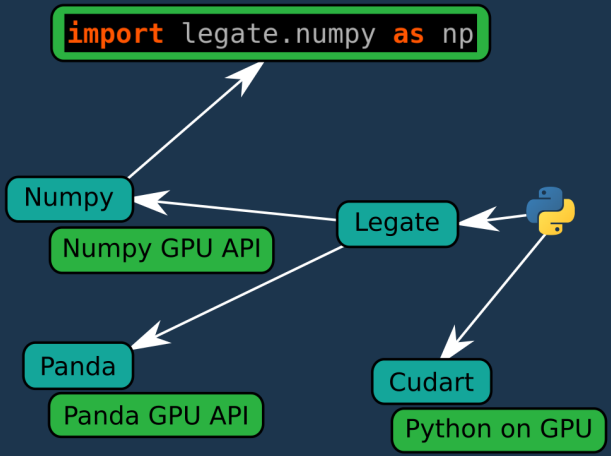
Python (ecosystem) evolution (NVidia)



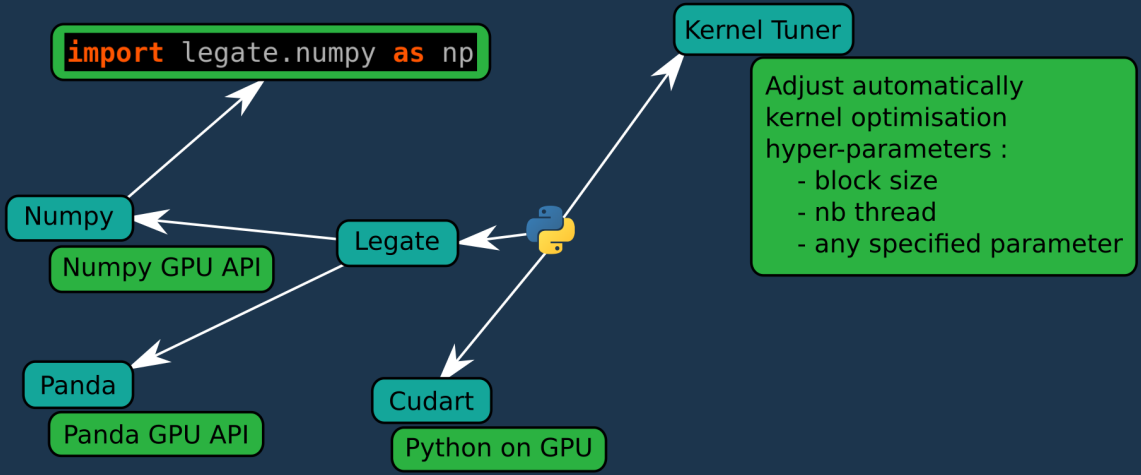
Python (ecosystem) evolution (NVidia)



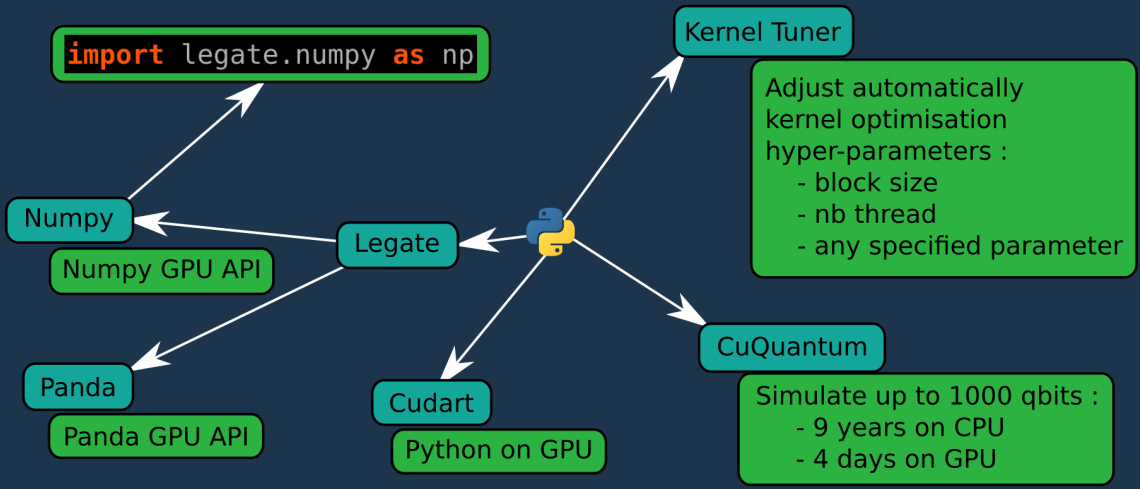
Python (ecosystem) evolution (NVidia)



Python (ecosystem) evolution (NVidia)



Python (ecosystem) evolution (NVidia)

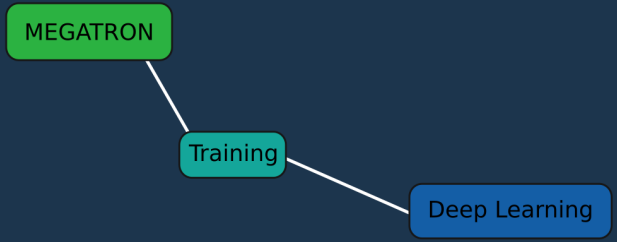


Deep Learning evolution

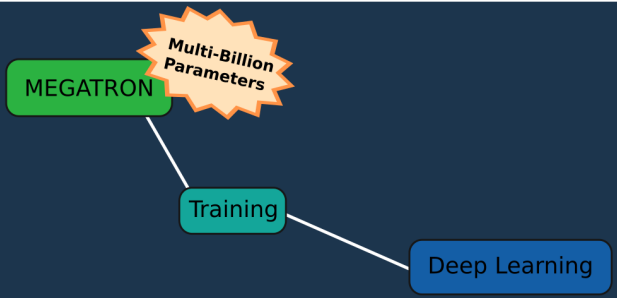
Deep Learning

Deep Learning evolution

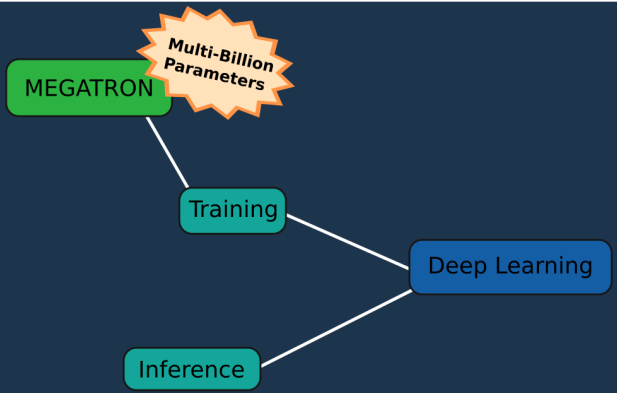
Deep Learning evolution



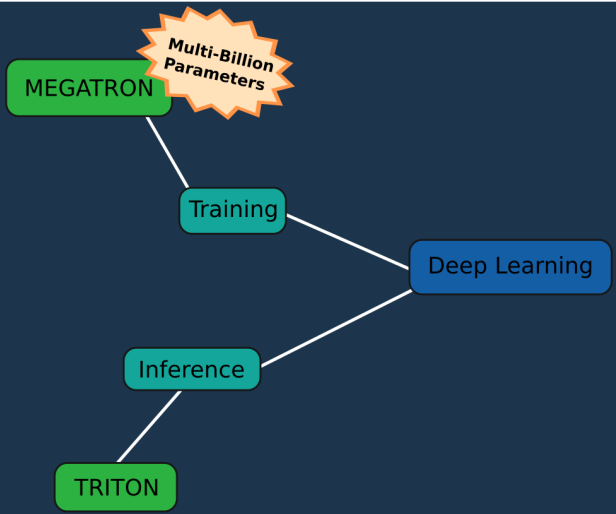
Deep Learning evolution



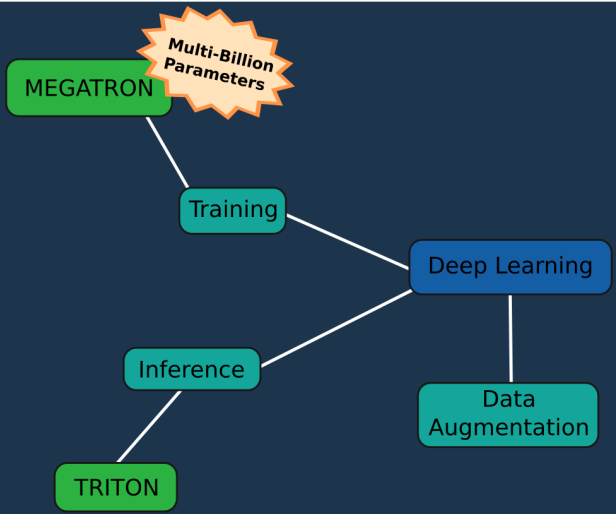
Deep Learning evolution



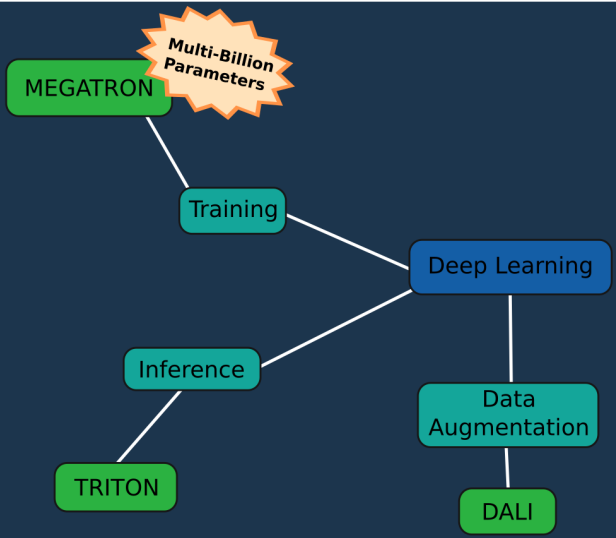
Deep Learning evolution



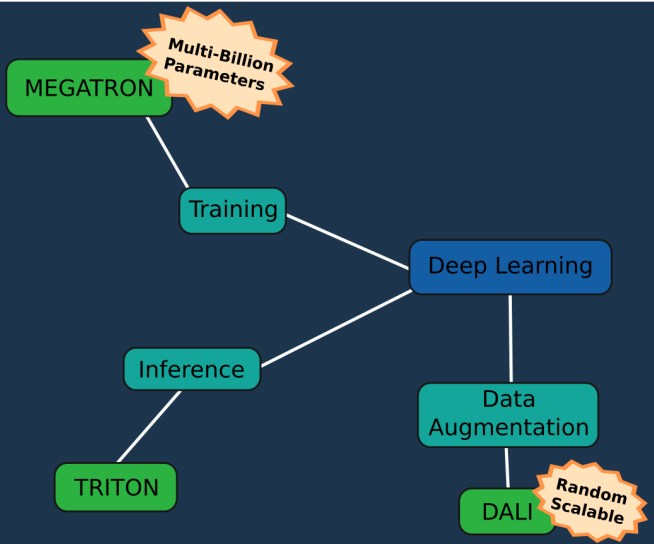
Deep Learning evolution



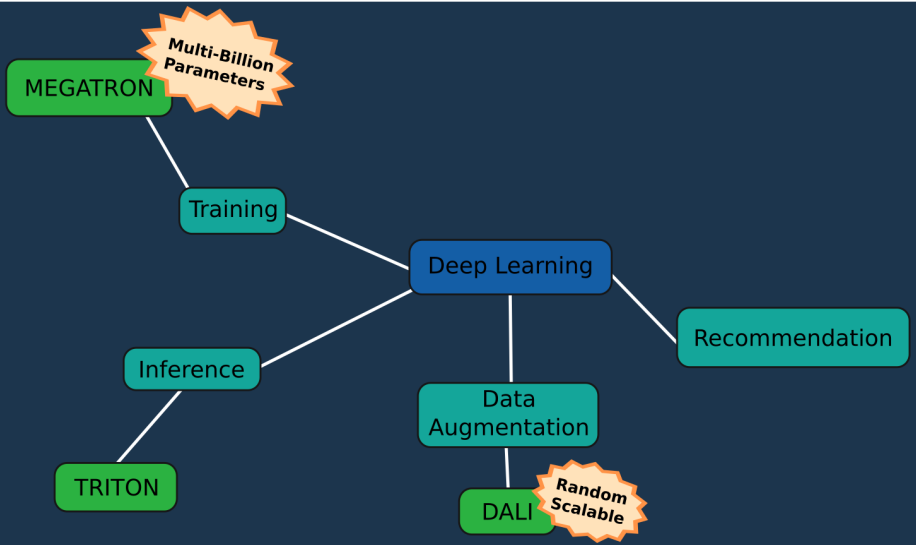
Deep Learning evolution



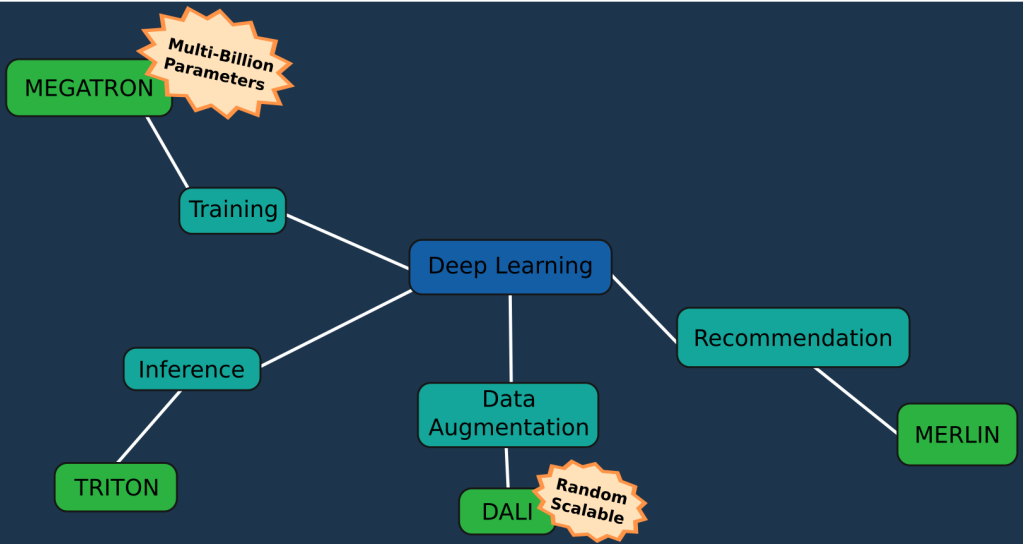
Deep Learning evolution



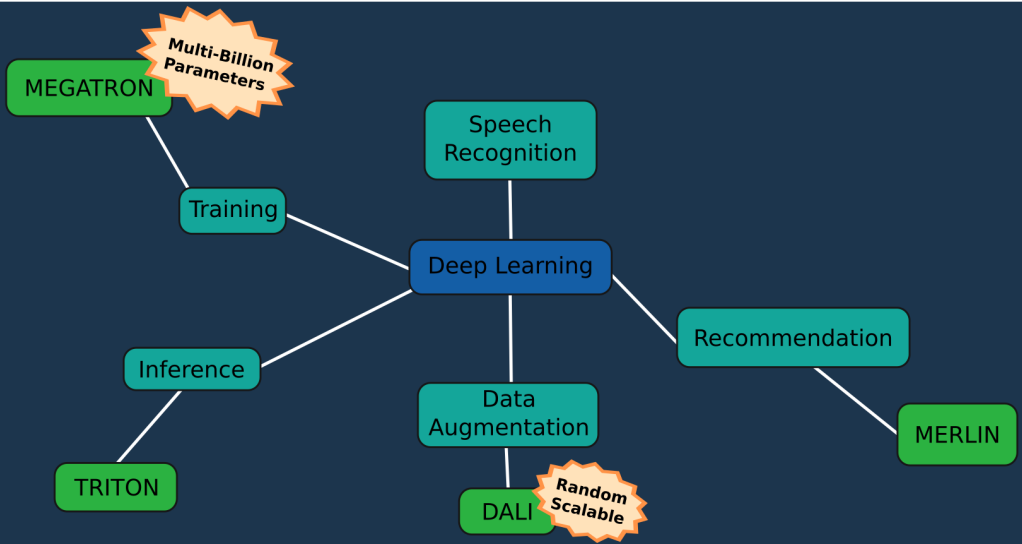
Deep Learning evolution



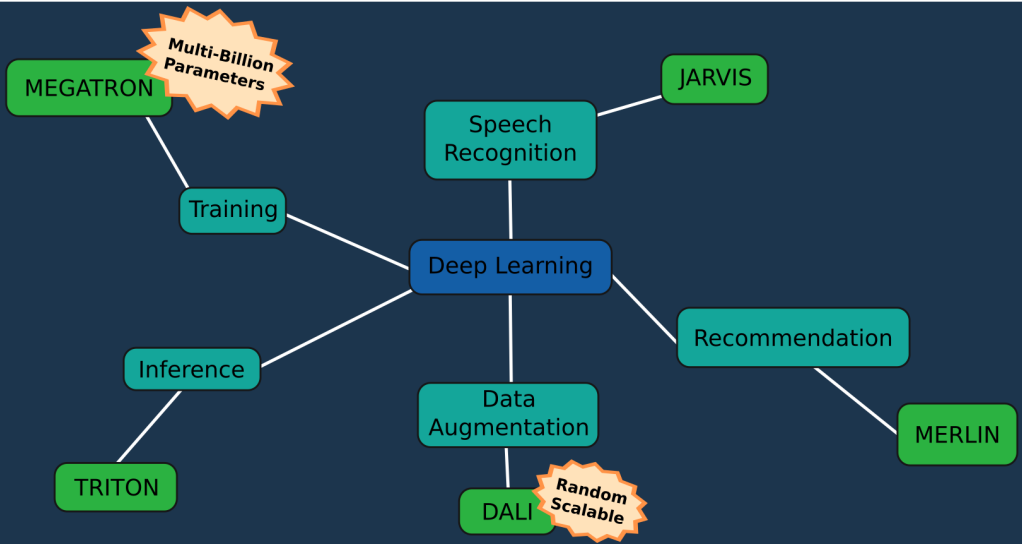
Deep Learning evolution



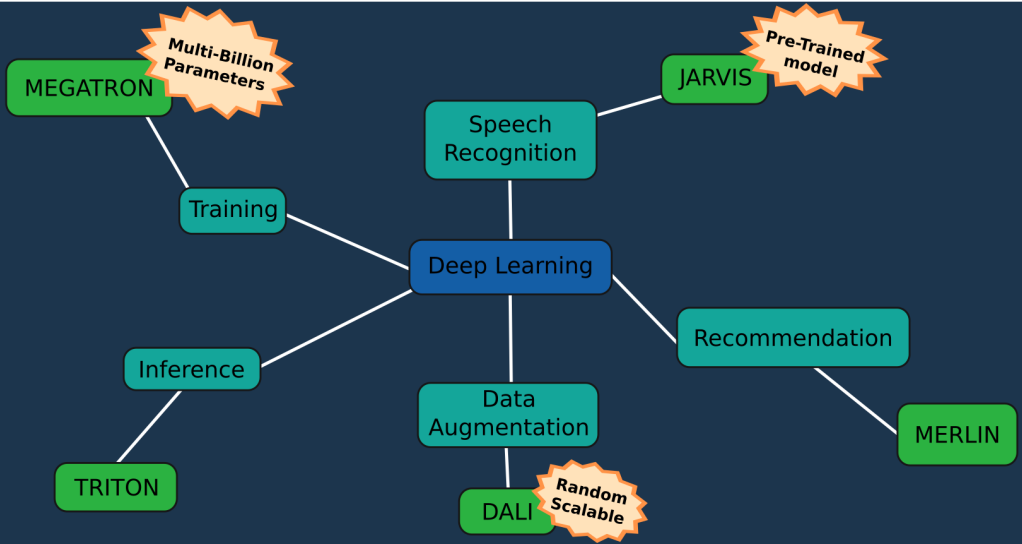
Deep Learning evolution



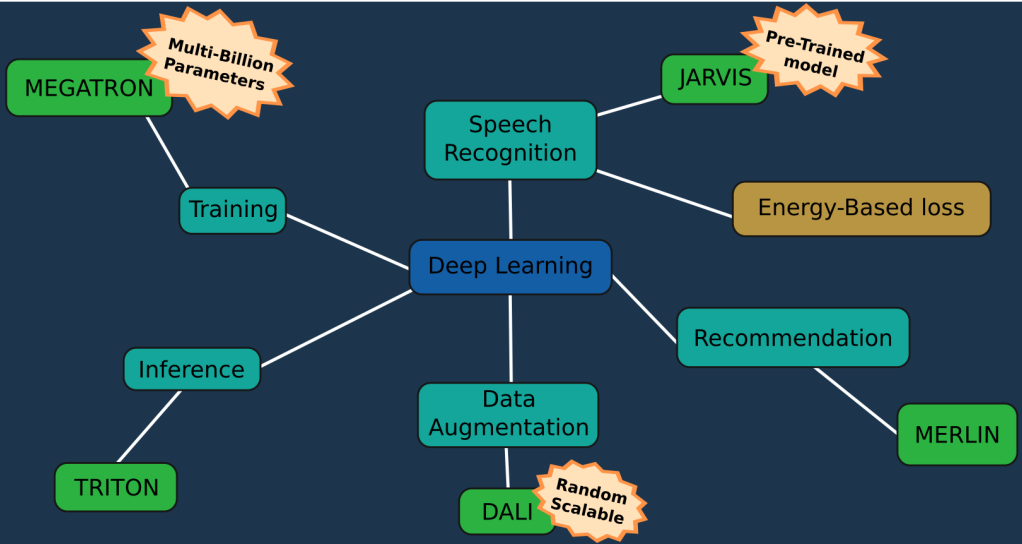
Deep Learning evolution



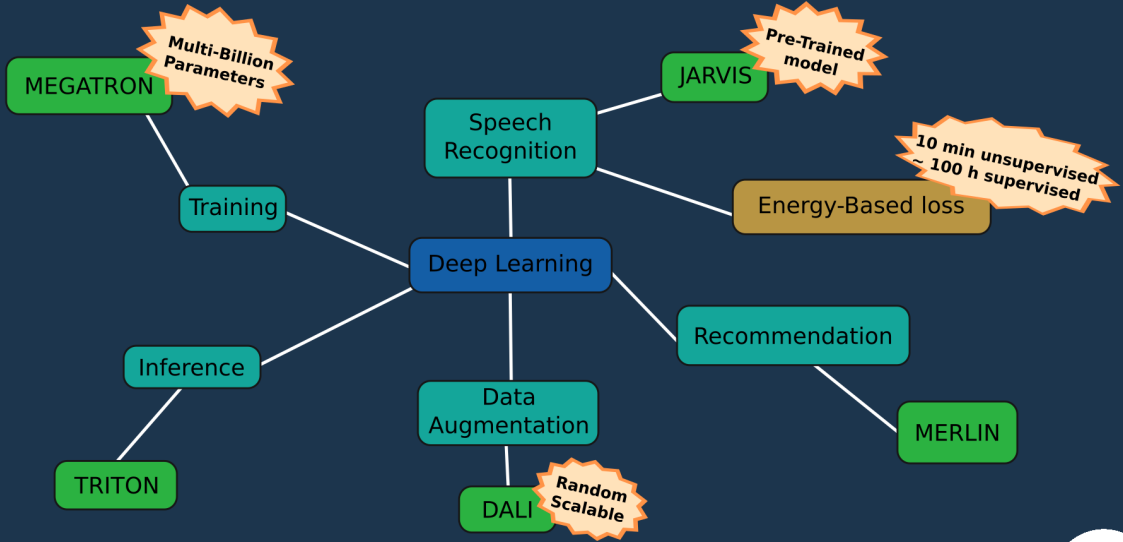
Deep Learning evolution



Deep Learning evolution

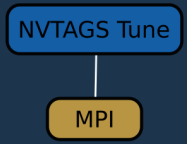


Deep Learning evolution

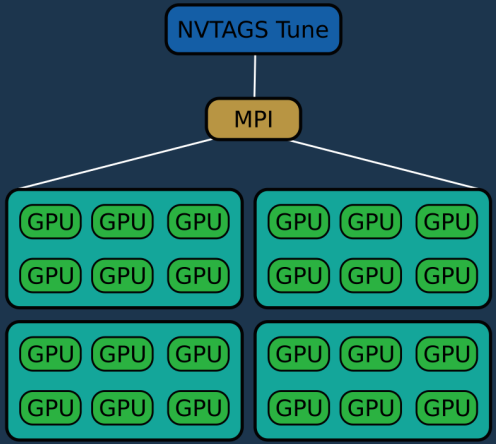


Topology-Aware GPU Selection (NVTAGS)

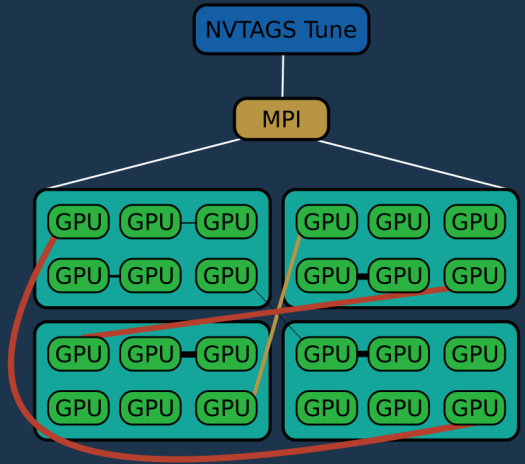
Topology-Aware GPU Selection (NVTAGS)



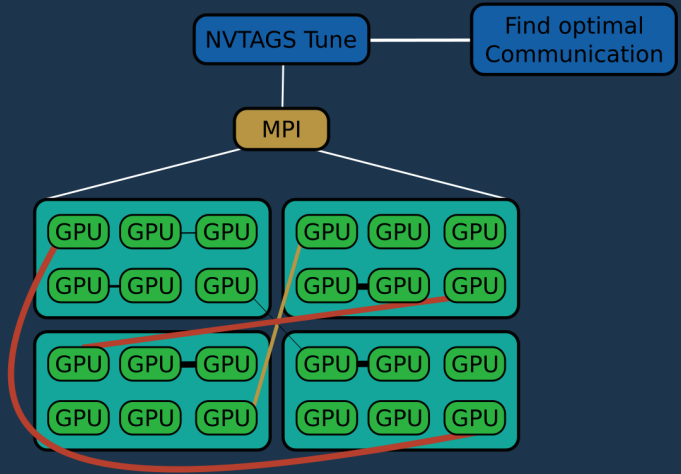
Topology-Aware GPU Selection (NVTAGS)



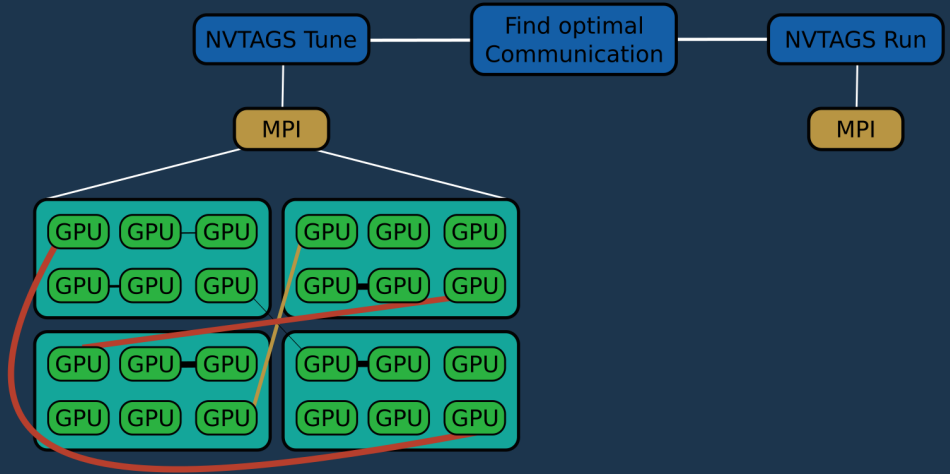
Topology-Aware GPU Selection (NVTAGS)



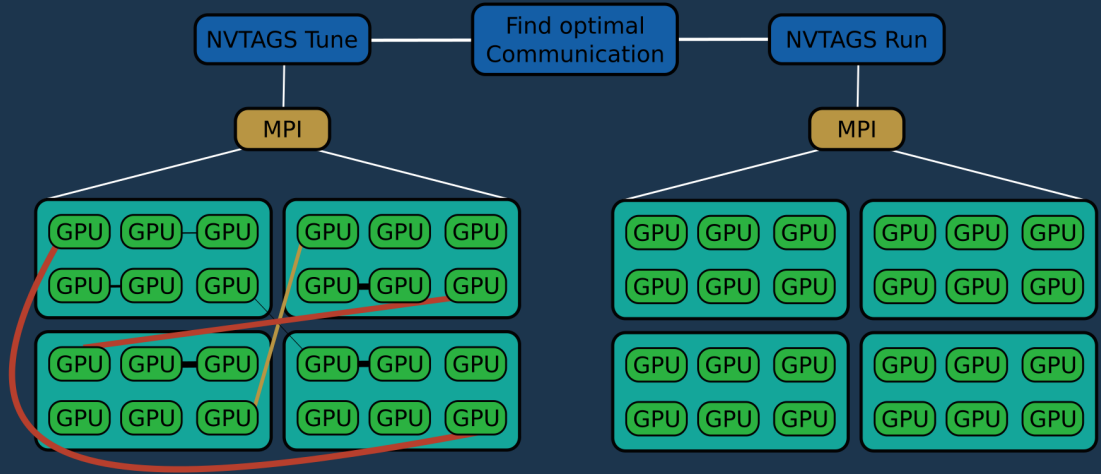
Topology-Aware GPU Selection (NVTAGS)



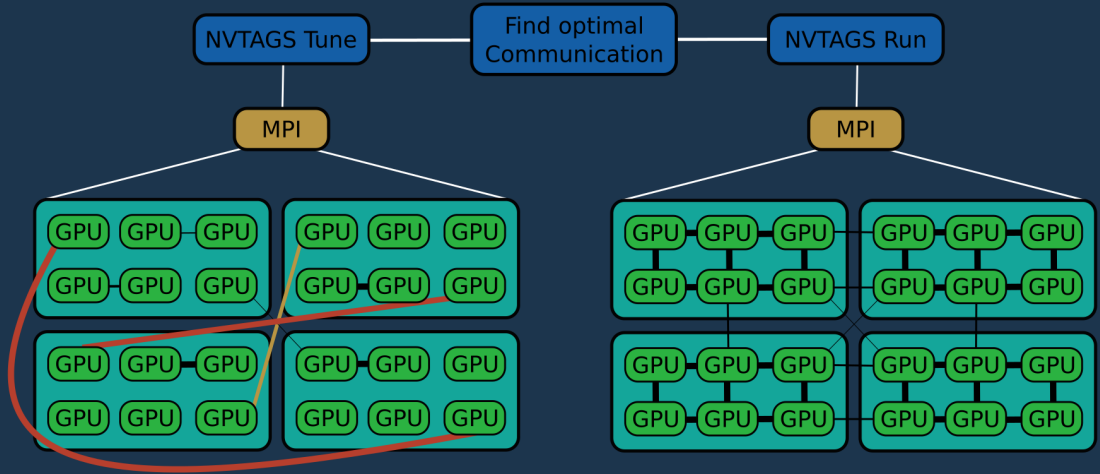
Topology-Aware GPU Selection (NVTAGS)



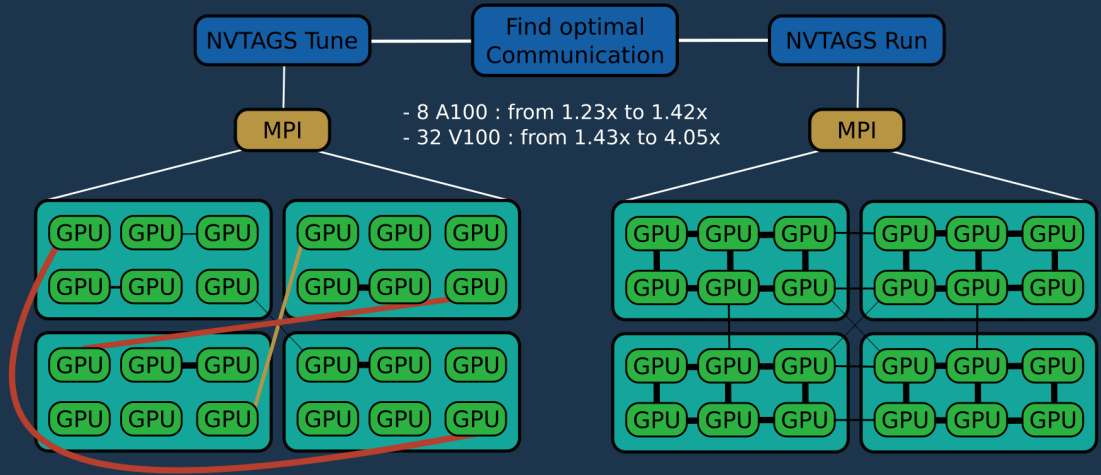
Topology-Aware GPU Selection (NVTAGS)



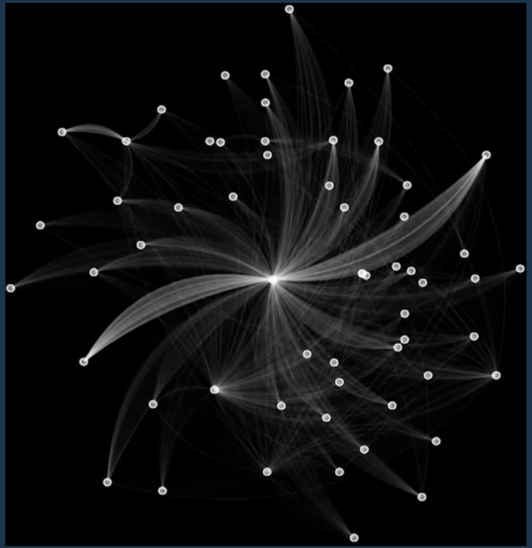
Topology-Aware GPU Selection (NVTAGS)



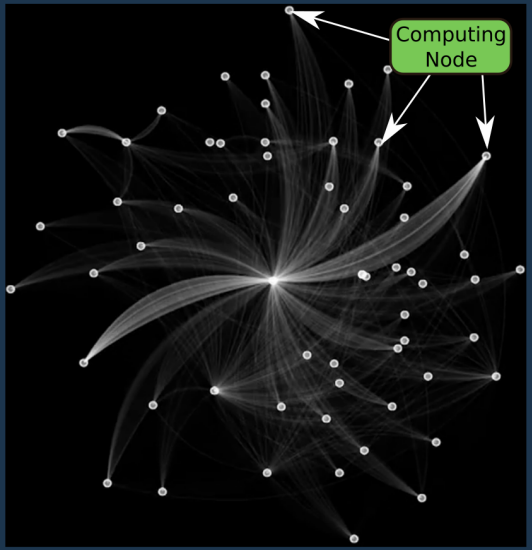
Topology-Aware GPU Selection (NVTAGS)



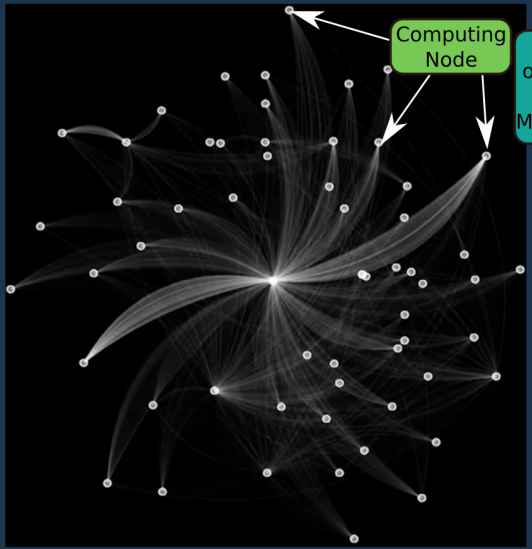
Morpheus (NVidia)



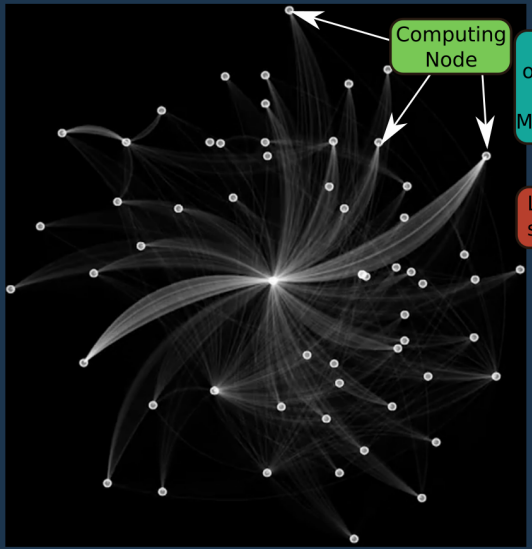
Morpheus (NVidia)



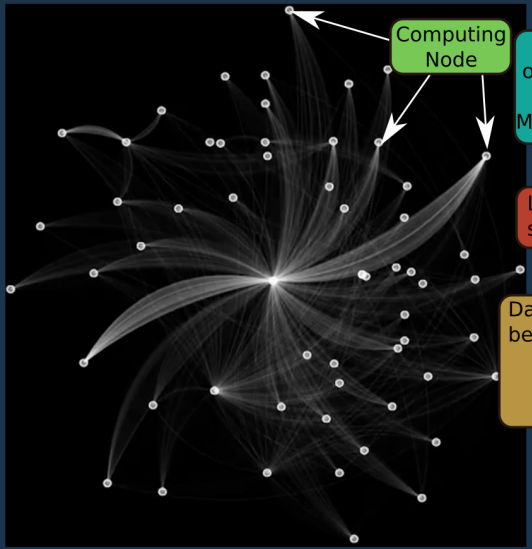
Morpheus (NVidia)



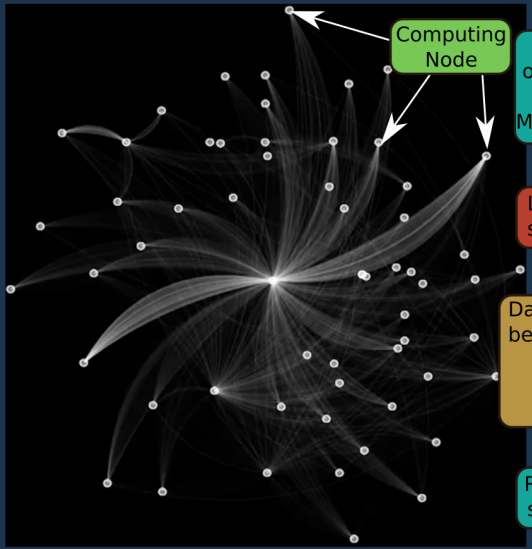
Morpheus (NVidia)



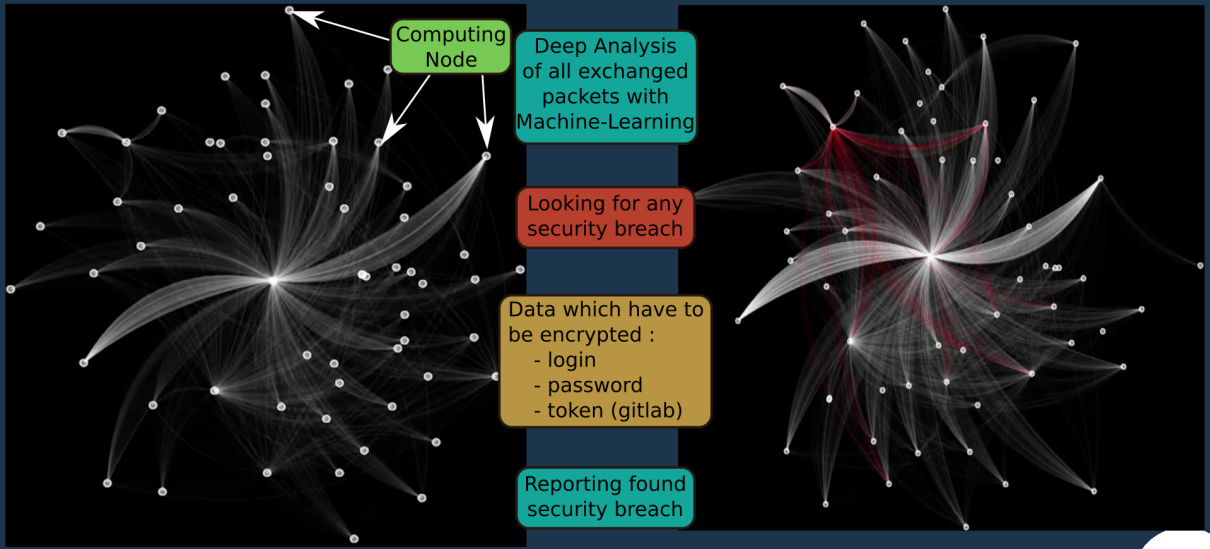
Morpheus (NVidia)



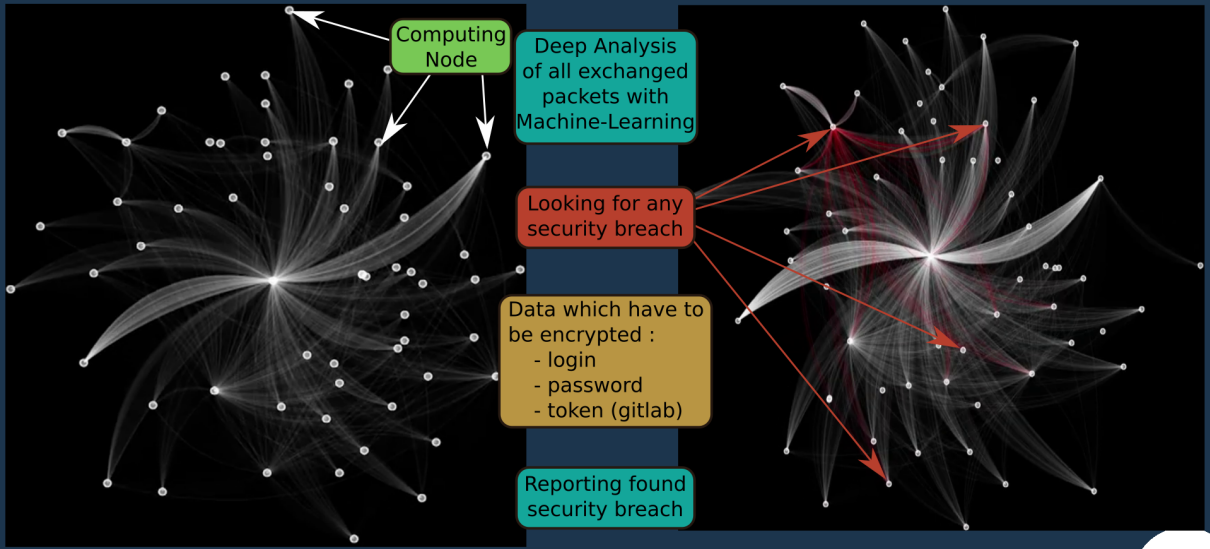
Morpheus (NVidia)



Morpheus (NVidia)



Morpheus (NVidia)



Energy consumption (ATOS)

2010 : 0.23 GFlops/W
2020 : 25 GFlops/W (Juelich supercomputer - 3 744 A100), free cooling

Energy consumption (ATOS)

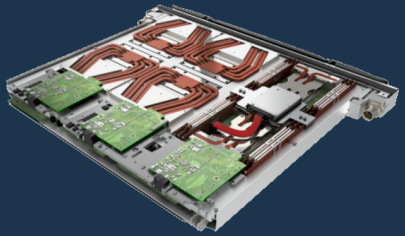
2010 : 0.23 GFlops/W
2020 : 25 GFlops/W (Juelich supercomputer - 3 744 A100), free cooling

- Classic Data Center : 100 % computing, 100 % cooling
- Water cooling Data Center : 100 % computing, 50 % cooling
- Direct liquid cooling Data Center : 100 % computing, 5 % cooling

Energy consumption (ATOS)

2010 : 0.23 GFlops/W
2020 : 25 GFlops/W (Juelich supercomputer - 3 744 A100), free cooling

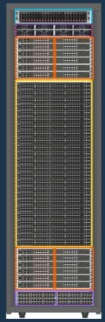
- Classic Data Center : 100 % computing, 100 % cooling
- Water cooling Data Center : 100 % computing, 50 % cooling
- Direct liquid cooling Data Center : 100 % computing, 5 % cooling



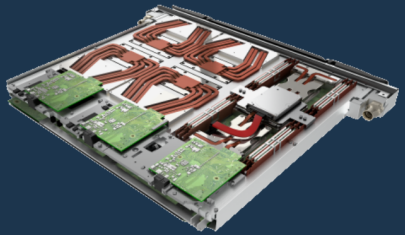
Bull-SequanaX1125

Energy consumption (ATOS)

2010 : 0.23 GFlops/W
2020 : 25 GFlops/W (Juelich supercomputer - 3 744 A100), free cooling



- Classic Data Center : 100 % computing, 100 % cooling
- Water cooling Data Center : 100 % computing, 50 % cooling
- Direct liquid cooling Data Center : 100 % computing, 5 % cooling



Bull-SequanaX1125

Everything is cooled :
- CPU
- GPU
- switches

Future of HPC

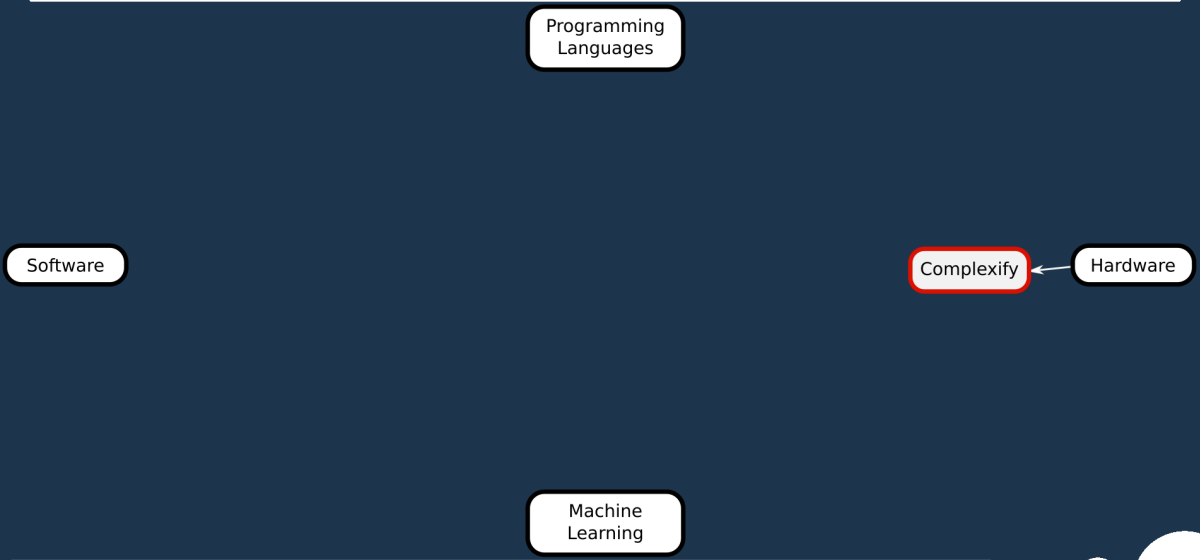
Programming Languages

Software

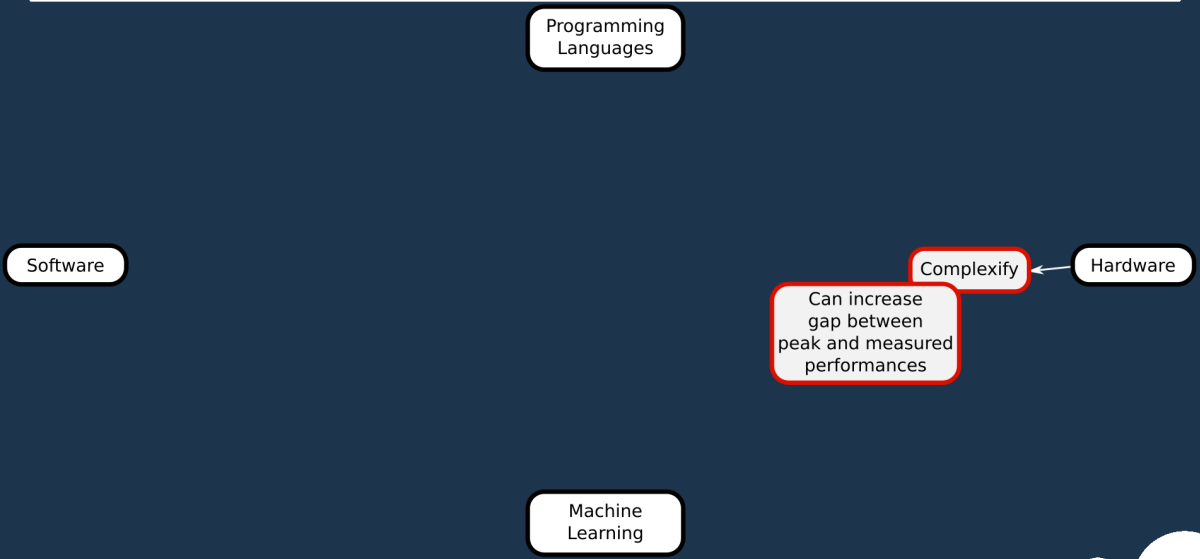
Hardware

Machine Learning

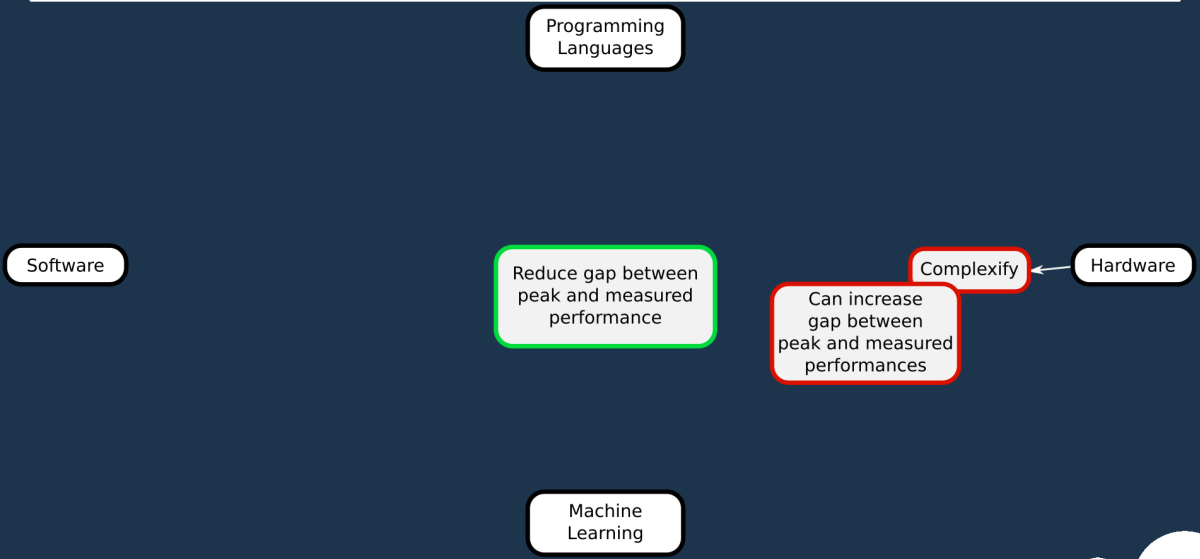
Future of HPC



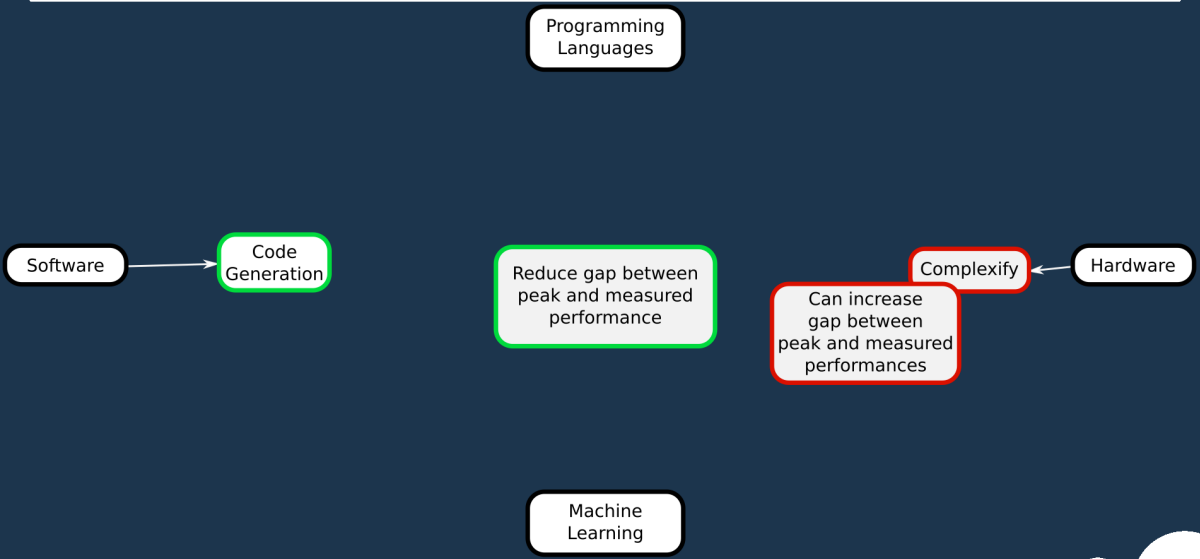
Future of HPC



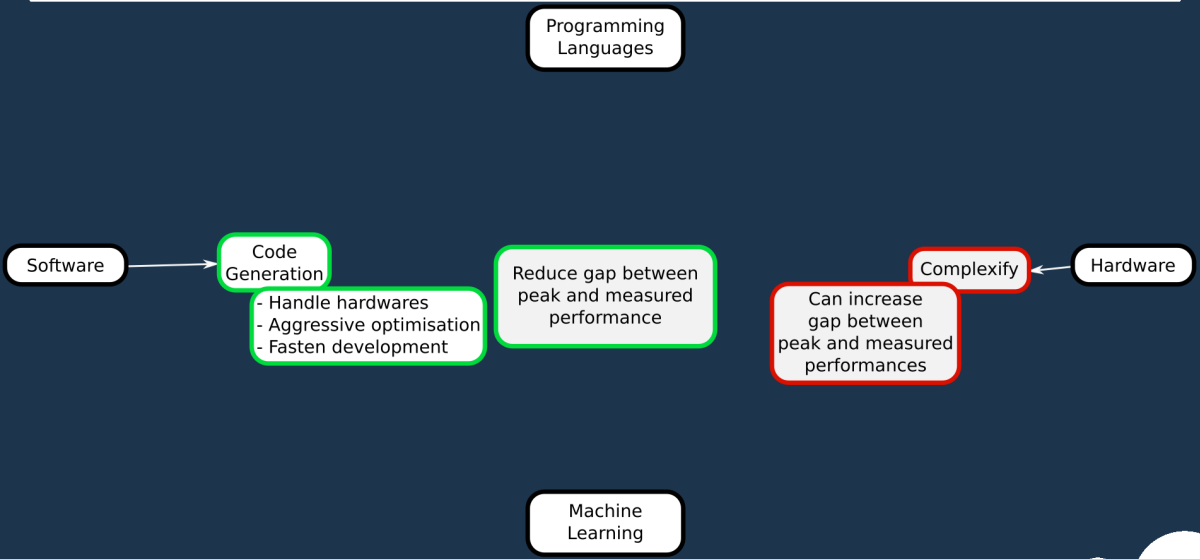
Future of HPC



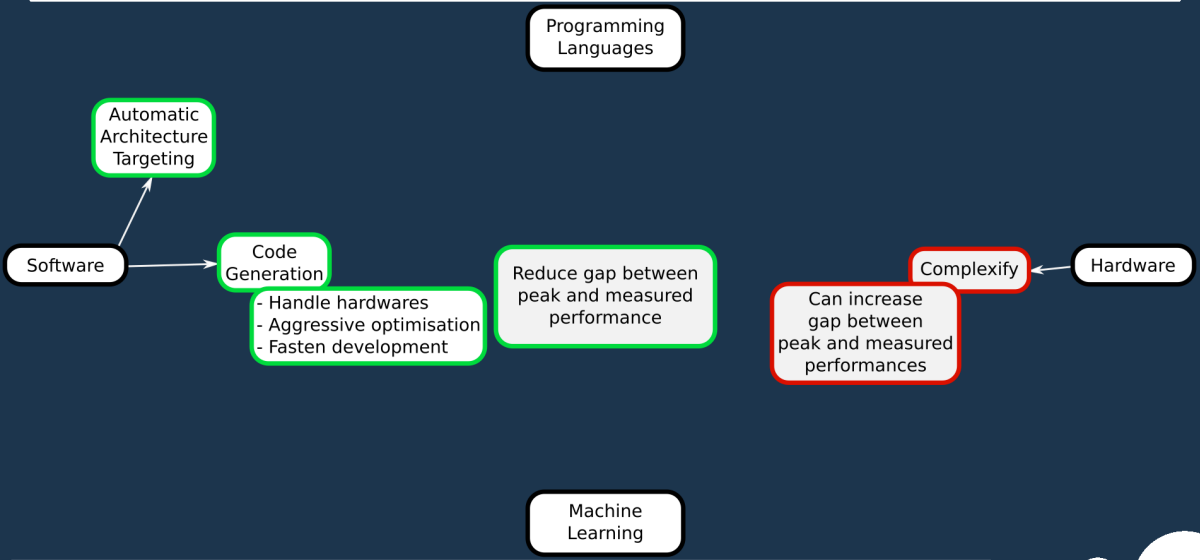
Future of HPC



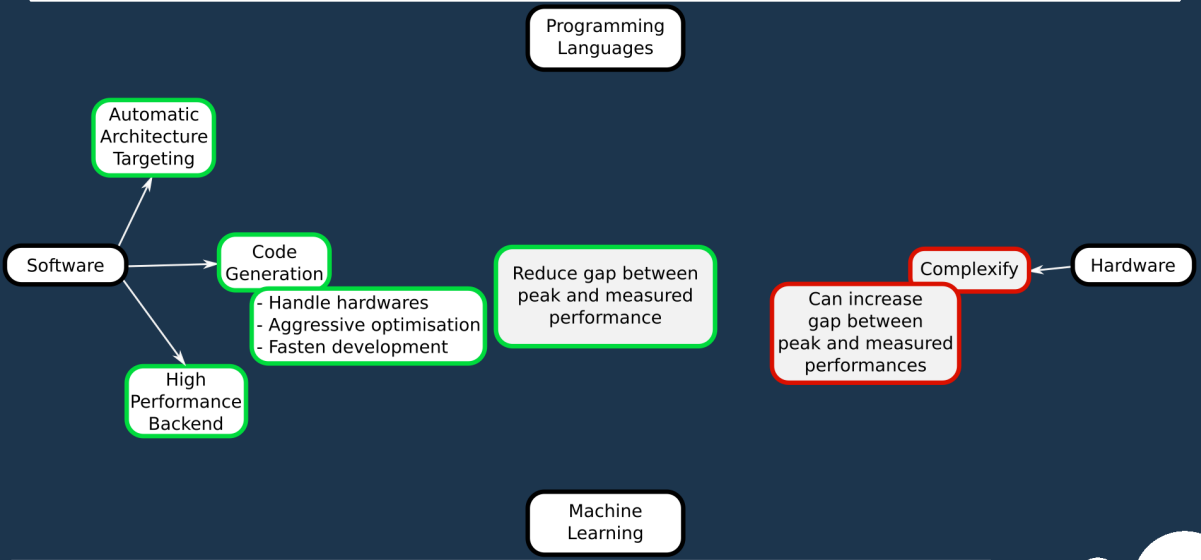
Future of HPC



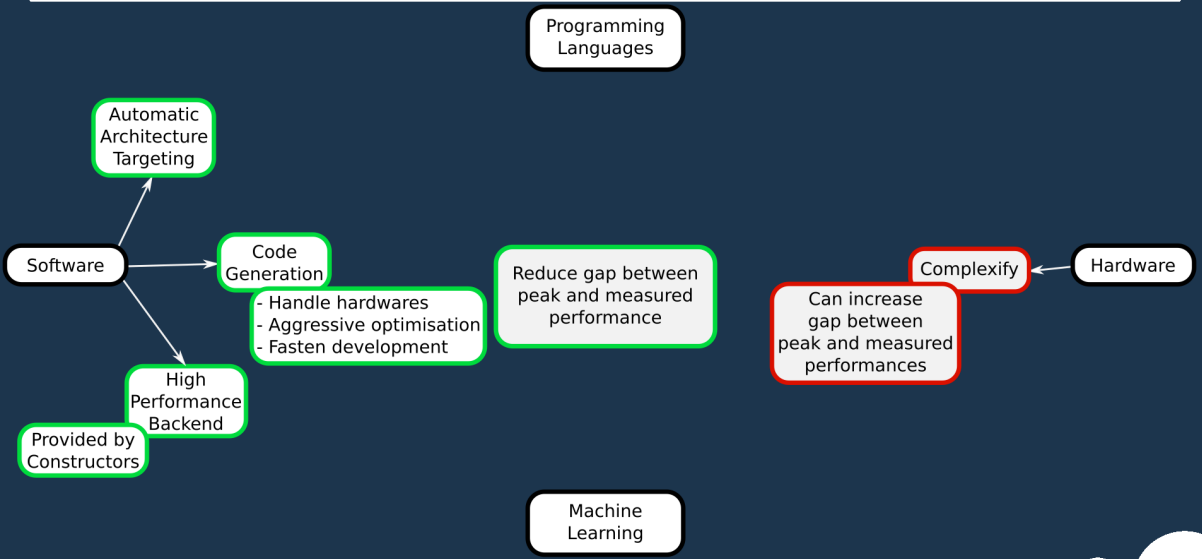
Future of HPC



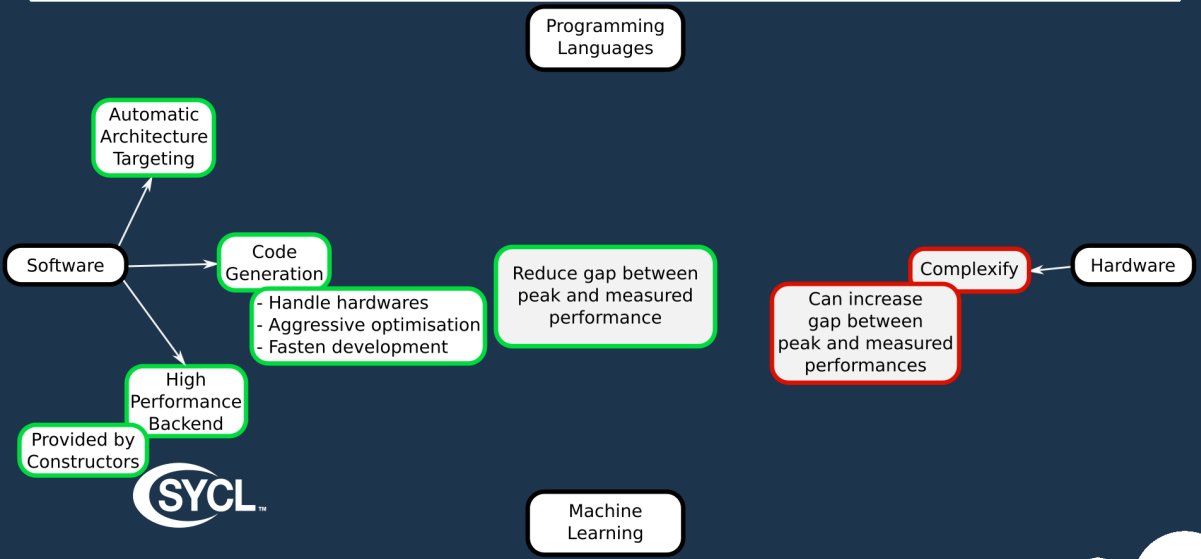
Future of HPC



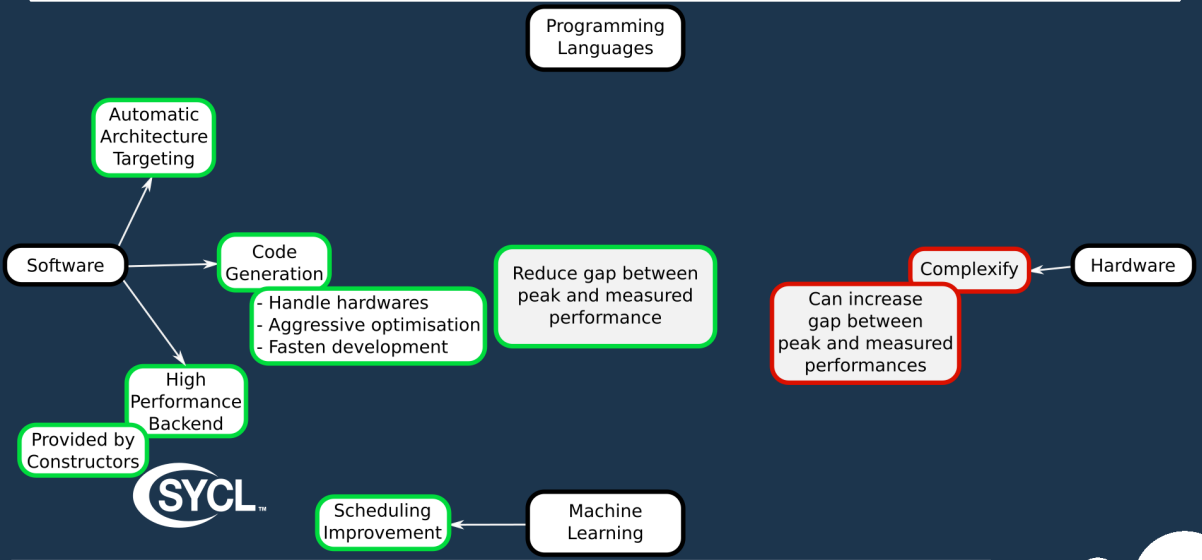
Future of HPC



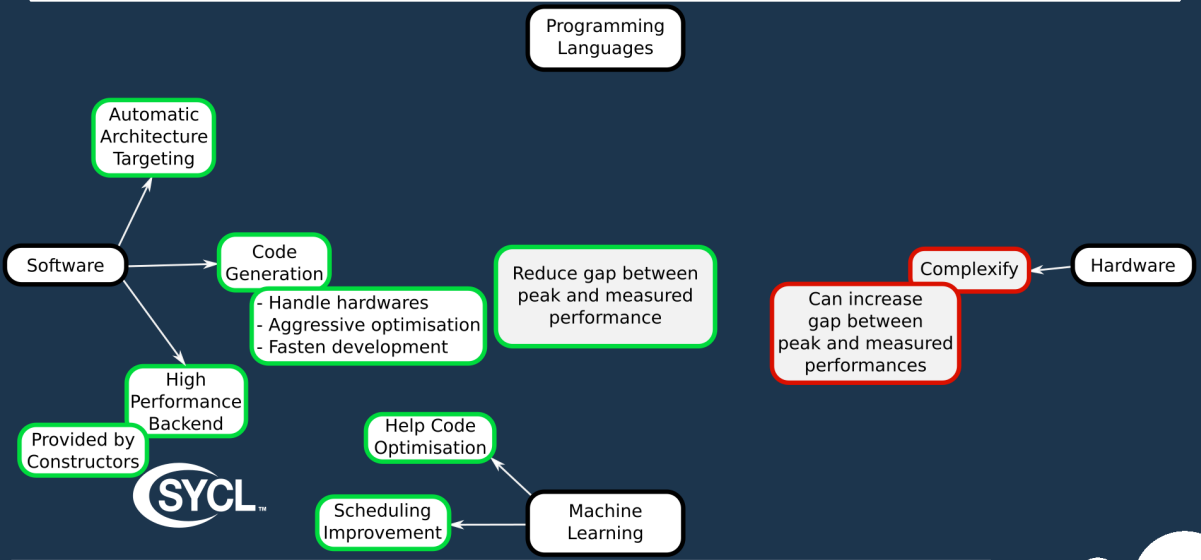
Future of HPC



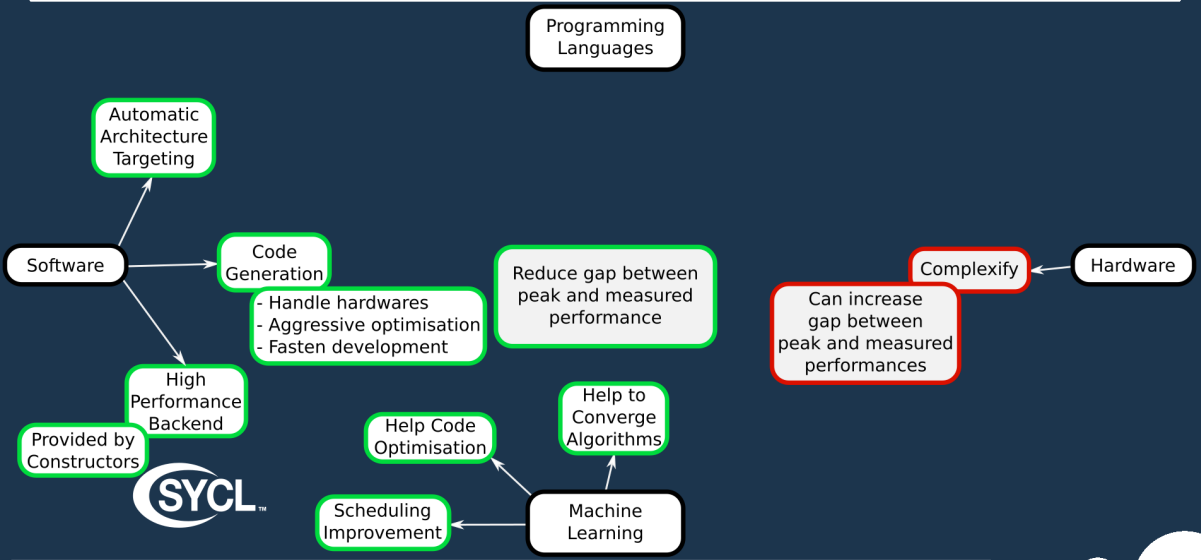
Future of HPC



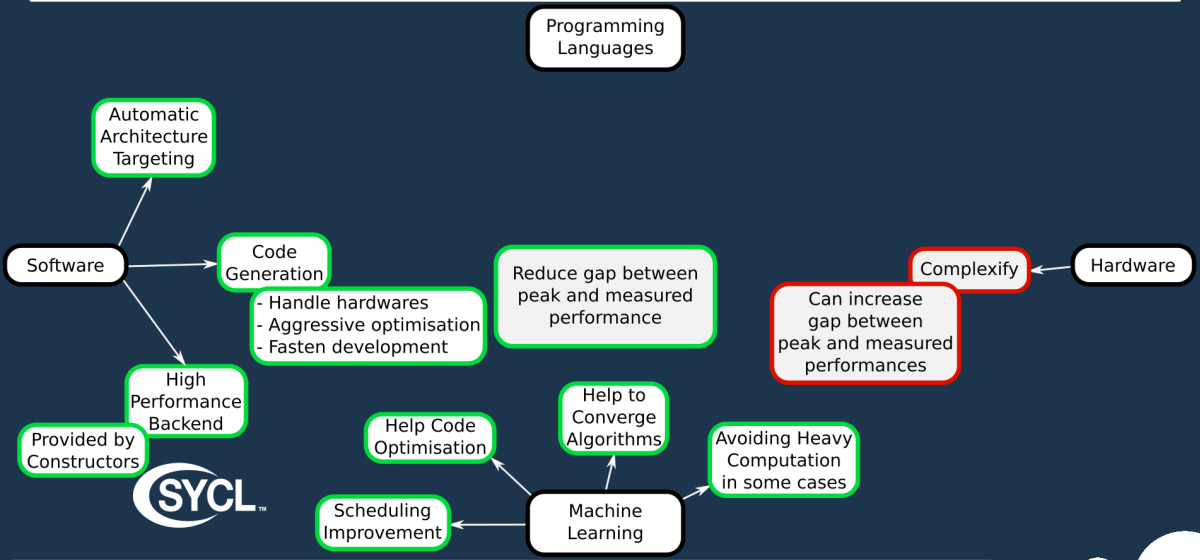
Future of HPC



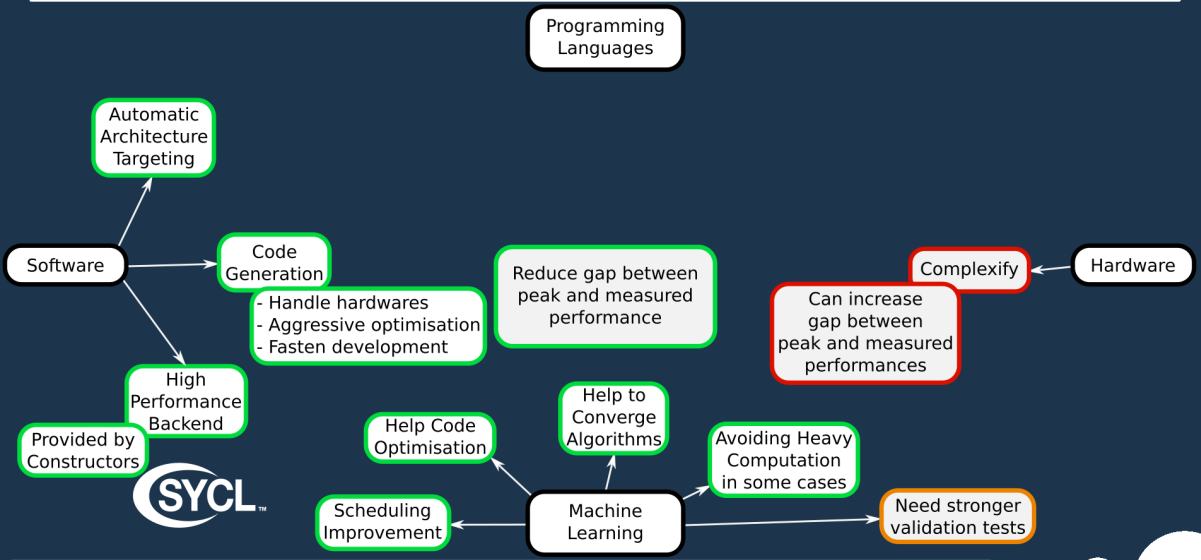
Future of HPC



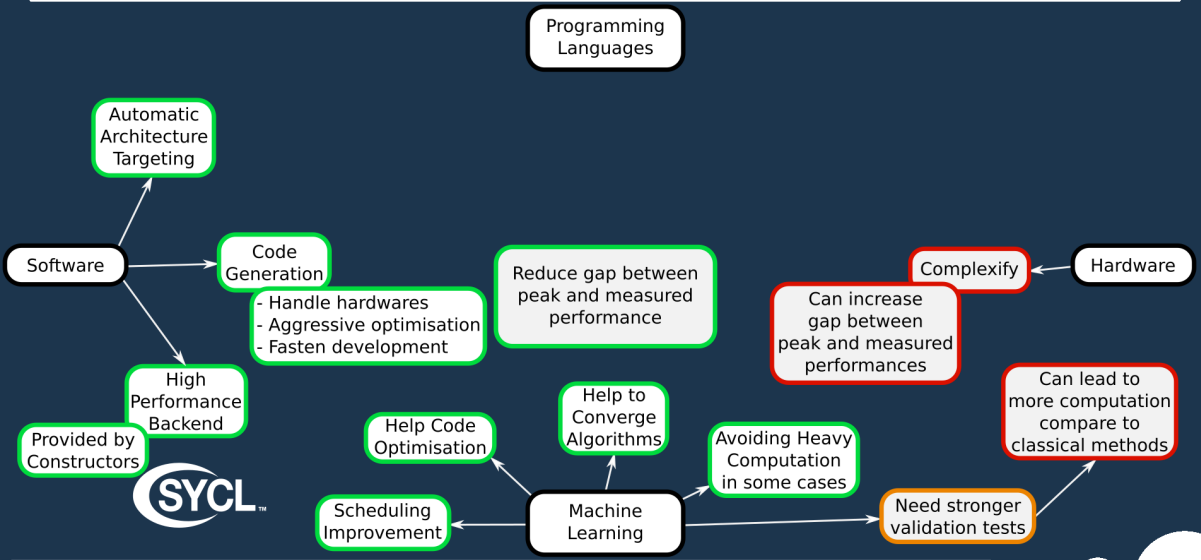
Future of HPC



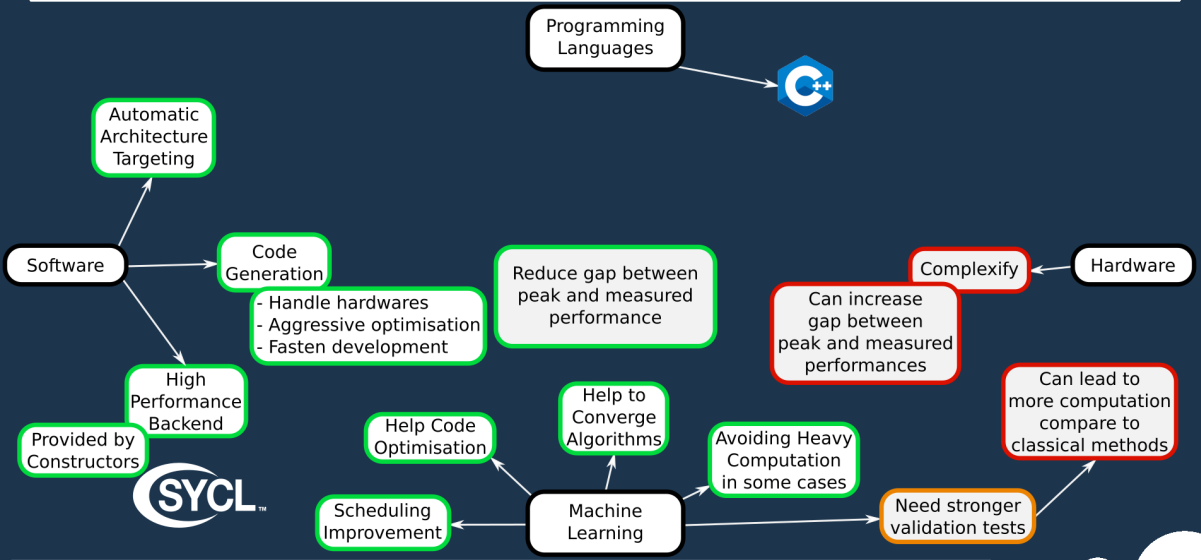
Future of HPC



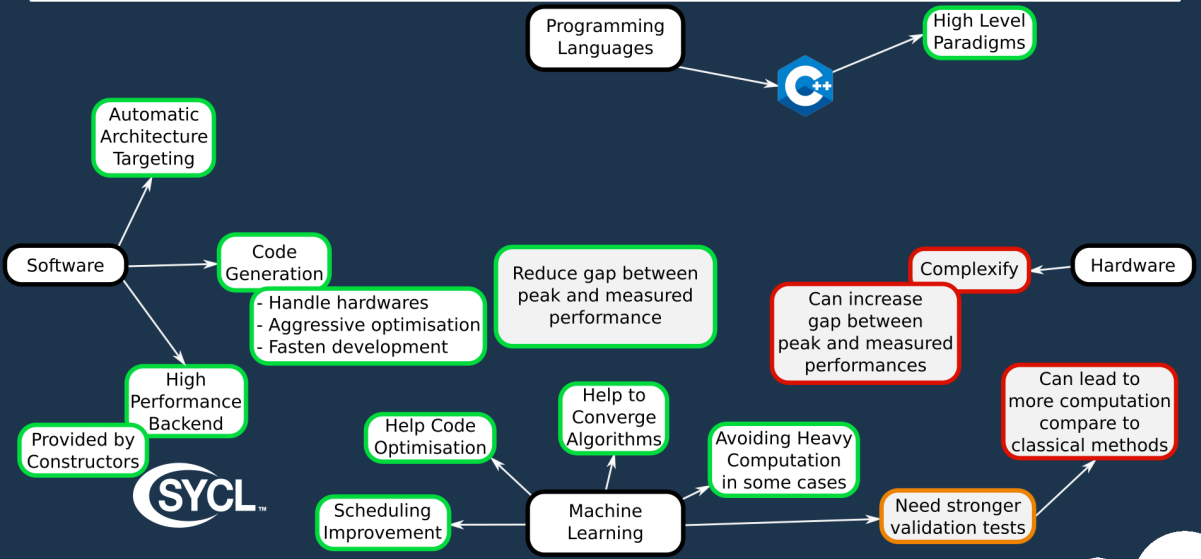
Future of HPC



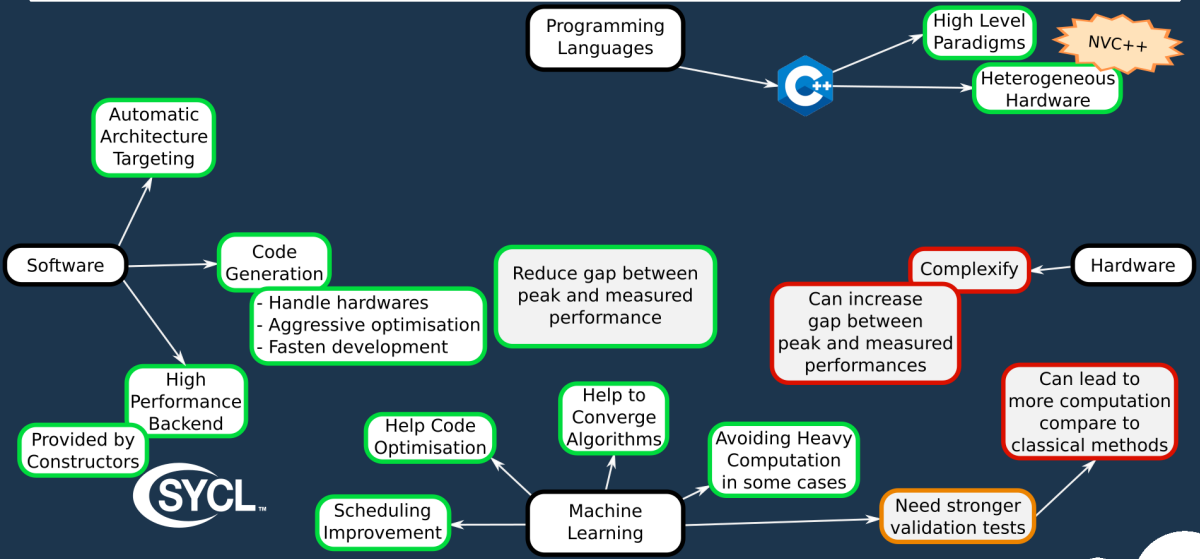
Future of HPC



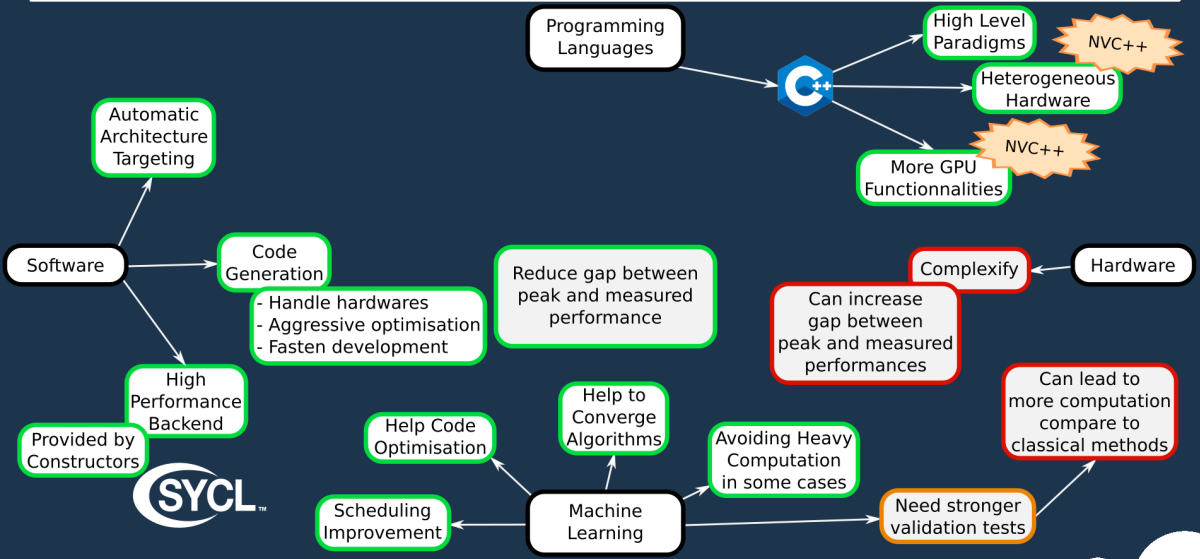
Future of HPC



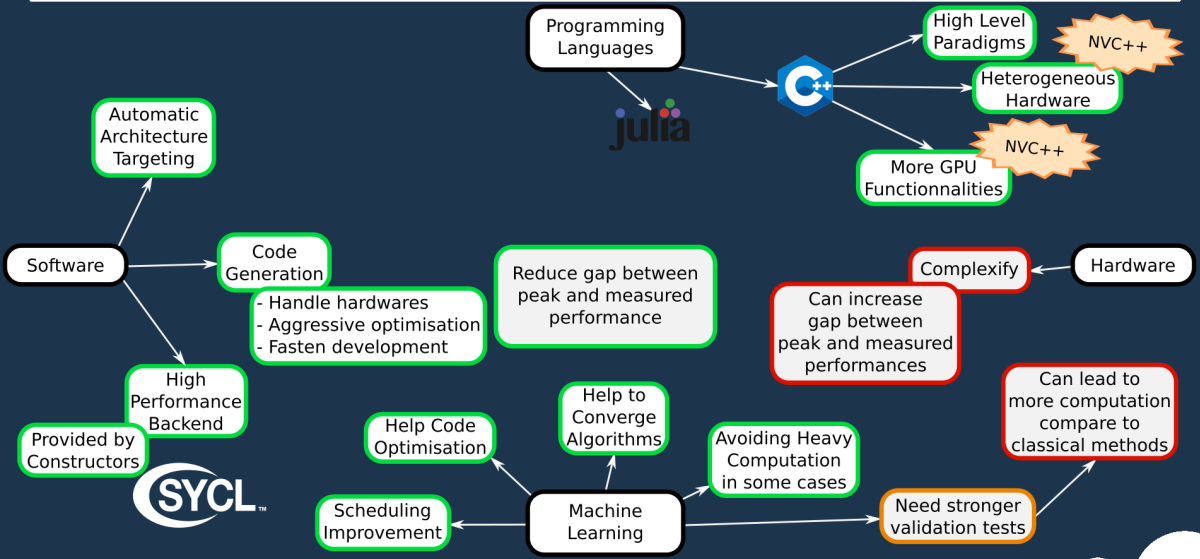
Future of HPC



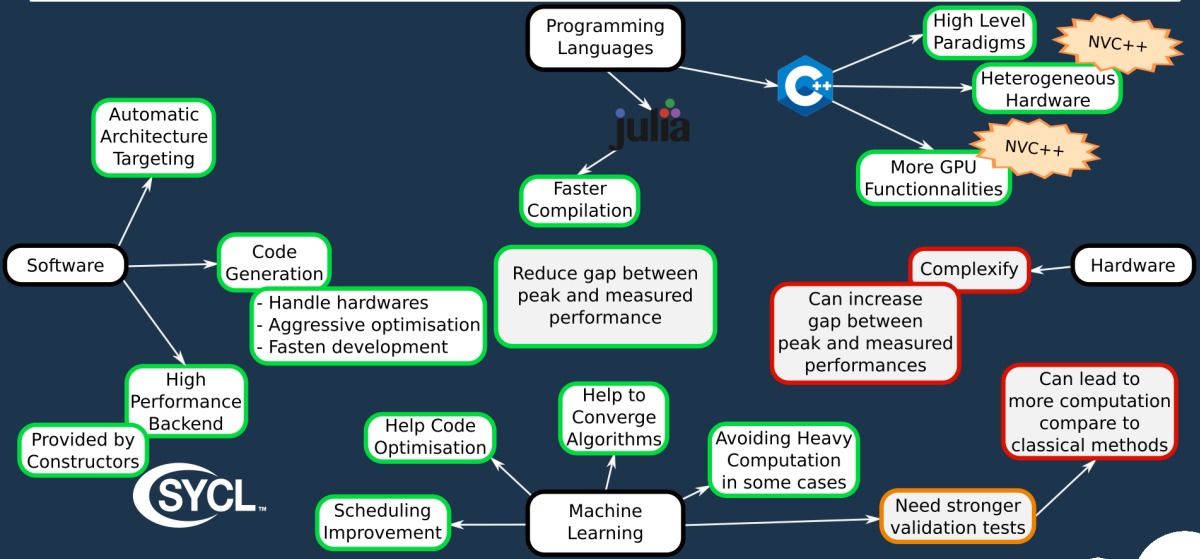
Future of HPC



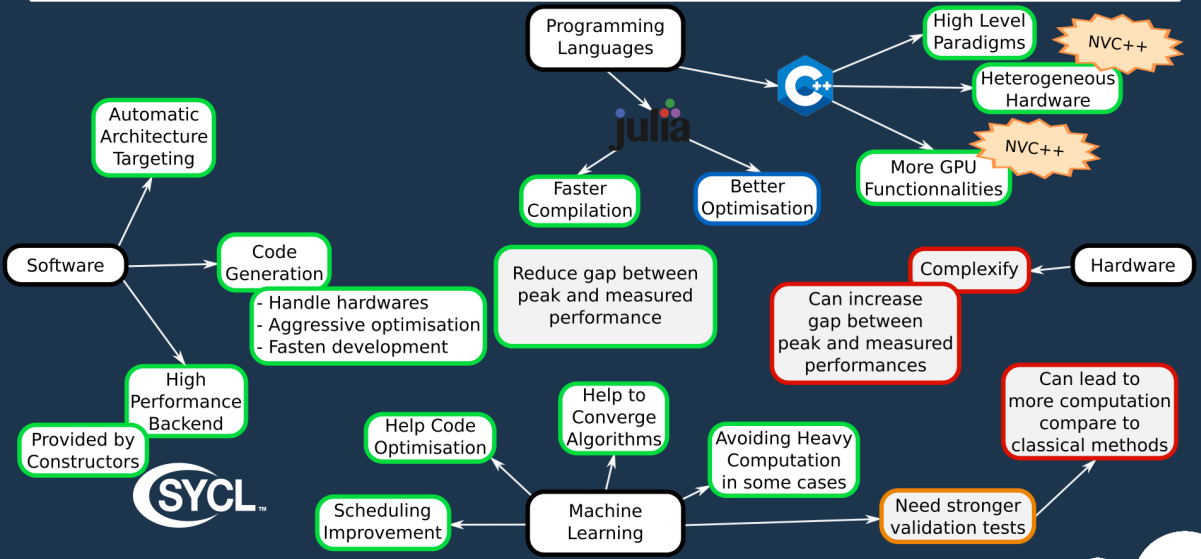
Future of HPC



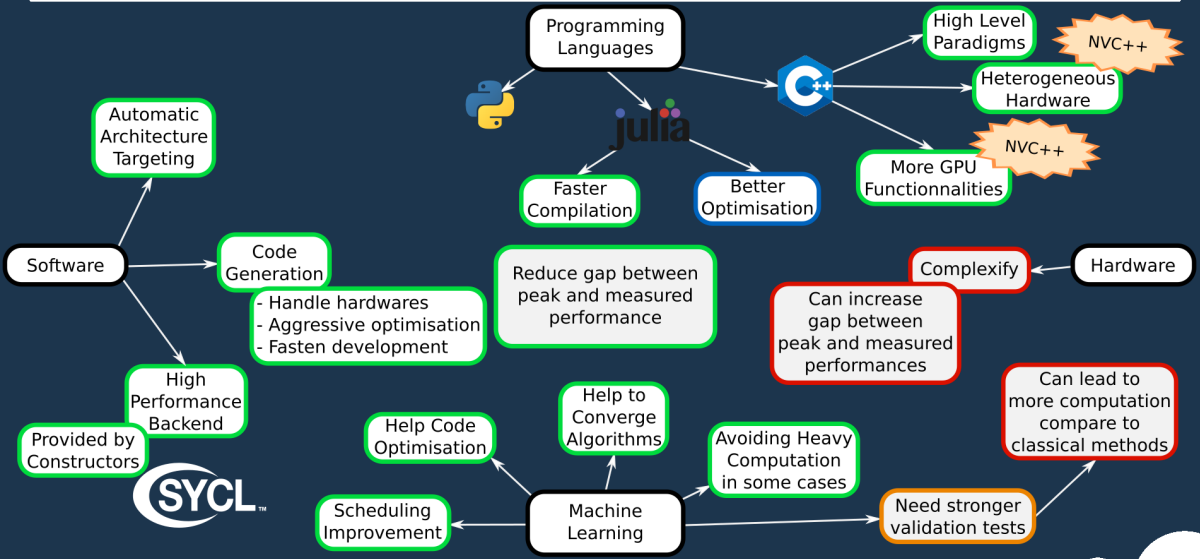
Future of HPC



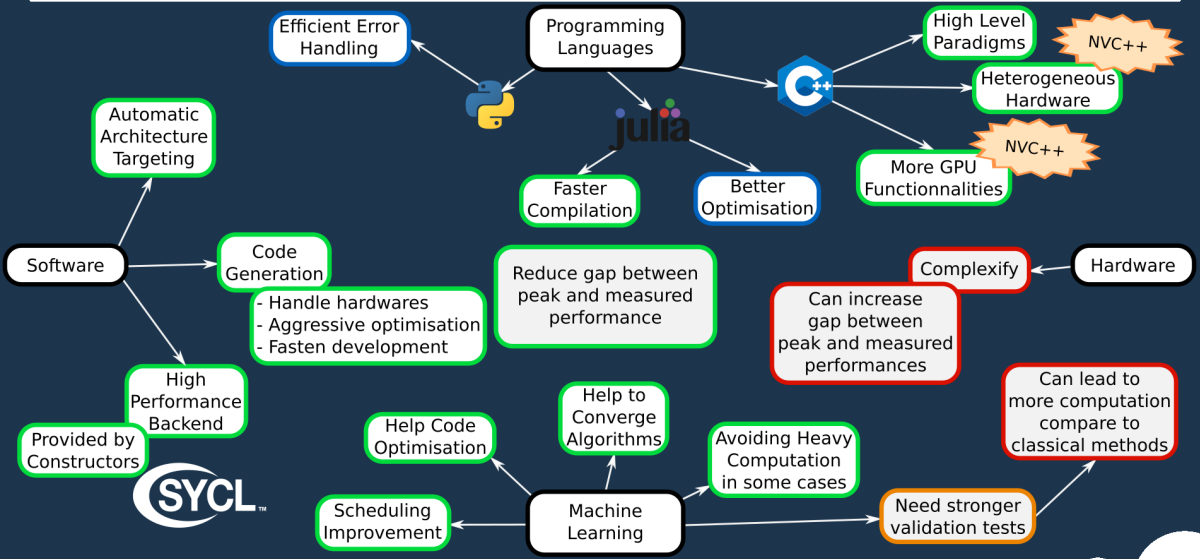
Future of HPC



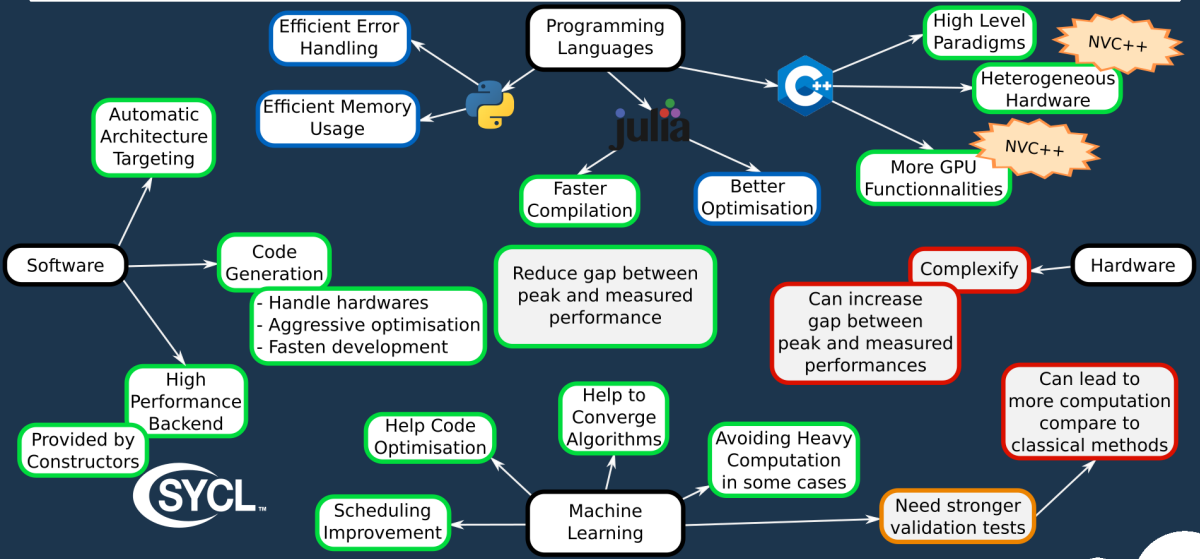
Future of HPC



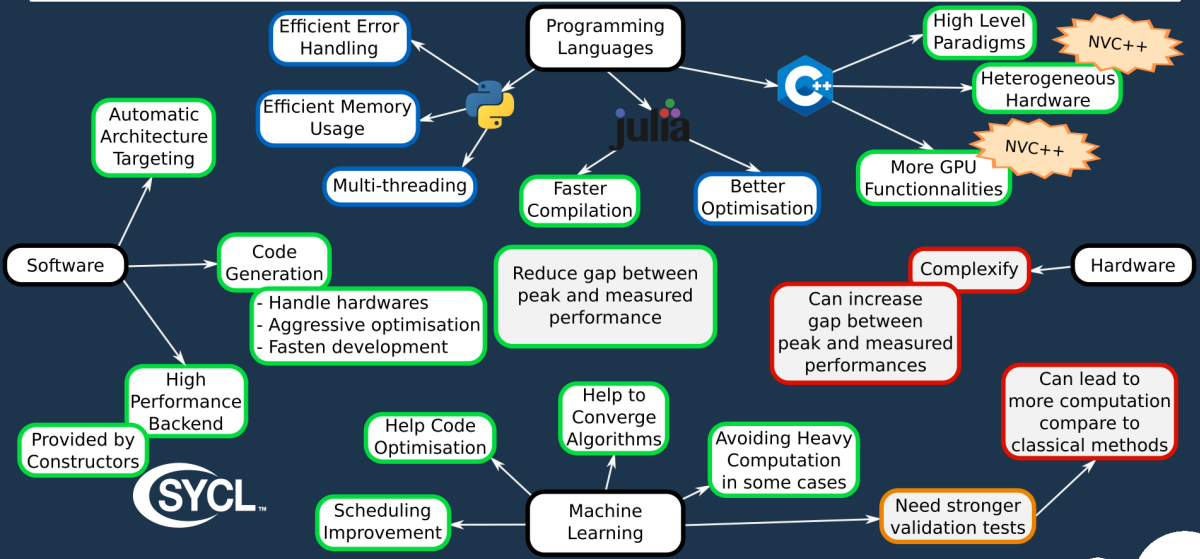
Future of HPC



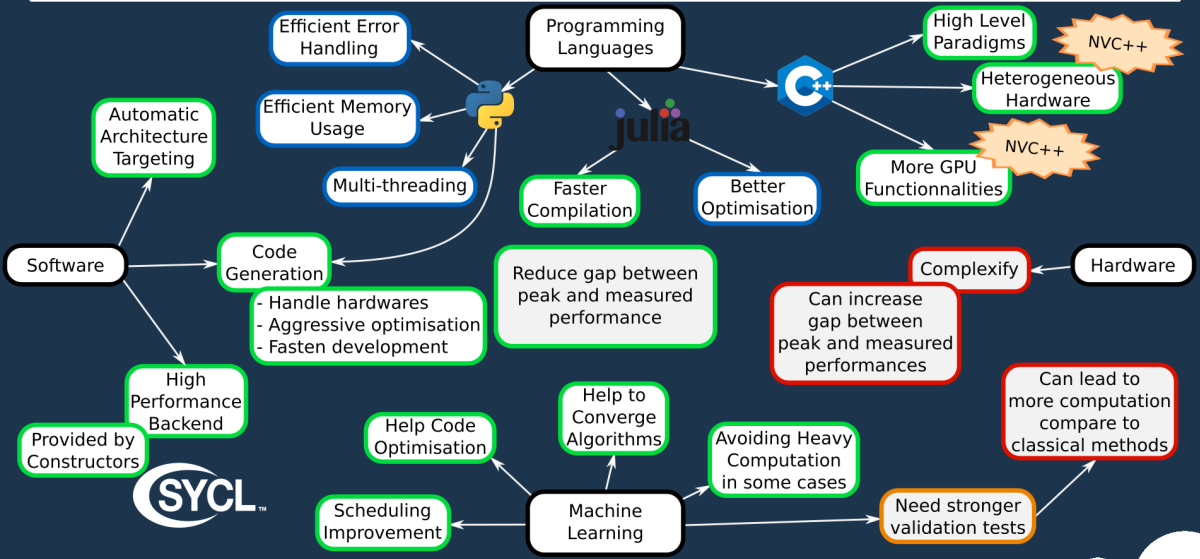
Future of HPC



Future of HPC



Future of HPC



Future of compilation stack

Goal

Time

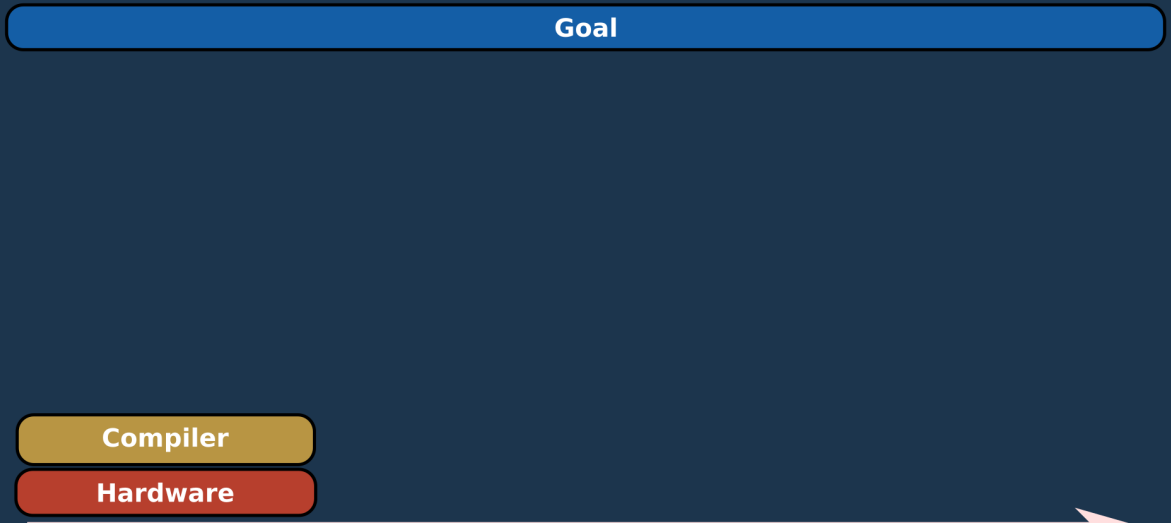
Future of compilation stack

Goal

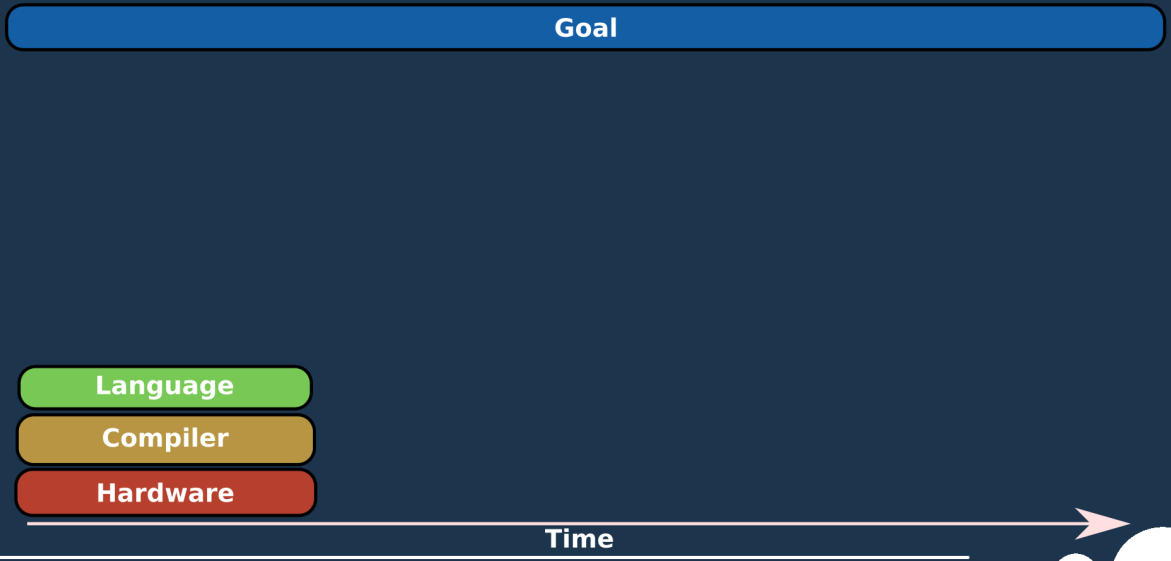
Hardware

Time

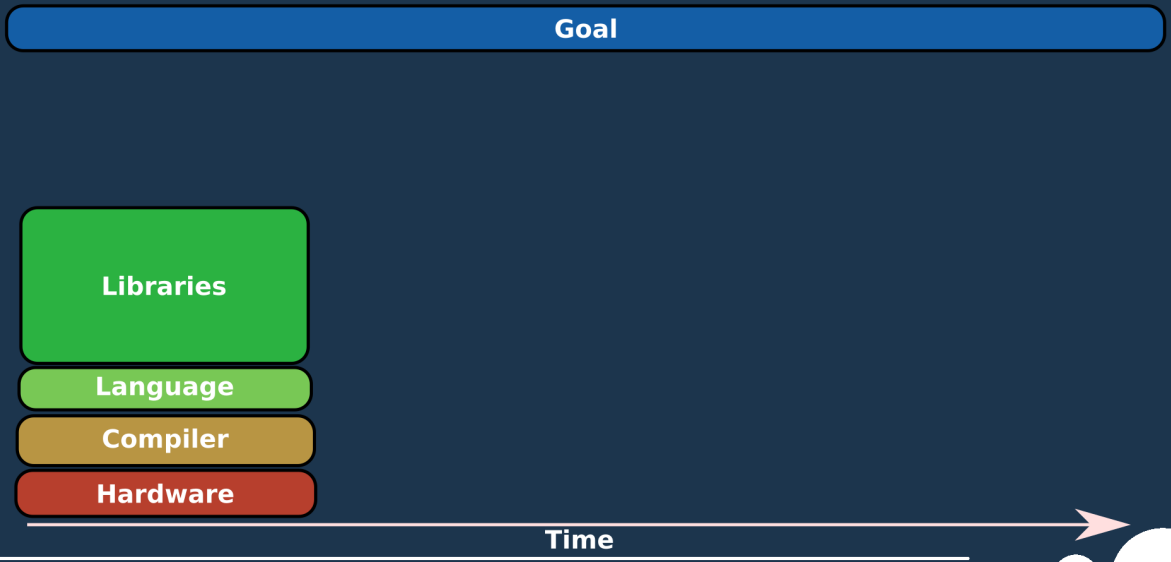
Future of compilation stack



Future of compilation stack

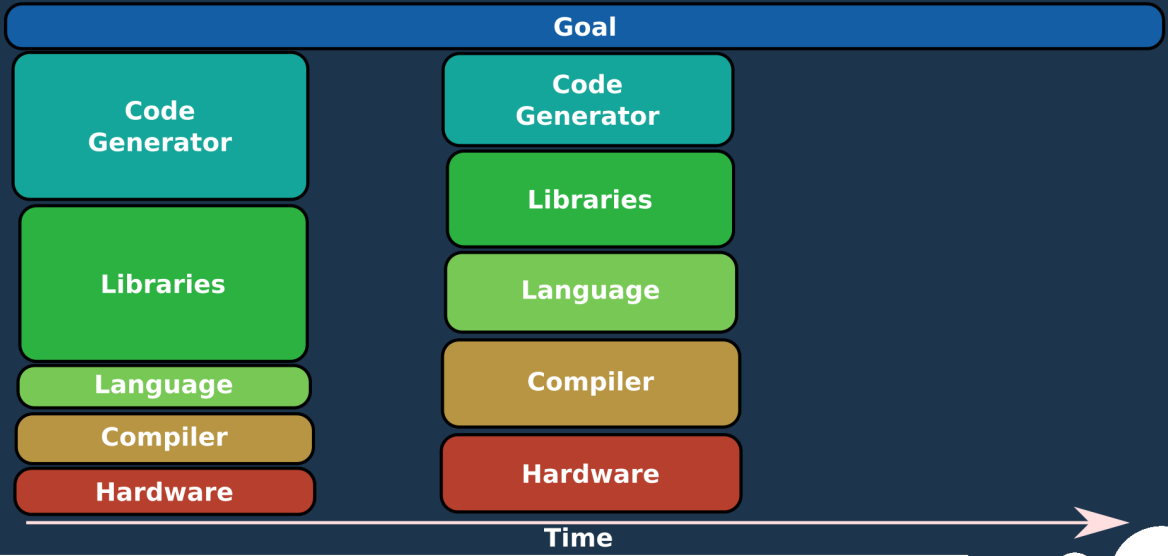


Future of compilation stack

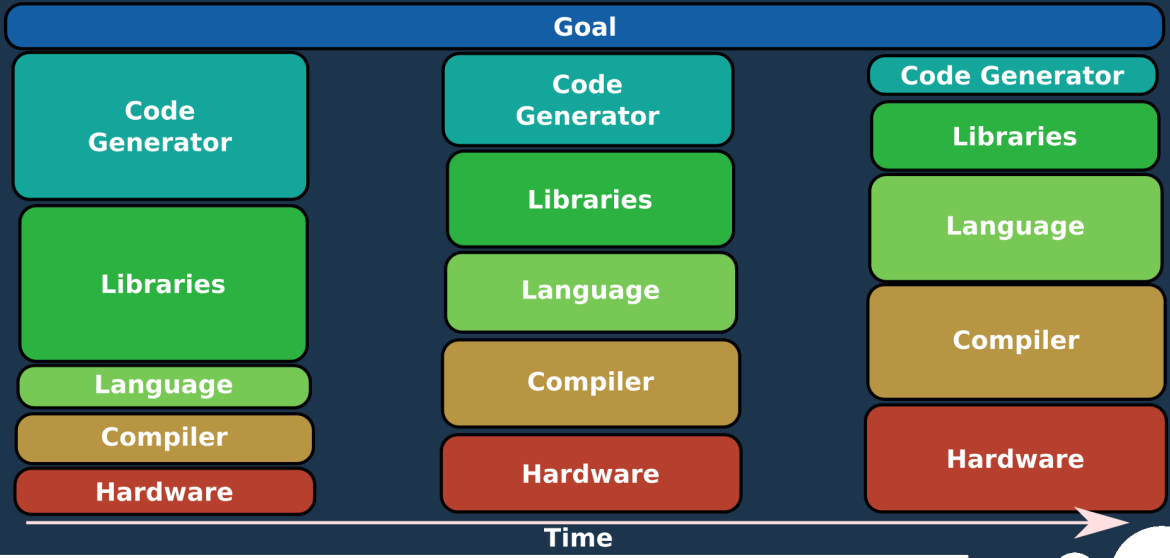


Future of compilation stack

Future of compilation stack



Future of compilation stack



HPC

- **Advancing Exascale: Faster, Smarter, and Greener! (Presented by Atos) [SS32805]**
  - From 0.23 GFlops/W in 2010 to 25 GFlops/W 2020 (**Juelich** supercomputer - 3 744 A100), free cooling
- **Overcoming HPC Application Communication Bottlenecks with Intelligent and Automatic Resource Selection [S31547]**
  - **NVTAGS** : speed up from 1.42x to 4x just with jobs placment
  - **NVTAGS doc**
- **Accelerating GPU-Enabled HPC and Data Science Applications with On-the-Fly Compression [S31664]**
- **Optimizing Communication on GPU-Based HPC Systems for Dask and cuML Using MVAPICH2-GDR [S31627]**
- **Simplify Integrating Large-Scale Hybrid CPU and GPU Systems to Accelerate HPC Workloads (Presented by QCT) [SS32967]**

Hardware

- **Go Beyond HPC: GPU Direct Storage, Parallel File Systems, and More (Presented by NetApp) [SS33181]**
  - Up to 17 GB/s in **NVME** read
- **Advancing Exascale: Faster, Smarter, and Greener! (Presented by Atos) [SS32805]**
  - **Classic** Data Center : 100 % computing, 100 % cooling
  - **Water cooling** Data Center : 100 % computing, 50 % cooling
  - **Direct liquid cooling** Data Center : 100 % computing, 5 % cooling

GPU

- **CUDA Code Optimization and Auto-Tuning Made Easy [S31429]**
  - **Kernel Tuner code generator**
- **RTCore for Compute: Exploiting Computational Patterns Using NVIDIA RTX [S31809]**
  - Particles simulation using **RTX** (noisy first 5 min)
- **What's New in OptiX [S31736] 55min**
- **Accelerating FFT toward Exascale Computing [P31661]**
- **Optimizing Lossless Compression Algorithms on the GPU [S32401]**
- **Accelerating GPU-Enabled HPC and Data Science Applications with On-the-Fly Compression [S31664]**
- **Fast and Simple Hash Tables [S31466] 38min**
- **GPU Accelerated Signal and Sensor Processing from Prototype to Deployment**

Compilers and languages

- **Inside NVC++ and NVFORTRAN [S31358] 33min**
  - **C++ 17** and **Fortran 202X** on GPU with **nvc++**
- **A Deep Dive into the Latest HPC Software [S31286]**
  - **Pragmas** for **OpenACC**
- **CUDA: New Features and Beyond**
  - **CUDA Stream** with **Asynchronous allocation**
  - **Parallel** functions loading

Machine Learning

- **The Energy-Based view of Self-Supervised Learning (Yann Lecun)**
  - Same results with 10 min **unsupervided** compare to 100 h of **supervised** for speach recognition
- **NVIDIA DALI: GPU-Powered Data Preprocessing**
  - Up to 96% of performance peak with **Triton**, and **DALI** even if data are compressed
- **GPU Performance for Recommendation DNNs**
- **The Challenges in Hardware Aware Inference Optimization for Deep Learning**
- **Fast and FuriousA GPU Inference Acceleration for State-of-the-Art YOLO v4 Model**
- **Starting a Deep Learning team with DGX Station A100 and Cnvrg.io Presented by Microway Inc.**

Cybersecurity + Machine Learning

- **Morpheus**
  - Spots sensitive data exchange (login, password, token)
- **Morpheus: AI Inferencing for Cybersecurity Pipelines [S33291]**
- **Building a GPU-Accelerated Cyber Flyaway Kit with NVIDIA Morpheus (Presented by Booz Allen Hamilton) [S32077]**
- **Cybersecurity is a Data Problem: Using NVIDIA Morpheus to Increase Network Visibility and Aid in Identifying Potential Network Problems [S32761]**

Link to a big summary

GTC 2021 notes from Reprises