



CFD Suite – HPC scalability report AI-ACCELERATED CFD

Performance analysis of byteLAKE's CFD Suite (Computational Fluid Dynamics accelerated with Artificial Intelligence), an HPC application, across various configurations, incl. single node (server) with 2 Intel CPUs and 2 NVIDIA GPUs, and many nodes of a CPU-only cluster (HPC). The report also summarizes CFD Suite's scalability with Intel technologies for AI models training and OpenVINO-optimized inferencing.

*Artificial
Intelligence*

*Machine
Learning*

Deep Learning

Computer Vision

Edge AI

*Cognitive
Automation*

RPA

HPC

FPGA / GPU



byteLAKE

Europe & USA

+48 508 091 885

+48 505 322 282

+1 650 735 2063

Technologies

byteLAKE's CFD Suite (AI-accelerated CFD)

CFD Suite is a collection of AI models (Artificial Intelligence) to significantly accelerate the execution of CFD simulations (Computational Fluid Dynamics).



CFD Suite is a data-driven solution designed to enable straightforward and simple integration with leading CFD CAE (Computer-Aided Design) software tools and workflows. Thanks to a close collaboration between byteLAKE and CFD software providers, CFD Suite can act as an add-on that allows engineers to ultimately cut the time to results of complex and expensive simulations. CFD Suite is also compatible with leading open-source CFD software. Hardware-wise, it is a cross-platform solution and supports both CPUs and GPUs with FPGAs being on the roadmap. Furthermore, CFD Suite addresses industries' requirements of supporting CPU-only architectures and therefore has been optimized in terms of performance and scalability for all Intel technologies.

It has been experimentally proven that the CFD Suite can reduce the time to results at least by a factor of 10x and keep the accuracy of predictions above 93%.

Key features include:

- **CFD solvers accelerated with AI / Artificial Intelligence**
(time to results significantly shorter & better speedup than with hardware accelerators)
- **Augmentation of numerical solvers** with complementary AI models
(faster analysis, reduced cost of trial & error experiments, 100% compatible with existing tools)
- **Solver specific AI models for Digital Twin industrial scenarios**
(generalized AI models for different use cases, geometries, and parameters' values)
- **Designed to scale up fast**
(solver specific AI model within 2-4wks vs. 1-2yrs efforts for hardware adaptations)

From simulations to predictions

Typically, it takes 3 simple steps to get started with CFD Suite:

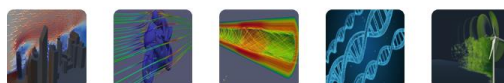
1. **CFD Solver selection:** pick the one you'd like to accelerate
 - a) Typical candidate: simulations that take too long
2. **Setting the targets for AI Model(s)**
 - a) What level of acceleration do you expect? Reduce time/iterations? Anything else?
 - b) What is the accepted accuracy? Within 85-95%? Higher? Lower?
 - c) Should AI generate only the result of the simulation or intermediate steps as well?
 - d) Consultancy to assess the solver's input/output (types, ranges, etc.)
3. **AI Model training**
 - a) Based on the above, byteLAKE will train the CFD Suite to accelerate your CFD Solver(s)
 - b) CFD Suite will be 100% compatible with your workflow
= no need for any changes in infrastructure or data formats/types

New AI models are constantly added to the collection which gradually increases the number of CFD simulations that can be handled by the CFD Suite off-the-shelf. The roadmap below shows the direction of the CFD Suite development:

byteLAKE's CFD Suite

Roadmap highlights

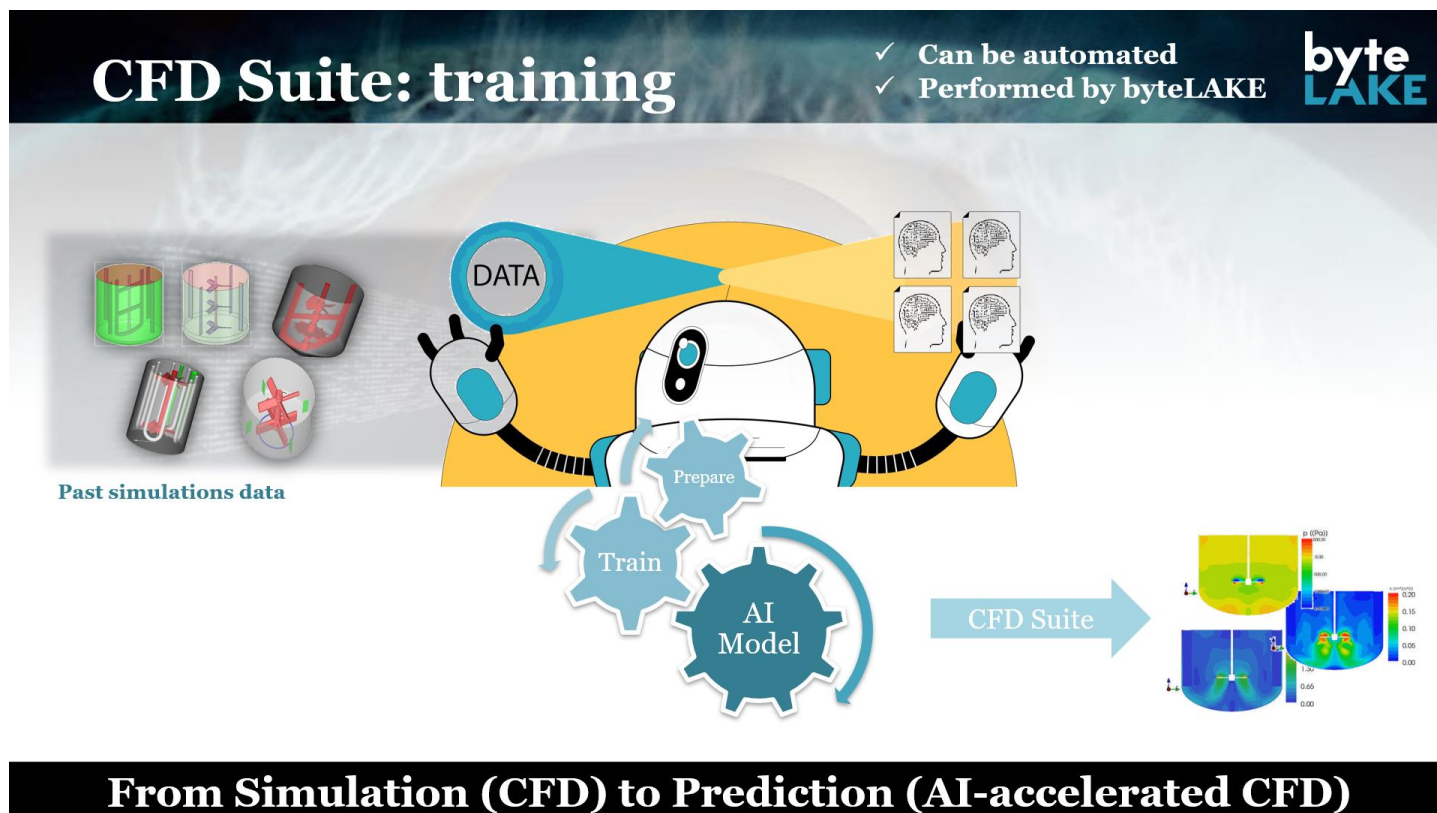
- **CFD Suite framework**
 - 2D, 3D domains
 - Steady-state simulations
 - AI models generalization
- **Solver specific AI models**
 - Chemicals mixing
 - Flow around buildings
- **Accelerate solvers from leading CFD Tools incl. open source**
 - Compatibility of input/output
- **Beyond one-node (HPC)**
 - Large simulations: >2M cells, 5M, >10M
- **Integration with partners' CAE / CFD tools**
 - Tridiagonal Solutions' [MixIT](#)
 - others
- **Beyond on-premise**
 - CFD Suite as a Service via Cloud providers
- **Transient / unsteady simulations**
- **AI Models generator**
- **byteLAKE CFD Suite unleashed**
 - Growing collection of innovative AI Models for CFD across industries
- **Channel partnerships**



CFD Suite

Training

CFD Suite deployment starts with the AI model(s) training. Therefore, **past CFD simulations are required (usually between 8-50 such simulations)** to produce accurate predictions and ensure AI models proper generalization level. Generalization enables predictions across a wide range of various input parameter values incl. geometries.



Inferencing (predictions)

Once the AI models are trained, they can be run as part of the CFD CAE software in a form of an add-on. As the current version of the CFD Suite (March 2021) is data-driven, the prediction is organized alongside these steps:

1. First, the initial part (i.e., 3-10%) of the simulation is done with a CFD solver
2. Then the CFD Suite takes over the results and its AI model(s) predict the results, accelerating time to results
3. Note that CFD Suite can work with 3D data, producing 100% compatible results with the client's existing CFD software tools. Therefore, results predicted by CFD Suite can be taken for further processing and analysis right away and no further formatting is necessary.

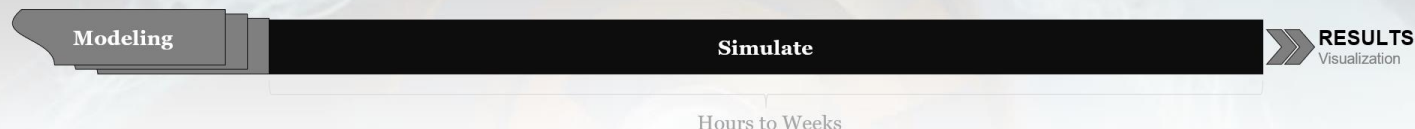


AI-powered CFD predictions

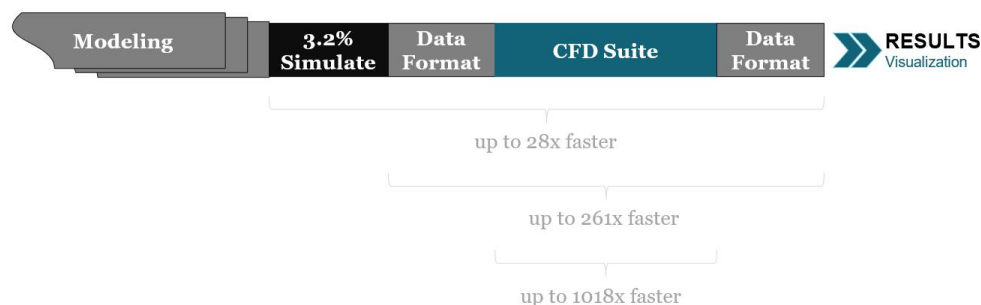
CFD Suite AI models are produced per solver and can accurately predict results for a family of simulations. Once the AI models have been trained, the users can freely modify input parameters and still get accurate and fast results. Breaking down the inferencing (prediction) part into steps leads to the following structure of the prediction process:

Early benchmark (FAST)

- **Traditional workflow**



- **CFD Suite (mode: fast)**



example time-to-results:

~9 mins

vs 4hrs

AI models can work in 3 modes:

- ACCURATE - focused on the accuracy of the predictions,
- MEDIUM - offering a trade-off between accuracy and acceleration
- FAST - maximizing reduction of time to results.

Currently, steady-state CFD simulations are supported with transient being on the roadmap.

In the example presented above (FAST mode), the process requires 3.2% of a conventional CFD simulation done by a CFD solver. Afterward, CFD Suite takes over the results, does the data formatting for AI prediction purposes, predicts the outcome of the simulation, and re-calculates the data back to the configured formats. Overall, the acceleration is about getting the results at least 28 times faster for the FAST mode.

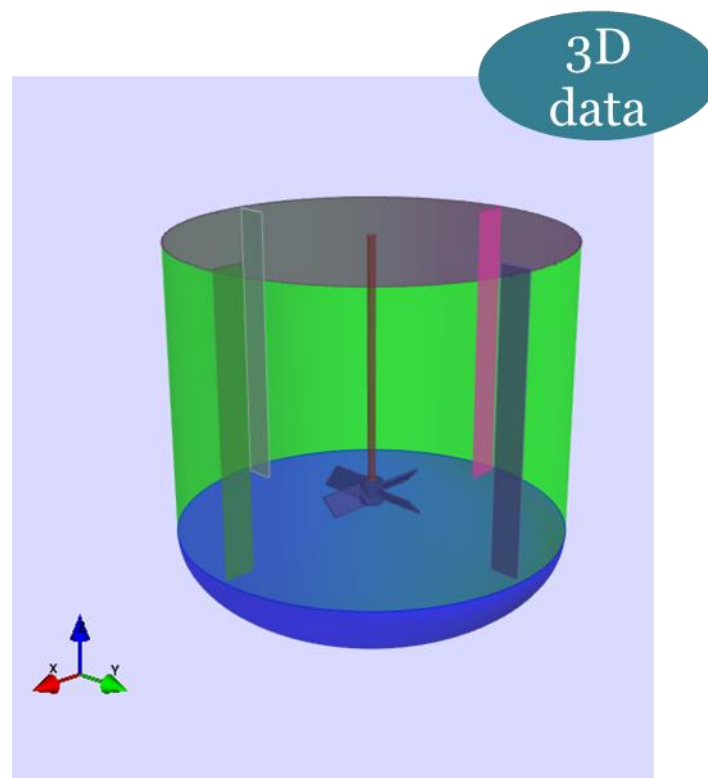
On the inside, CFD Suite is based on the reduced ResNet network organized as residual blocks. This is combined with byteLAKE's proprietary AI models. And summing up, CFD Suite delivers:

- **AI Model(s) compatible with your existing workflow**
 - No need to change any data types, replace toolchains, etc.
 - No hardware/infrastructure upgrades or changes needed
 - Cross-platform compatibility (CPU-only, CPU+GPU, PC/ data center/ cloud, etc.)
- **AI Model(s) easy to use**
 - Highly compatible solution ready to take data as CFD Solvers do
 - Easy adaptation. Working the same way, the standard CFD Solvers do
 - Minimum 10x faster results comparing to CFD Solvers
 - Highly adaptive solution ready to follow changes in CFD Solvers
- **AI Model(s) enabling new possibilities**
 - Much faster simulations
 - Significantly reduced cost of trial & error experiments

Example case study: chemical mixing

The goal was to compute the converge state of the liquid mixture in a tank equipped with a single impeller and a set of baffles. **Based on different settings of the input parameters, the simulation predicted the following set of quantities**

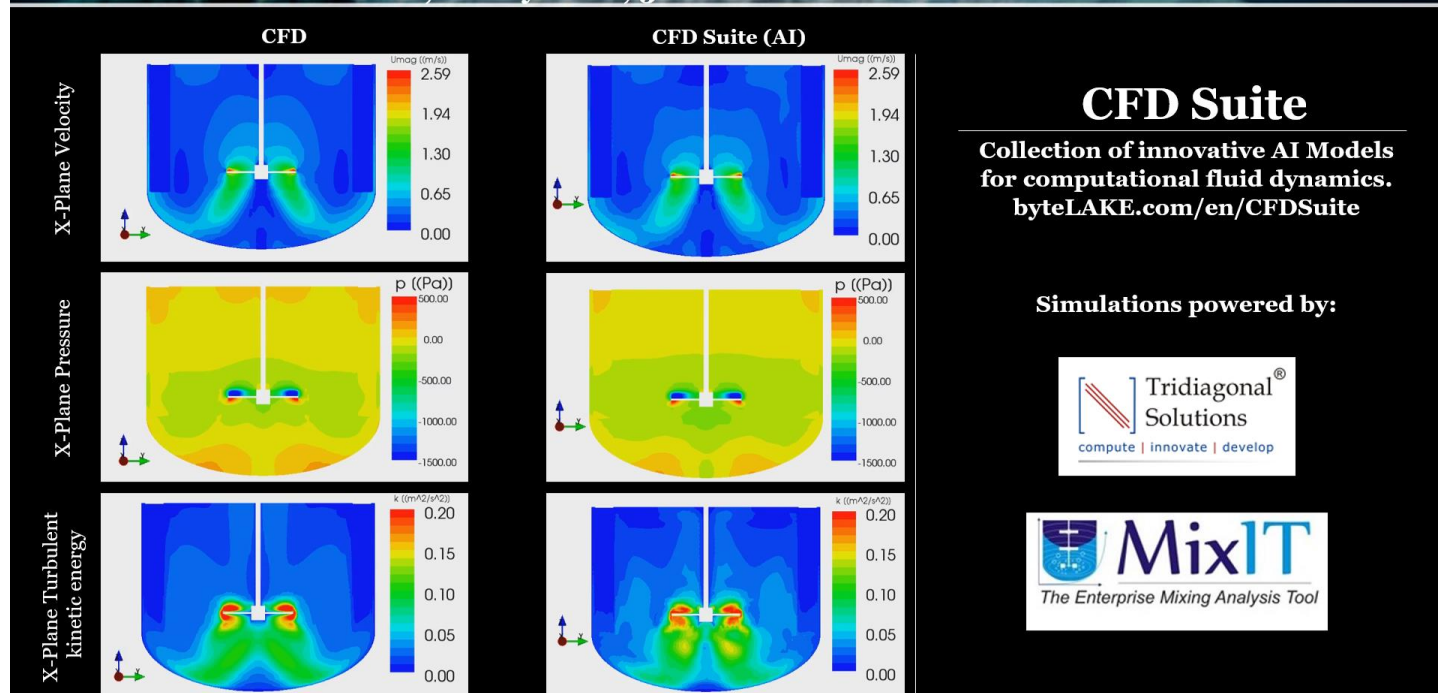
- the velocity vector field (U)
- pressure scalar field (p)
- turbulent kinetic energy (k) of the substance
- turbulent dynamic viscosity (μ_t)
- turbulent kinetic energy dissipation rate (ϵ)



Here are the example results for the chemical mixing case. We used a mesh size of <2M cells and the simulation required 5,000 (five thousand) steps to converge into a final state (steady-state simulation). We used a tool called MixIT, provided by Tridiagonal Solutions to generate the results headlined as “CFD”. The results visible on the right-hand side (headlined as “CFD Suite (AI)”) have been generated by byteLAKE’s CFD Suite using the ACCURATE mode.

Chemical mixing accelerated with AI

Simulation: <2M cells, steady-state, 5K iterations



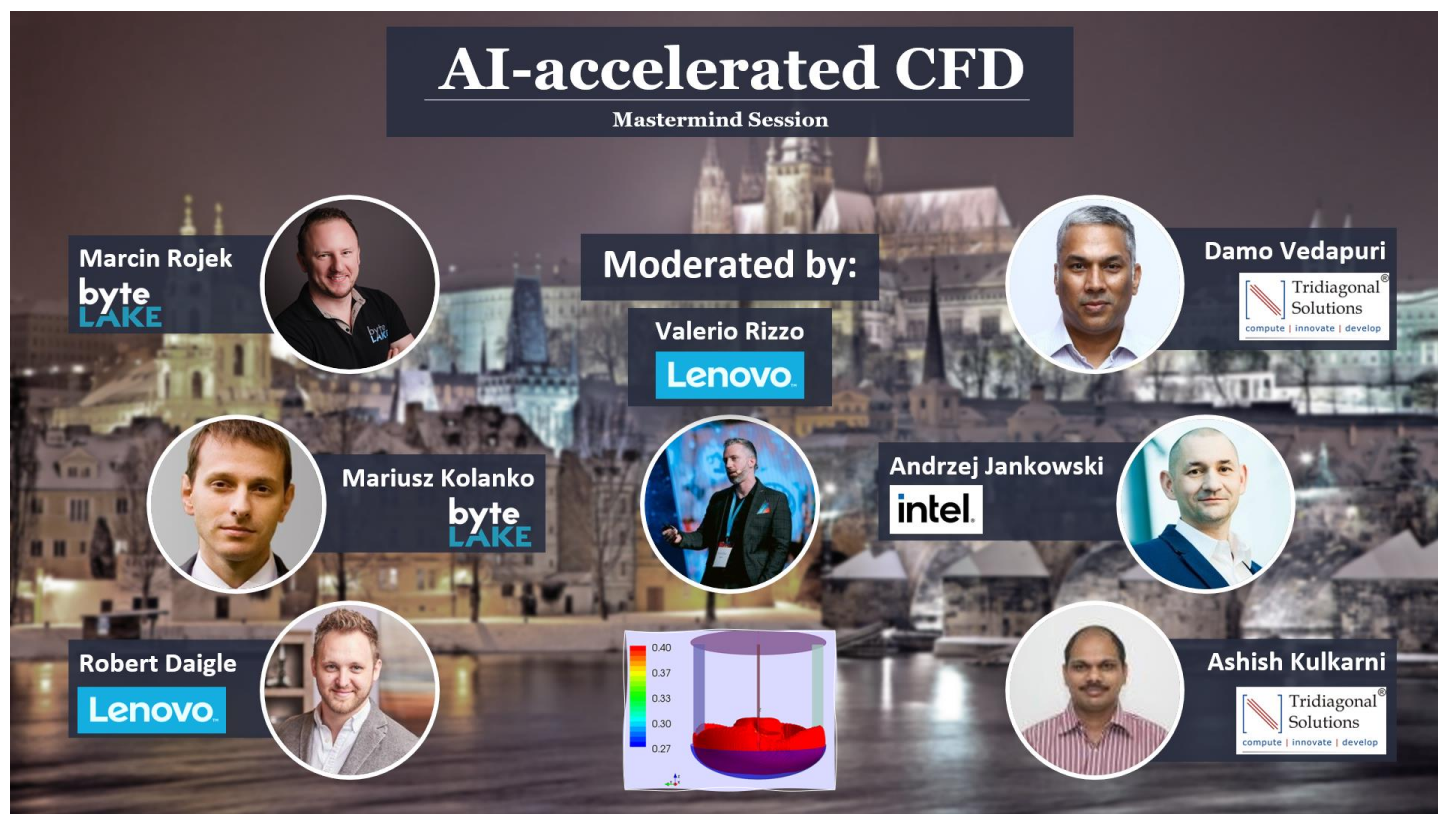
In that case, further analysis of the results led us to the following conclusions:

Parameter	CFD	CFD Suite (AI)	
		Value	% Deviation
Velocity magnitude	0.3103384	0.30982	0.16
Imp-1 Pressure Torque	11.81192	11.97289	1.4
k	0.02946442	0.029898	1.5
Epsilon	0.05131064	0.0509264	0.7

Quantity	Pearson's c.	Spearman's c.	RMSE	Histogram comp. [%]
U	0.990	0.935	0.016	89.1
p	0.993	0.929	0.004	90.1
epsilon	0.983	0.973	0.023	90.3
k	0.943	0.934	0.036	99.4
mut	0.937	0.919	0.147	93.5
Average	0.969	0.938	0.045	92.5

Learn more

Learn more about CFD Suite by visiting our website www.byteLAKE.com/en/CFDSuite. We constantly update our product with new features so follow our blog post series where we share the latest updates. Also, consider joining dedicated LinkedIn and Facebook groups to stay in touch with the CFD Suite community. You might as well be interested in listening to our panel discussion about how CFD Suite has been successfully used in the CFD/chemical mixing space.



AI-accelerated CFD
Mastermind Session

Moderated by:
Valerio Rizzo
Lenovo

Panelists:

- Marcin Rojek
byte LAKE
- Mariusz Kolanko
byte LAKE
- Robert Daigle
Lenovo
- Damo Vedapuri
Tridiagonal Solutions
compute | innovate | develop
- Andrzej Jankowski
intel
- Ashish Kulkarni
Tridiagonal Solutions
compute | innovate | develop

The graphic features a background image of a city at night. In the center, there is a circular inset showing a man (Valerio Rizzo) speaking. Below this, there is a small image of a CFD simulation showing a red and purple fluid mixture in a cylindrical container with a stirrer, accompanied by a color scale from 0.27 to 0.40.

Reach out to us to get access to the on-demand recording (CFDSuite@byteLAKE.com).

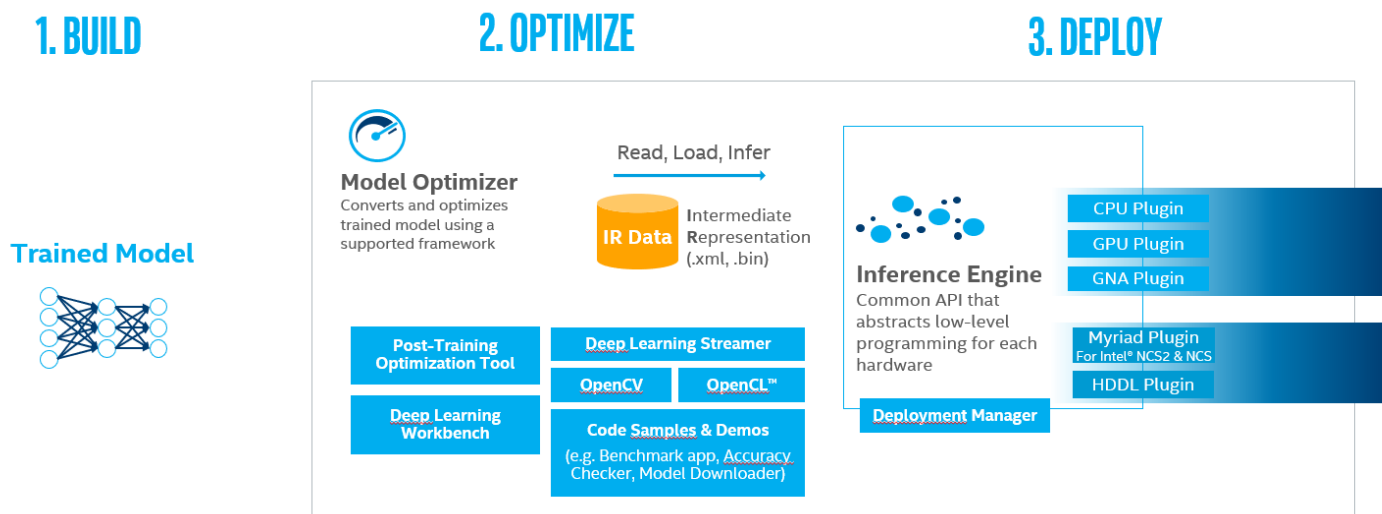
Intel® Distribution of OpenVINO™ toolkit

A toolkit for developing applications and solutions that use deep learning. The toolkit extends workloads across Intel® hardware (including accelerators) and maximizes performance.

- Enables deep learning inference from edge to cloud
- Accelerates AI workloads, including computer vision, audio, speech, language, and recommendation systems
- Supports heterogeneous execution across Intel® architecture and AI accelerators—CPU, iGPU, Intel® Movidius™ Vision Processing Unit (VPU) and Intel® Gaussian & Neural Accelerator (Intel® GNA)—using a common API
- Speeds up time to market via a library of functions and preoptimized kernels
- Includes optimized calls for OpenCV, OpenCL™ kernels, and other industry tools and libraries.

Learn more: <https://software.intel.com/content/www/us/en/develop/tools/openvino-toolkit.html>

You might also want to read another byteLAKE's report about how we leveraged OpenVINO to optimize byteLAKE's Cognitive Services, our AI product for Industry 4.0 (AI-assisted Visual Inspection, AI-powered Big Data Analytics). Find out more at: www.byteLAKE.com/en/CognitiveServices.



Report

Performance and scalability analysis of byteLAKE's CFD Suite (AI-accelerated CFD) for Computational Fluid Dynamics simulations acceleration.

Hardware infrastructure

byteLAKE has collaborated with Lenovo (Lenovo Infrastructure Solutions Group), Intel, and Tridiagonal Solutions to perform the performance analysis of the CFD Suite across the following configurations:

1. Single HPC node: 2* Intel Xeon Gold 6148 CPU @ 2.40GHz and 2 * NVIDIA V100 16GB, and 400GB RAM (**Gold, V100 respectively**)
2. Intel CPU-only HPC cluster, BEM supercomputer (860 TFLOPS); 22,000 cores; 724 nodes; 1,600 Intel Xeon CPU E5-2670 @ 2.30GHz – 12 cores processors; 74,6 TB RAM (**BEM for cluster or E5-2670 for a single node**)
3. Desktop platform: Intel Core i7-3770 CPU @ 3.40GHz – 4 cores (**Core-i7 or i7**) + NVIDIA GeForce GTX TITAN GPU (**TITAN**)
4. Single HPC node: Intel Xeon CPU E5-2695 @ 2.30GHz – 12 cores (**E5-2695**)

Abbreviations in brackets are used further in this document when referring to a particular platform.

Software environment

1. Python: 3.8.2
2. TensorFlow: 2.4.1
3. Horovod 0.21.3
4. OpenVino: 2021.2.200
5. NVIDIA Cuda: 10.1
6. cuDNN: 7.6.5

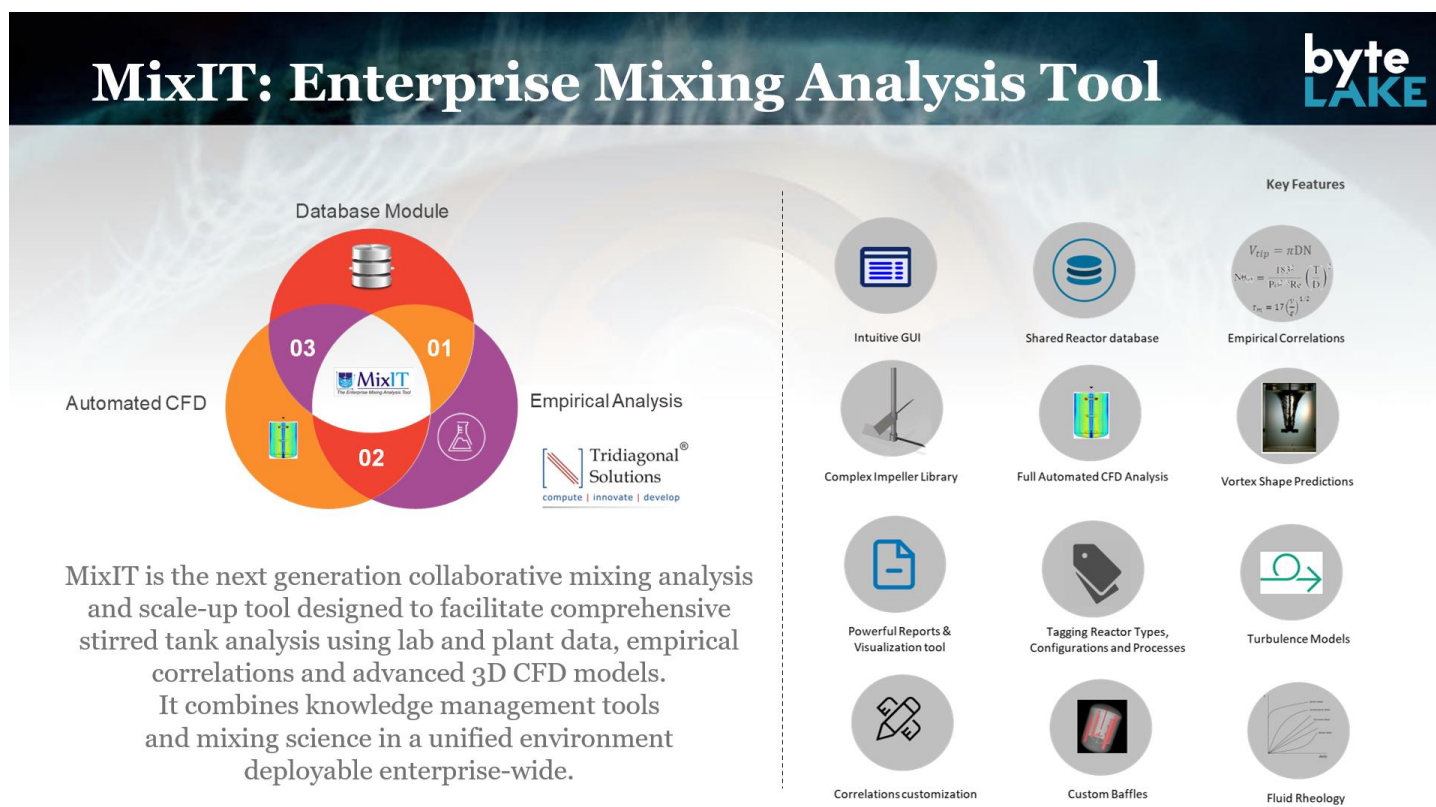
In addition, CFD Suite has been optimized to Intel technologies to further measure and ensure the scalability of the solution and maximize the performance.

CFD configuration for benchmarking purposes

1. Mesh size: 400 and 40,000 for AI models training benchmark and 1,000,000 for inferencing benchmark (relevant trained AI models have been used)
2. CFD Suite mode: ACCURATE
3. DATA: 3D (for both: conventional CFD solver and AI predictions)

Benchmark overview

Our research includes methods of accelerating CFD simulations by integrating a conventional CFD solver with byteLAKE's CFD Suite (AI-accelerated CFD). The investigated phenomenon is about chemical mixing. The considered CFD simulations belong to a group of steady-state simulations and utilize the MixIT tool from Tridiagonal Solutions, which is based on the open-source CFD toolbox.

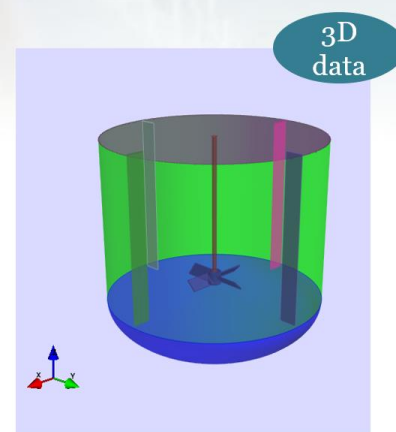


byteLAKE's CFD Suite's AI module is implemented as a CNN (Convolutional Neural Network) supervised learning algorithm. Data is distributed by creating separate AI sub-models for each quantity of the simulated phenomenon. These sub-models can then be pipelined during the inference stage to reduce the execution time or called one-by-one to reduce memory requirements.

The AI model is responsible for getting results from a set of iterations as the input, feed the network, and return the final iteration. Each iteration has a constant geometry and processes the 3D mesh. To reduce memory requirements, CFD Suite creates a set of sub-models that independently work on a single quantity of the simulated phenomenon. Thanks to this approach, all the sub-models are learned sequentially, which significantly reduces memory requirements.

Simulation of Chemical Mixing

- **Selected phenomenon for the simulation**
 - The goal is to compute the converged state of the liquid mixture in a tank equipped with a single impeller and a set of baffles
- **Based on different settings of the input parameters, we simulate a set of quantities**
 - the velocity vector field (U)
 - pressure scalar field (p)
 - turbulent kinetic energy (k) of the substance
 - turbulent dynamic viscosity (μ_t)
 - turbulent kinetic energy dissipation rate (ϵ)



Benchmark #1: DISC vs. RAM (training)

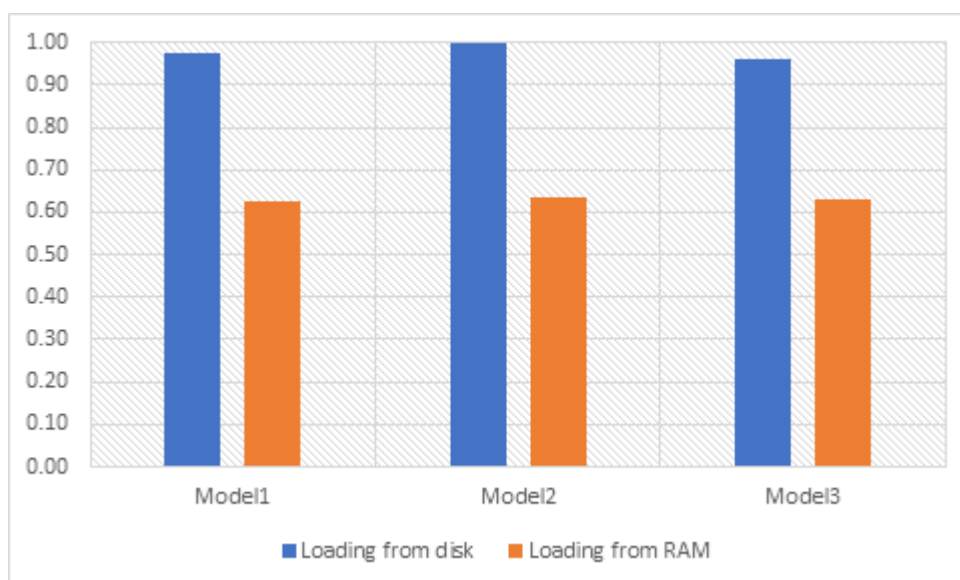
The first benchmark aims to validate the performance improvement by optimizing data access to the training dataset. Here we test the training phase of 3 quantities, where the first and second represent scalar quantities (pressure – **model1**, viscosity – **model2**), while the third one represents the vector quantity (velocity – **model3**) of 3-dimensional mesh. The following configuration of the training is used:

epochs	2048	nodes	1
cells	400	Threads /node	8
Input iterations	1		

We take a single iteration as an input and return the converged state. We use here 8 OpenMP (shared memory model) threads for learning. We compare the method, where the data is loaded from the disk to reduce as much as possible of the host memory requirements, and the optimized version, where the data is stored in thread-safe structures in the host memory. Here we used the BEM node. The execution times [s] are included below:

	Model1	Model2	Model3
Loading from disk	972.38	996.92	956.40
Loading from RAM	623.07	634.19	628.01
Speedup	1.56	1.57	1.52

The normalized performance results are shown below:



The method allows us to reduce the execution time by a factor of 1.5. The remaining experiments are based on the optimized version of CFD Suite.

Benchmark#2: Small vs. Big mesh scalability (training)

The next test aims to validate the scalability of the training module based on the Horovod distributed parallelization. Here we compare two kinds of training, the first one is based on a relatively small mesh (400 cells with a single iteration as an input) and a big mesh (40,000 cells with 16 input iterations).

Training A – small (configuration):

epochs	2048	nodes	1-64
cells	400	Threads /node	8
Input iterations	1		

Performance results (execution time [s]):

nodes	Threads	model1	model2	model3
1	8	623.07	634.19	628.01
2	16	408.90	377.70	360.45
4	32	218.85	216.68	170.92
8	64	91.60	66.59	107.15
16	128	278.98	172.94	262.59
32	256	103.62	79.52	85.38
64	512	140.20	74.98	118.06

Training B – big (configuration):

epochs	512	nodes	1-64
cells	40000	Threads /node	8
Input iterations	16		

Performance results (execution time [s]):

nodes	Threads	model1	model2	model3
1	8	1645.27	1712.49	1799.80
2	16	858.89	878.00	934.80
4	32	446.48	448.72	485.62
8	64	228.62	232.19	247.27
16	128	119.52	117.76	129.90
32	256	65.18	65.34	69.80
64	512	36.85	35.59	38.41

Speedup analysis of the A – small training across 64 nodes of the BEM cluster:

nodes	model1	model2	model3
1	1.00	1.00	1.00
2	1.52	1.68	1.74
4	2.85	2.93	3.67
8	6.80	9.52	5.86
16	2.23	3.67	2.39
32	6.01	7.98	7.36
64	4.44	8.46	5.32

Speedup analysis of the B – big training across 64 nodes of the BEM cluster:

nodes	model1	model2	model3
1	1.00	1.00	1.00
2	1.92	1.95	1.93
4	3.69	3.82	3.71
8	7.20	7.38	7.28
16	13.77	14.54	13.86
32	25.24	26.21	25.78
64	44.65	48.12	46.86

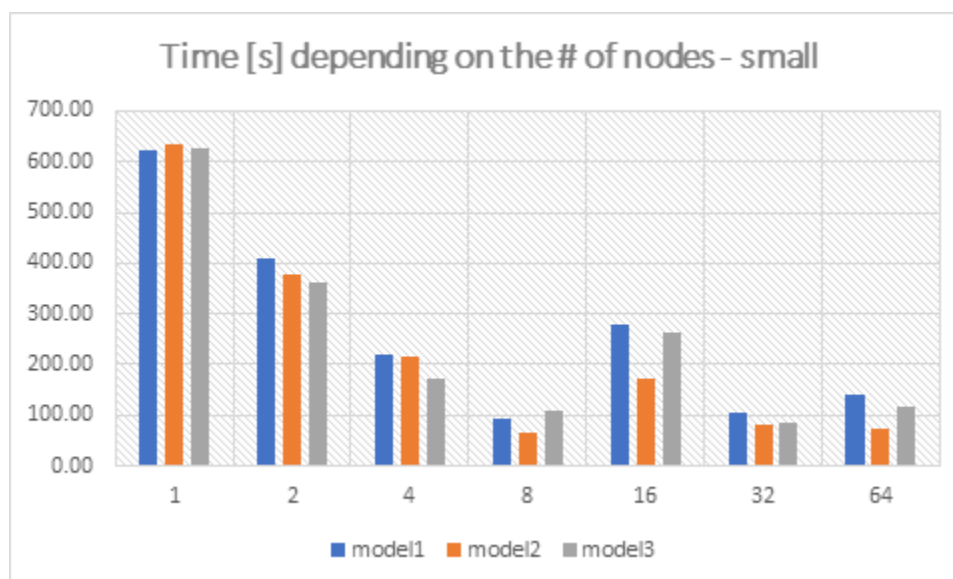
Efficiency analysis of the A – small training across 64 nodes of the BEM cluster:

nodes	model1	model2	model3
1	1.00	1.00	1.00
2	0.76	0.84	0.87
4	0.71	0.73	0.92
8	0.85	1.19	0.73
16	0.14	0.23	0.15
32	0.19	0.25	0.23
64	0.07	0.13	0.08

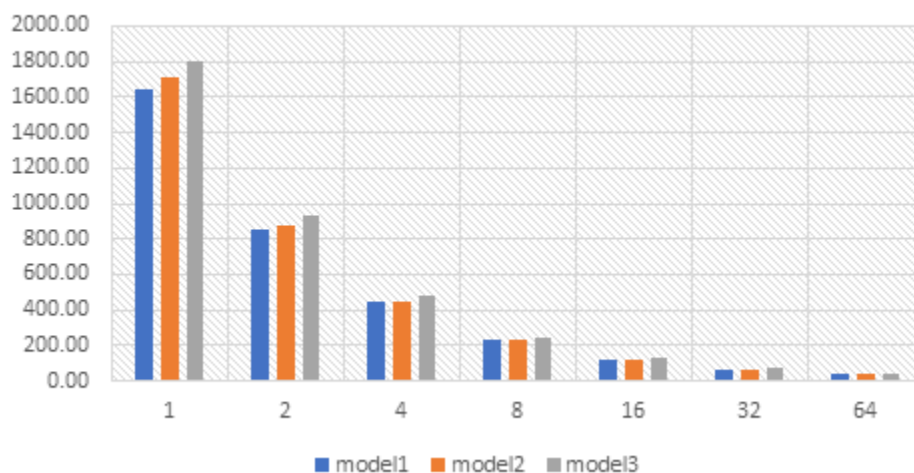
Efficiency analysis of the B – big training across 64 nodes of the BEM cluster:

nodes	model1	model2	model3
1	1.00	1.00	1.00
2	0.96	0.98	0.96
4	0.92	0.95	0.93
8	0.90	0.92	0.91
16	0.86	0.91	0.87
32	0.79	0.82	0.81
64	0.70	0.75	0.73

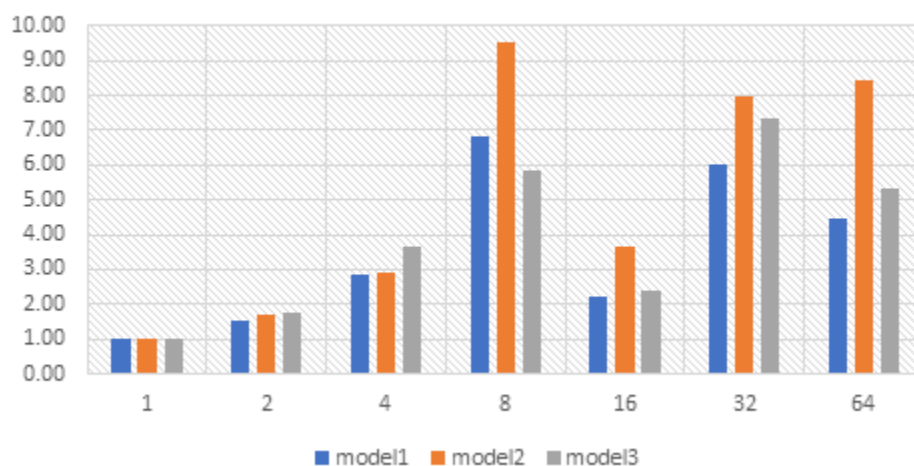
Scalability of the training based on the small (A) and big (B) meshes:



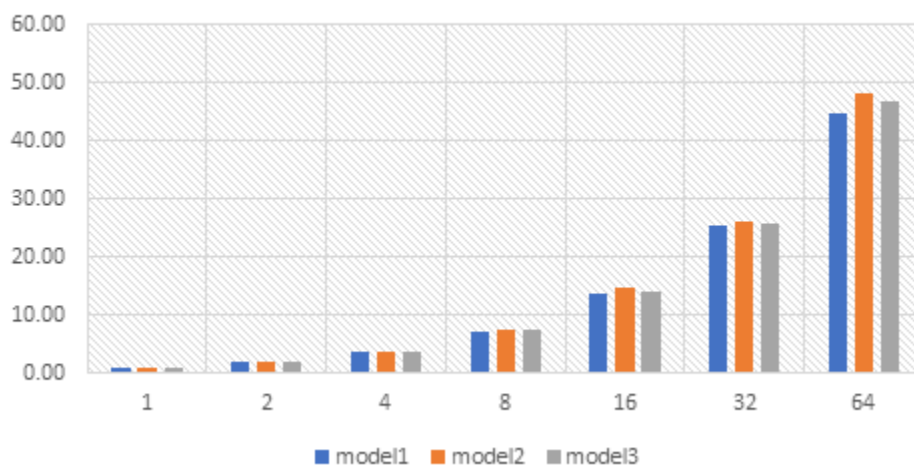
Time [s] depending on the # of nodes - big



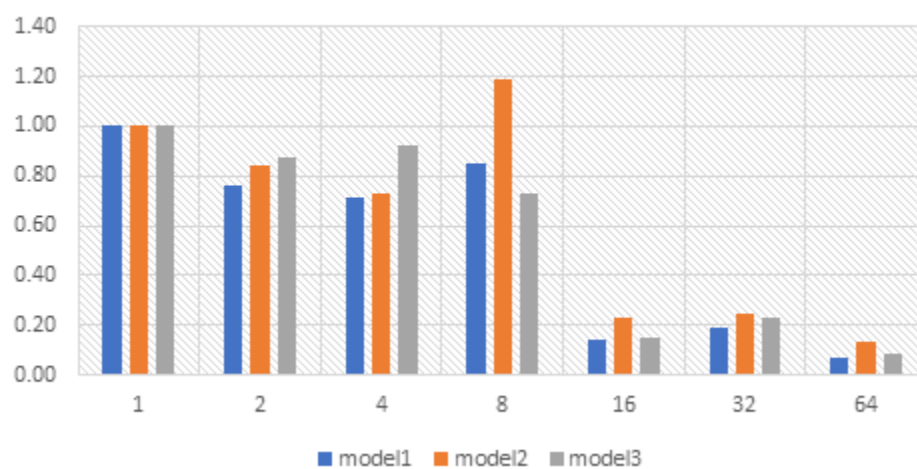
Speedup - small mesh



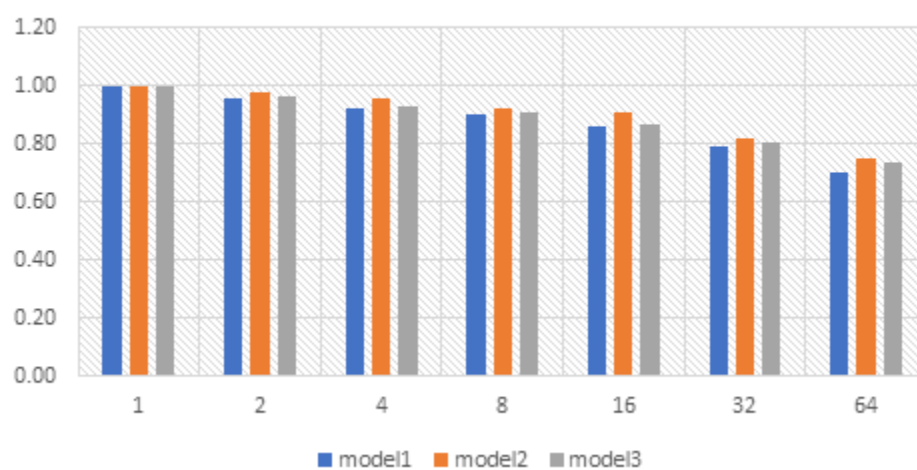
Speedup - big mesh



Efficiency - small mesh



Efficiency - big mesh



Conclusions:

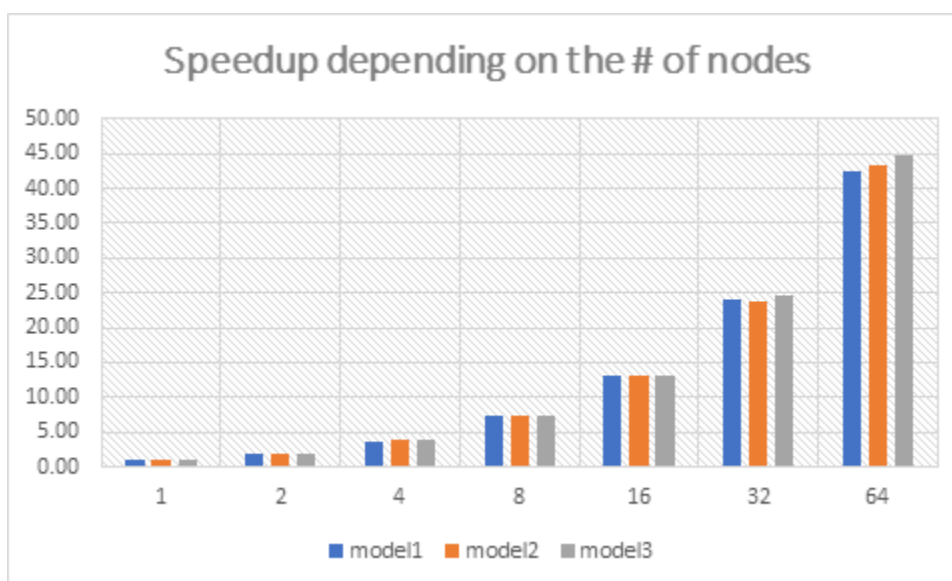
- **Small mesh:**
 - o Superefficiency observed in some tests for small meshes (8 nodes – model2) – shows that performance results are noised by the communication among nodes
 - o The high impact of nodes communication on the performance for small meshes for more than 16 nodes
 - o Stable speedup and high efficiency up to 8 nodes (64 threads)
 - o Not enough computation to saturate the performance of 16 and more nodes
- **Big mesh:**
 - o A benchmark shows time to results reduction by a factor of 48 for 64 nodes
 - o **We can observe a stable efficiency** (>90% up to 8 nodes, ~70% up to 64 nodes)
 - o Low impact of the distributed communication on the performance for all 3 models
- The level of parallelization for both cases is $64 \text{ nodes} * 8 \text{ threads} = 512 \text{ threads}$

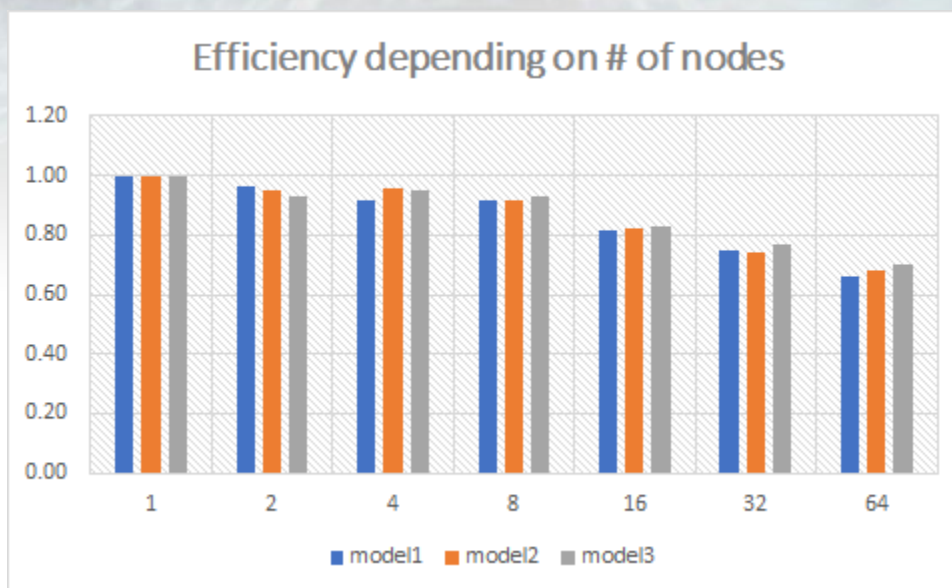
Benchmark#3: Platforms comparison (training)

The goal of this benchmark is to compare the performance of the AI training on all platforms listed in a “Hardware infrastructure” section above. In this test, we utilize 12 cores per BEM cluster node, 4 cores for i7, and 20 cores for Gold. In this section, we also verify the performance of shared memory and distributed memory parallelization. The configuration of this test includes the following parameters:

epochs	512	nodes	1-64
cells	40000	Threads /node	12
Input arrays	16		

After increasing the number of threads per node the scalability is still high (>70% efficiency up to 64 nodes) which confirms the previous benchmark:





Speedup of the cluster-based training over a single node execution:

nodes	model1	model2	model3
1	1.00	1.00	1.00
2	1.93	1.91	1.86
4	3.66	3.82	3.79
8	7.33	7.35	7.42
16	13.08	13.13	13.23
32	23.98	23.66	24.62
64	42.42	43.44	44.73

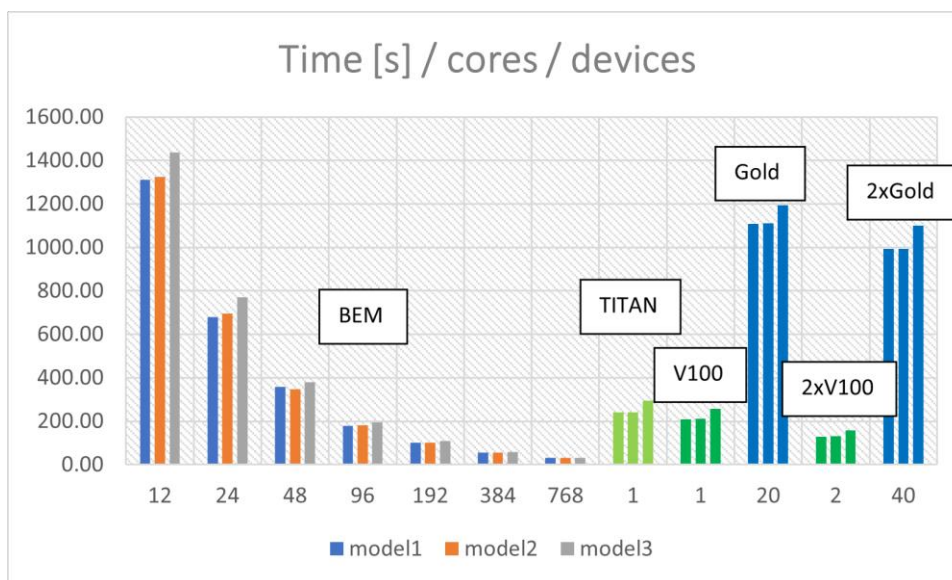
Efficiency of the cluster-based training over a single node execution:

nodes	model1	model2	model3
1	1.00	1.00	1.00
2	0.97	0.95	0.93
4	0.91	0.95	0.95
8	0.92	0.92	0.93
16	0.82	0.82	0.83
32	0.75	0.74	0.77
64	0.66	0.68	0.70

The third model, which is the most compute-intensive since it feeds a network with a vector quantity achieves the best efficiency. It shows that the more compute-intensive model the better scalability is achieved. The level of parallelization is 64 nodes * 12 threads = 768 threads. We can conclude that the Horovod based implementation is well scalable for a distributed training.

Below are included execution times for all the platforms:

Platform	nodes	Threads or number of GPUs/CPU's	model1	model2	model3
BEM	1	12	1312.60	1324.13	1437.99
BEM	2	24	679.48	694.70	771.08
BEM	4	48	358.69	346.82	379.13
BEM	8	96	179.13	180.27	193.72
BEM	16	192	100.37	100.86	108.71
BEM	32	384	54.74	55.96	58.42
BEM	64	768	30.94	30.48	32.15
TITAN	1	1 GPU	241.30	241.30	294.18
V100	1	1 GPU	209.68	210.29	255.81
Gold	1	20 (1 Gold)	1108.96	1109.45	1193.04
V100	1	2 GPU	127.90	130.38	156.81
Gold	1	40 (2 Golds)	991.91	991.71	1101.05



Conclusions:

The CFD models are generally memory-bound algorithms. Here we use 3D meshes. For the AI-acceleration, we have a relatively big amount of data to feed the model. It enforces the use of reduced AI model architectures to make it possible to feed the network. Consequently, we have a lot of data processed by a relatively small network (up to 16 layers).

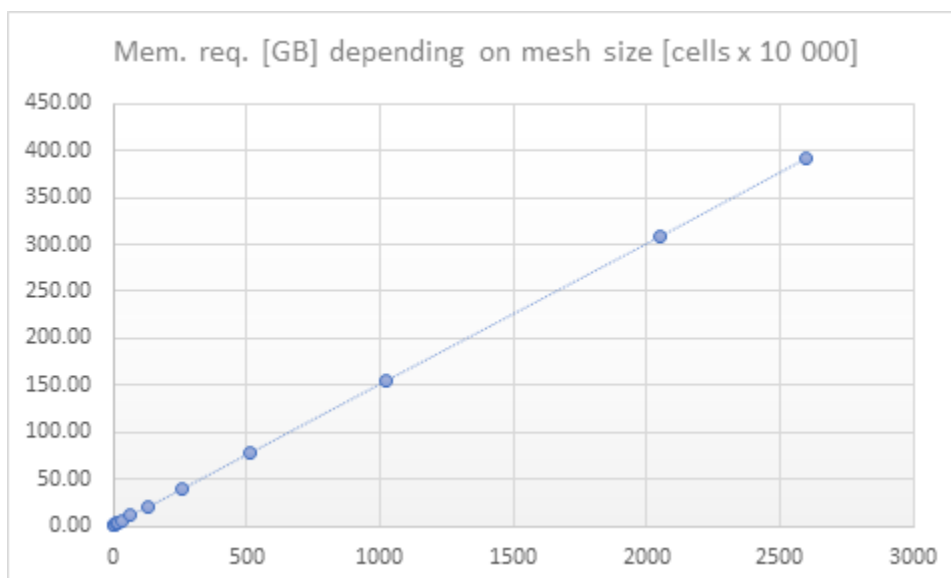
As result, we can observe that:

- There is no expected speedup across a single node (1 Gold, 20 cores vs. 2 Golds, 40 cores). Poor performance improvement across an OpenMP (shared memory model) parallelization when using more than 20 threads within a single node (speedup by a factor of $\sim 1.11x$). The reason is that the training is not compute-intensive enough.
- There is also no big difference between a single V100 and TITAN – since the training is not compute-intensive enough.
- **The performance improvement is much better in the case of distributed training (1xV100 vs 2xV100, or BEM – up to 64 nodes).**
- **The cluster implementation (BEM) based on the Horovod framework allows us to overtake the performance of a single V100 using 8 nodes.** 16 nodes are 1.3x faster than 2xV100. By comparing the Gold results and a single node of BEM (single E5-2670 CPU), we can assume that a cluster with 8xGold would allow us to achieve comparable results versus 2xV100.

Benchmark#4: Memory requirements for training

The goal of this experiment is to measure the memory requirements of the training process depending on mesh size. The results are obtained by analyzing the memory requirements after jobs submitting on the cluster with the Portable Batch System (or simply PBS) statistics. The results are listed below. Since the memory requirements increase linearly, and the available host memory per node is 128GB (BEM), the final 3 measurements are extrapolated.

Cells	Cells x10,000	Memory requirements [GB]	
10000	1	1.44	measured
20000	2	1.59	measured
40000	4	1.89	measured
80000	8	2.48	measured
160000	16	3.68	measured
320000	32	6.08	measured
640000	64	10.88	measured
1280000	128	20.47	measured
2560000	256	39.66	measured
5120000	512	78.03	measured
10240000	1024	154.78	extrapolated
20480000	2048	308.28	extrapolated
26000000	2600	391.03	extrapolated



Conclusions

The memory requirements increase linearly with increasing the number of cells.

Therefore:

- The BEM cluster allows us to train the models for meshes up to 5-8 million cells
- To train the AI models for meshes around 26 million cells, we would need a platform containing 400GB of the host memory (RAM). Such training on a CPU-only single node would take approximately 1 month. If BEM's RAM memory were upgraded, it is assumed that such training would take less than a day.

Inferencing

The AI-accelerated simulation (prediction/inferencing) is composed of a set of steps. First, we need to execute i.e. 10% of the conventional CFD solver. Then the remaining 90% is predicted by CFD Suite.

Example execution time of the conventional CFD solver is included in the table below.

	Execution time [s]
Conventional CFD solver (10%)	1424.12
Conventional CFD solver (90%)	12817.12
Conventional CFD solver (100%)	14241.24 (sum of above figures)

To predict the results of such conventional CFD solver with AI (using byteLAKE's CFD Suite), the following is performed by byteLAKE's CFD Suite:

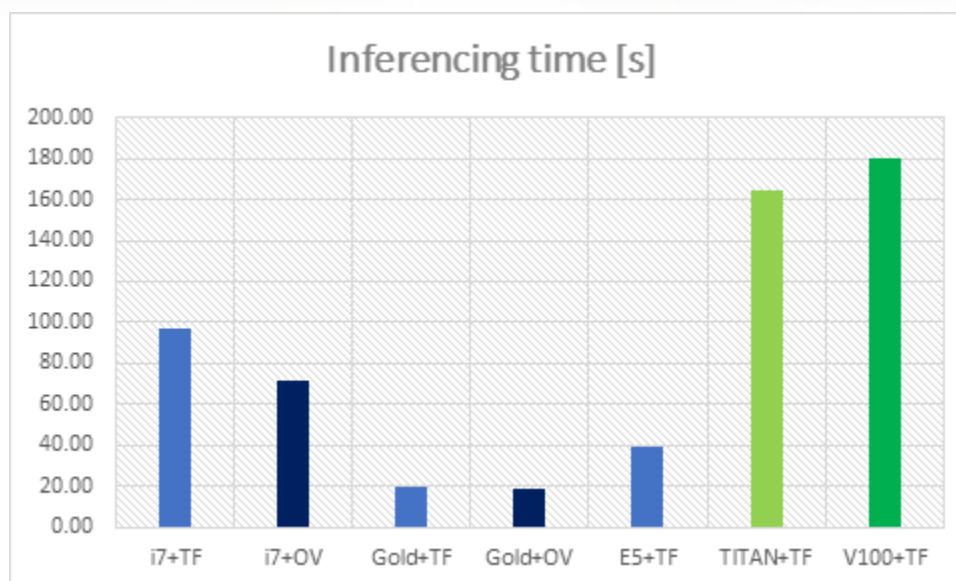
- data import from the conventional CFD solver ("Data import" in the table below) (as indicated above, just a fraction of the initial results)
- data normalization,
- **inferencing with AI-models** ("inferencing" row) (that is where we leverage trained AI models)
- and data export to the conventional solver format ("Data export") (so that we output results ready for analysis by existing CAE tools).

Table below summarizes the collective time of execution for all these steps in the row named as: “**CFD Suite (90%)**”.

Summing up, the total time to results for CFD Suite is then a sum of all the above steps (prediction) plus time of generating a fraction of the initial results of the conventional CFD solver: **Conventional CFD solver (10%) + CFD Suite (90%)**.

Device	Core-i7	Core-i7	2xGold	2xGold	2xE5-2695	TITAN	V100
Framework	Tensor Flow	Open VINO	Tensor Flow	Open VINO	Tensor Flow	Tensor Flow	Tensor Flow
<i>Data import</i>	25.29	25.48	22.42	22.87	40.78	25.32	22.21
<i>Data normalization</i>	81.59	81.60	64.03	64.00	230.18	81.60	66.24
<i>Inferencing</i>	97.41	71.10	19.66	18.91	39.39	164.38	180.31
<i>Data export</i>	10.68	10.69	9.58	9.56	16.16	10.69	9.51
CFD Suite (90%)	214.97	188.87	115.70	115.34	326.52	281.98	278.27
Conventional CFD solver (10%) + CFD Suite (90%)	1639.09	1613.00	1539.82	1539.47	1750.64	1706.10	1702.39
(prediction)							
<i>Abbreviation for charts</i>	<i>i7+TF</i>	<i>i7+OV</i>	<i>Gold+TF</i>	<i>Gold+OV</i>	<i>E5+TF</i>	<i>TITAN+TF</i>	<i>V100+TF</i>

First, we compare the performance of the inferencing stage only (“Inferencing” row in the table above; pure AI model inferencing without any overhead related to data formatting). In addition, we also analyze the inferencing time based on the TensorFlow framework, and inferencing optimized by the OpenVINO tool. Those results are presented below.



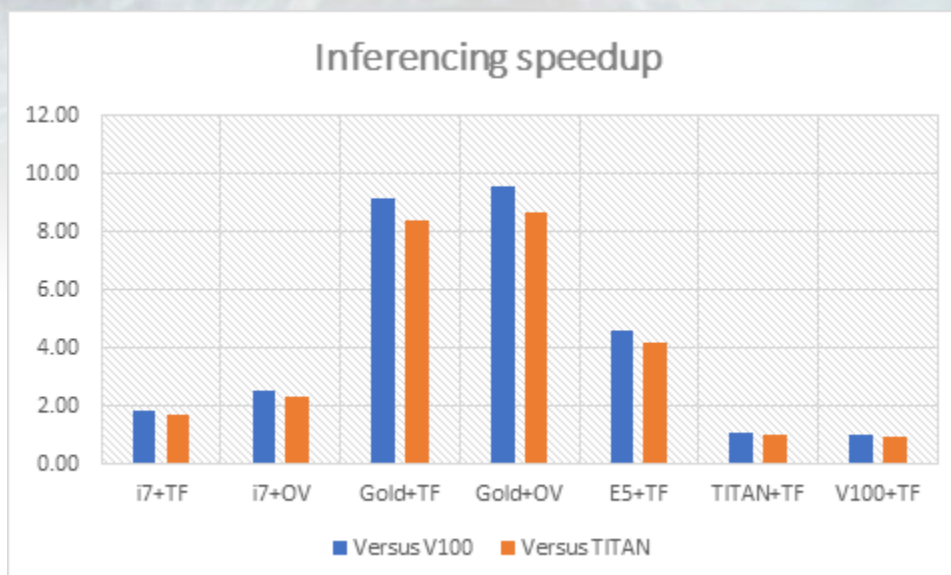
The OpenVINO framework allows us to reduce the execution time up to 1.36 times. However, for the Gold, the performance improvement is negligible. CFD Suite uses a relatively small number of layers (up to 16 layers), so there is a limited possibility to optimize the network by OpenVINO. In other byteLAKE’s product called Cognitive Services (AI for Industry 4.0), optimizations with OpenVINO led to acceleration of around 10x. Read more at www.byteLAKE.com/en/CognitiveServices.

The overall inference process within CFD Suite consists of a set of sub-models inferences that are pipelined across the CPU cores, or executed one-by-one on the GPU, which is parallelized by cuDNN. The inferencing on the GPU has a much higher overhead than on the CPU. The overhead is related to the memory allocation and data transfer thru PCIe from the host to the global memory of the GPU.

As result, GPU gives a significantly lower performance of the inference process.

The performance speedups comparison between CPUs versus GPUs are presented in the table below. **Inferencing is up to 9.5x faster on Gold (CFD Suite optimized with OpenVINO) than on V100.**

CPU \ GPU	i7+TF	i7+OV	Gold+TF	Gold+OV	E5+TF	TITAN+TF	V100+TF
Versus V100	1.85	2.54	9.17	9.53	4.58	1.10	1.00
Versus TITAN	1.69	2.31	8.36	8.69	4.17	1.00	0.91

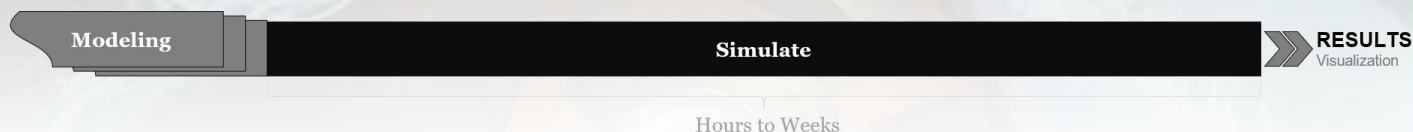


The below table shows the time of inferencing including overhead (data import, normalization, data export – row “90% of simulation” plus the time of generating a fraction of the initial results by the conventional CFD solver – row “100% of simulation”).

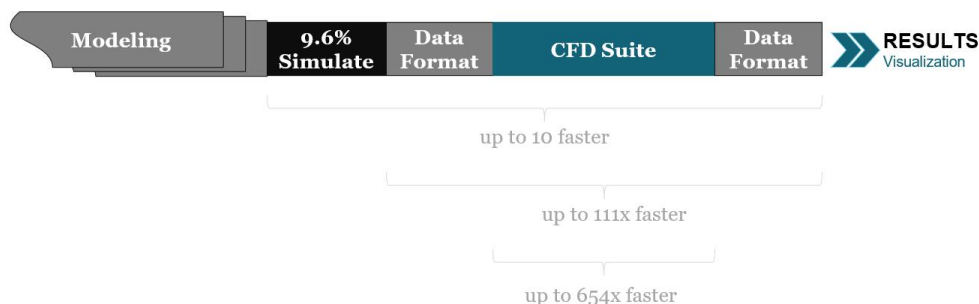
We conclude, that using 2xXeon Gold CPUs 90% of the simulation is predicted 111x faster than CFD solver computation, and 9.25x faster considering 10% overhead of the CFD solver.

Early benchmark (ACCURATE)

- Traditional workflow**



- CFD Suite (mode: accurate)**



bottom line

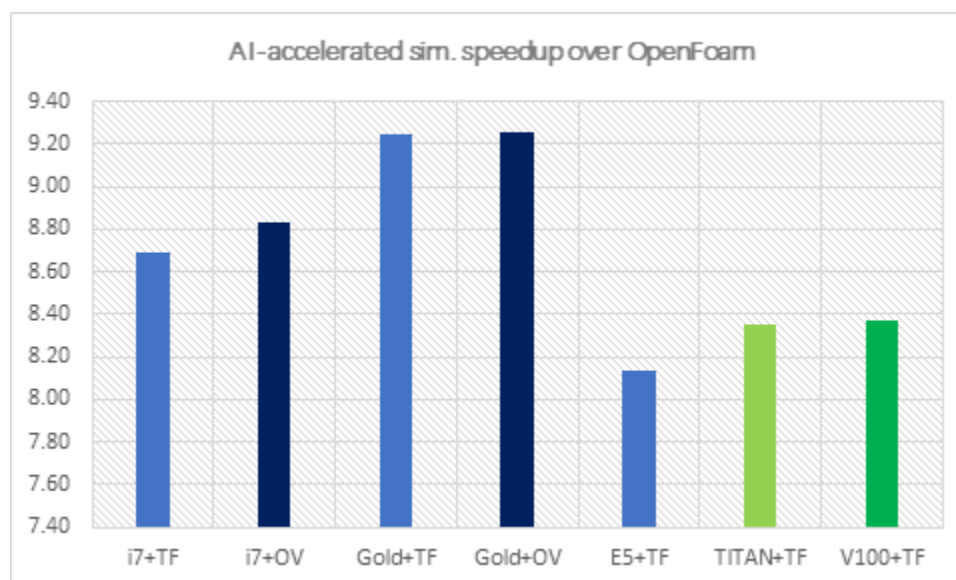
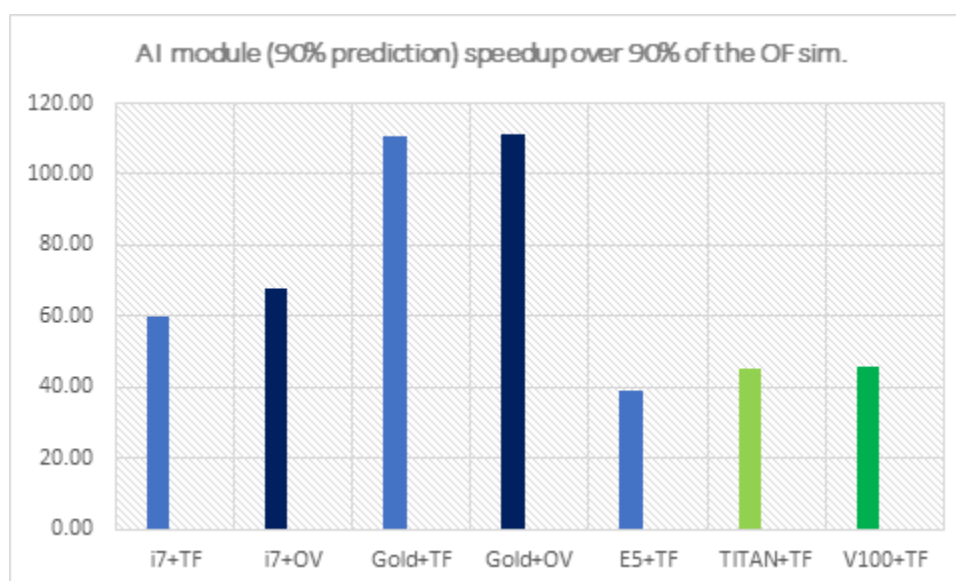
Accuracy: 93%

example time-to-results:

~24 mins

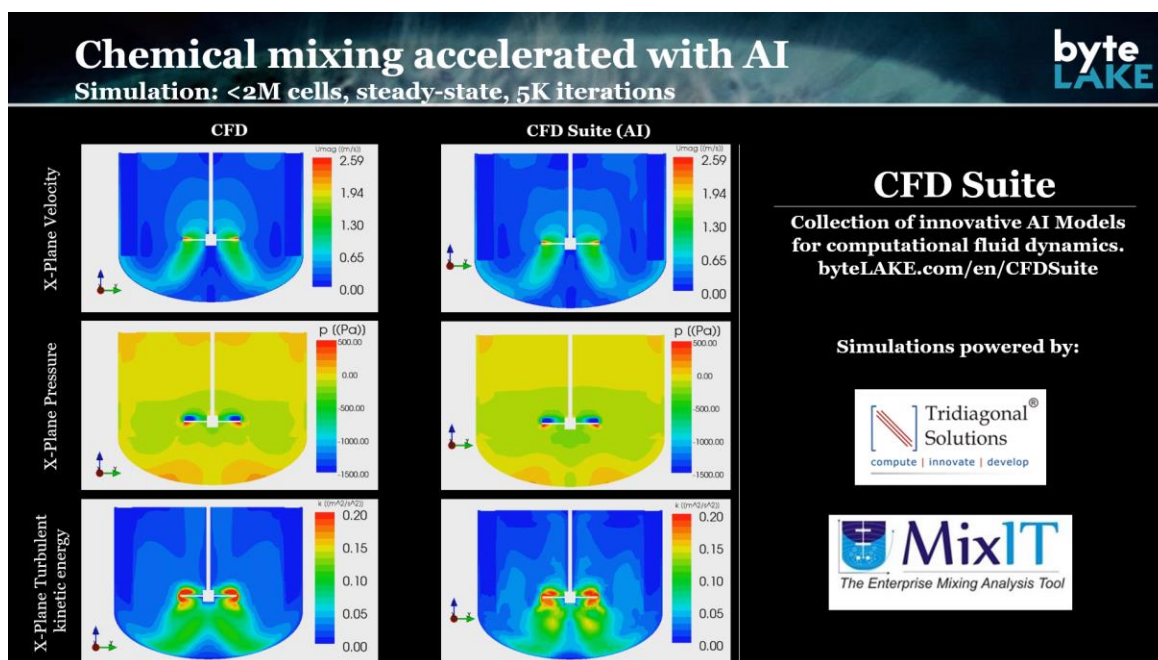
vs 4hrs

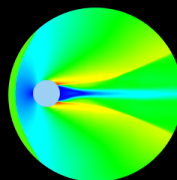
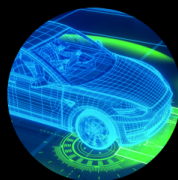
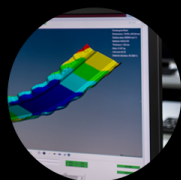
Device	Core-i7	Core-i7	2xGold	2xGold	2xE5-2695	TITAN	V100
Framework	Tensor Flow	OpenVino +TF	Tensor Flow	OpenVino +TF	Tensor Flow	Tensor Flow	Tensor Flow
90% of simulation	59.62	67.86	110.78	111.12	39.25	45.45	46.06
100% of simulation	8.69	8.83	9.25	9.25	8.13	8.35	8.37
Abbreviation for charts	<i>i7+TF</i>	<i>i7+OV</i>	<i>Gold+TF</i>	<i>Gold+OV</i>	<i>E5+TF</i>	<i>TITAN+TF</i>	<i>V100+TF</i>



Key takeaways

- **CFD Suite accelerates time to results for conventional CFD solvers by a factor of at least 10x and keeps the accuracy at the level of at least 93%.**
- New AI models are constantly added by byteLAKE to the CFD Suite which gradually increases the number of CFD simulations that can be handled by the CFD Suite off-the-shelf. To do so byteLAKE collaborates with a growing number of industry leaders.
- CFD Suite is an add-on to existing CAE/CFD tools and its integration is a straightforward process.
- **CFD Suite is a data-driven solution. Therefore, past simulations done by conventional CFD solvers are required to train its AI models so that they can predict the results.**
- CFD Suite is a scalable solution, and we observed a stable efficiency across cluster nodes.
- 8 nodes of the presented HPC cluster (BEM) can overtake a single V100 GPU for training. 16 nodes were 1.3x faster than 2xV100. By comparing the Gold results and a single node of BEM (single E5-2670 CPU), we can assume that a cluster with 8xGold would allow us to achieve comparable results versus 2xV100.
- The memory requirements increase linearly with increasing the number of cells (for CFD Suite training). Inferencing (prediction) can be done on typical desktop configurations.
- The inference process is executed up to 9.5x faster on the Intel Xeon Gold with CFD Suite optimized with OpenVINO compared with V100 GPU.
- By configuring the CFD Suite to use AI models in ACCURATE mode and using 2xXeon Gold CPUs, 90% of the simulation is predicted 111x faster than CFD solver computation, and 9.25x faster considering 10% overhead of the CFD solver.



byte
LAKEAI
for
CFD**CFD Suite**

Collection of innovative AI Models
for computational fluid dynamics.
byteLAKE.com/en/CFDSuite

byteLAKE

Artificial Intelligence for Industries. Products and Services.

www.byteLAKE.com**About byteLAKE**

byteLAKE is a bespoke AI & HPC software company developing AI-powered solutions for enterprises. The company offers both products and services, enabling innovative, AI-powered automation and data-driven, proactive operations across various industries i.e., AI-assisted Visual Inspection and Big Data analytics for manufacturing, AI-accelerated Computational Fluid Dynamics, AI for Industry 4.0, workflow and document processing automation etc. To learn more about byteLAKE's innovations, go to byteLAKE.com.