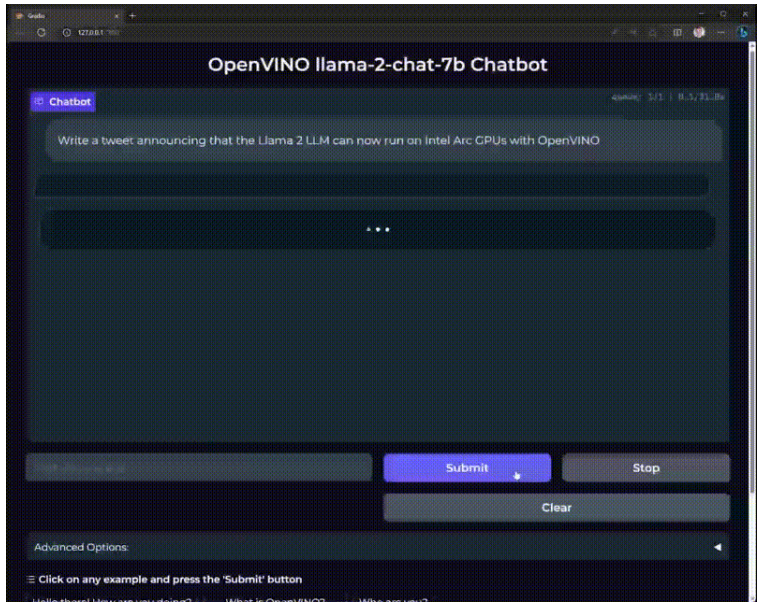
OpenVINO™ DEVCON Workshop Series 2023

OpenVINO 2023.1 アップデート概要

Computer Vision Specialist

Technical Sales Specialist, Sales & Mktg Grp.

2023.1



GenAI

LLM

OpenVINO 2023.1
生成系AI導入がしやすくなりました

- 簡素化されたワークフロー
- PyTorchとの親和性向上
- 大規模モデルの最適化
- 最新のnotebookサンプルプログラム

OpenVINO™ Toolkit

1 モデル

PyTorch TensorFlow TensorFlow Lite PaddlePaddle ONNX Keras Caffe mxnet KALDI

2 最適化

OpenVINO™

パフォーマンス最適化

CPU: intel Atom, intel Core, intel Xeon, arm (NEW IN 2023)

GPU: intel Iris Xe Graphics, intel Arc Graphics, intel Data Center GPU Flex Series

VPU: intel Movidius

FPGA: intel FPGA AI Suite

3 デプロイ

Windows Linux macOS

oneAPI

Powered by oneAPI

The productive, smart path to freedom for accelerated computing from the economic and technical burdens of proprietary alternatives.

新オフライン モデル変換ツール - ovc

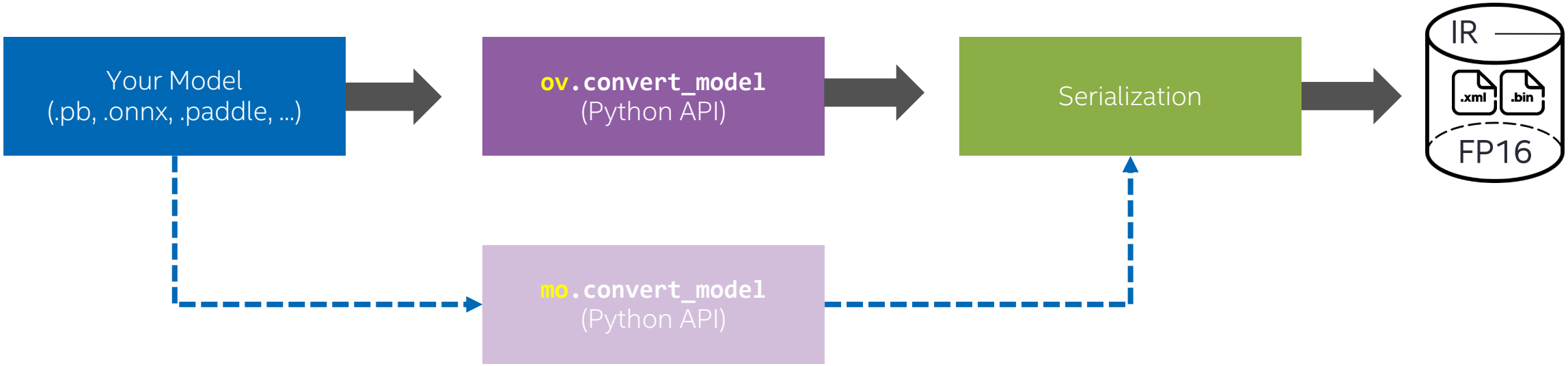
```
mo -h
```

```
pip install openvino-dev
```

```
ovc -h
```

```
pip install openvino
```

Python モデル変換API：ov.convert_model




```
ov_model = ov.convert_model("model.onnx", compress_to_fp16=True)

ov.save_model(ov_model, "converted_model.xml")
```

OpenVINO 2023.1

生成系AI導入がしやすくなりました

簡素化された
ワークフロー

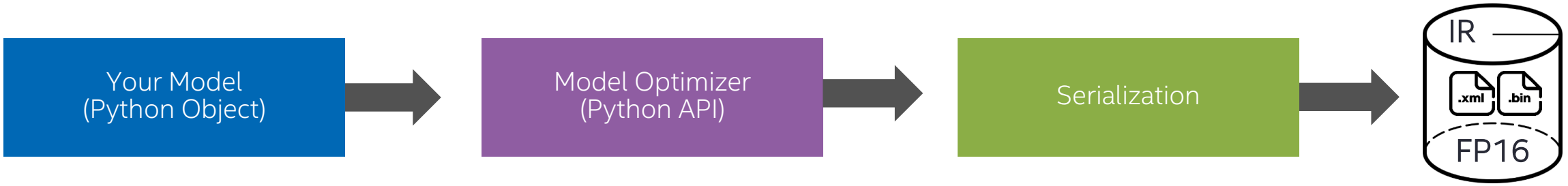
PyTorchとの親和性向上

大規模モデルの最適化

最新のnotebook
サンプルプログラム

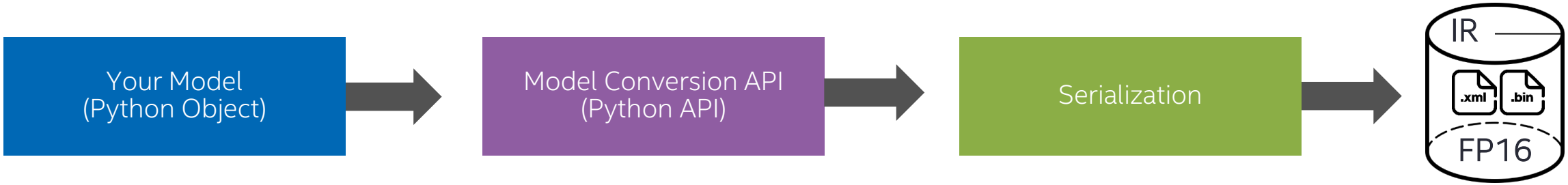
New PyTorch frontend

- `torch.nn.Module derived classes`
- `torch.jit.ScriptModule`
- `torch.jit.ScriptFunction`



New PyTorch frontend

- `torch.nn.Module` derived classes
- `torch.jit.ScriptModule`
- `torch.jit.ScriptFunction`




```
pytorch_model = torchvision.models.resnet50(pretrained=True)

ov_model = ov.convert_model(pytorch_model,
                            input_shape=[1,3,224,224],
                            compress_to_fp16=True)

ov.save_model(ov_model, "converted_model.xml")
```

New PyTorch frontend

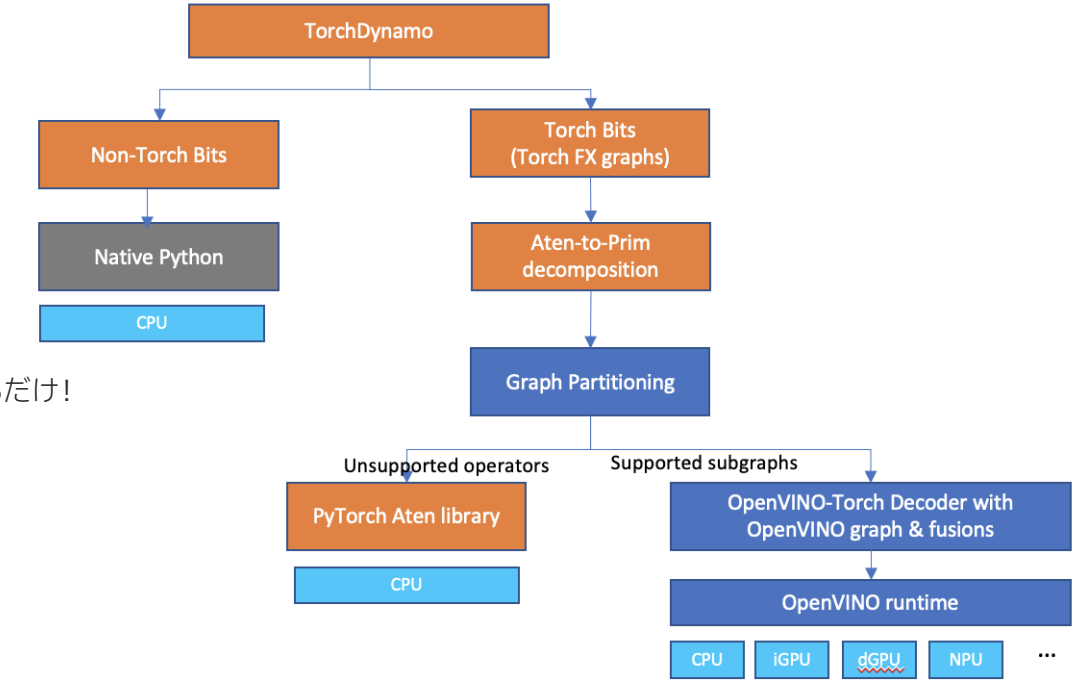
```
import openvino as ov
import torch
from torchvision.models import resnet50

model = resnet50(pretrained=True)

# prepare input_data
input_data = torch.rand(1, 3, 224, 224)

ov_model = ov.convert_model(model, example_input=input_data)

###### Option 1: Save to OpenVINO IR:

# save model to OpenVINO IR for later use (read by ov.read_model())
ov.save_model(ov_model, 'model.xml')

###### Option 2: Compile and infer with OpenVINO:

# compile model
compiled_model = ov.compile_model(ov_model, device_name='CPU')

# run the inference
result = compiled_model(input_data)
```

torch.compile

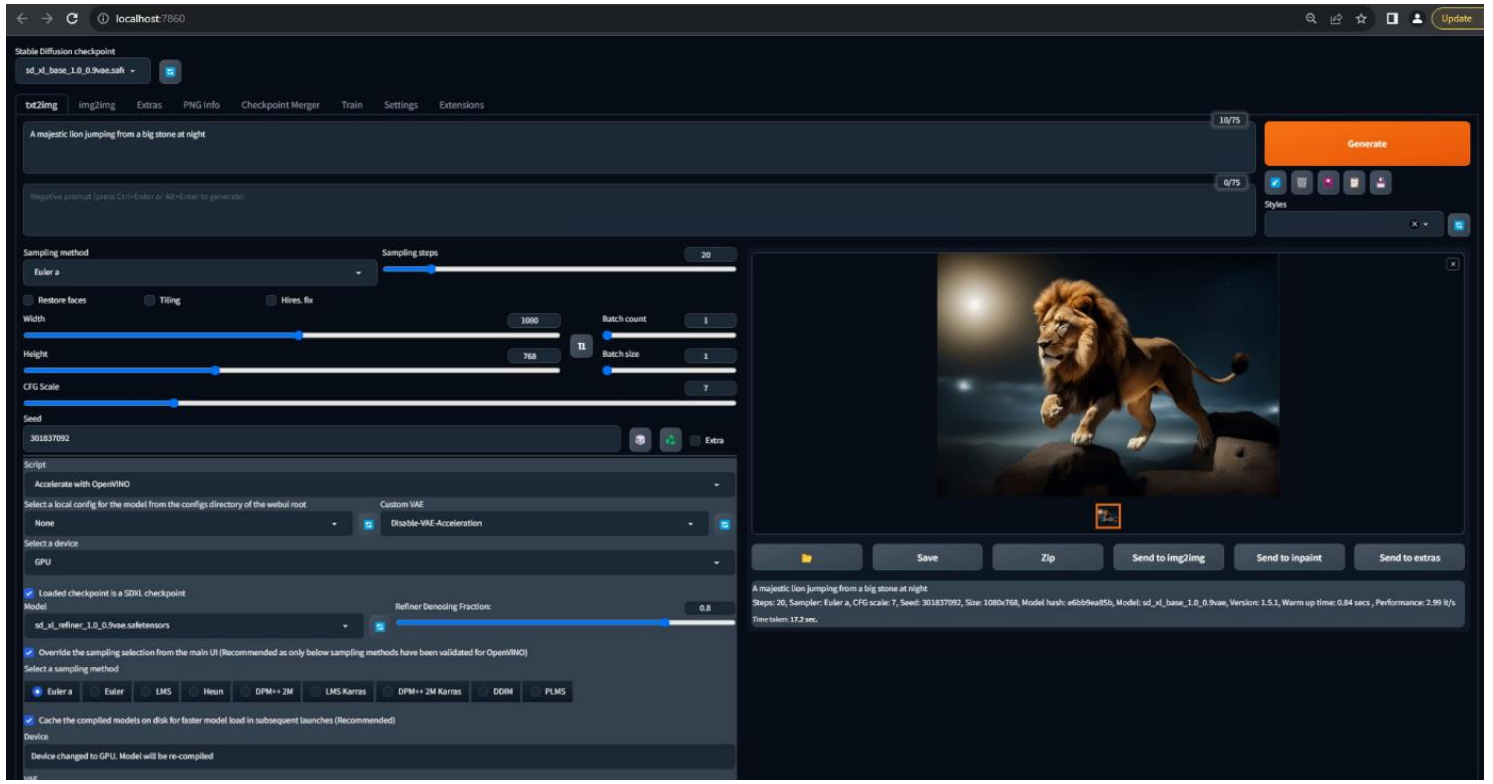
OpenVINOバックエンド

```python
import torch
import torchvision.models as models
import openvino.torch
model = models.resnet50(pretrained=True)
input = torch.rand((1,3,224,224))
model = torch.compile(model, backend='openvino')
pred = model(input)
```

2行変更するだけ!



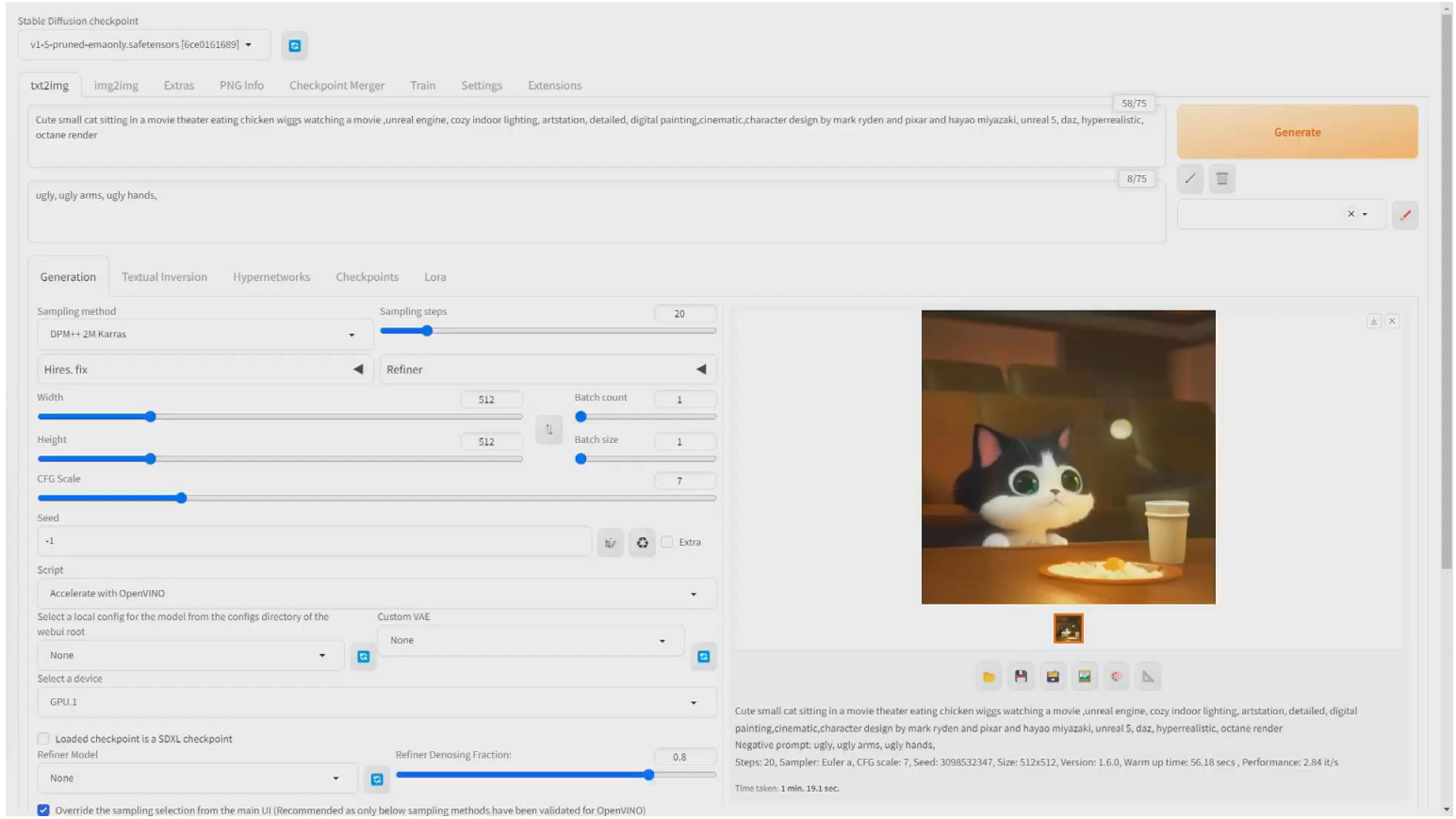
Automatic1111 Stable Diffusion Web UI



https://github.com/openvinotoolkit/stable-diffusion-webui

Demo

DEMO: Automatic1111 Stable Diffusion Web UI



Core i7-10700K, Arc A380, RAM 64GB, Win11, OV23.1

OpenVINO 2023.1

生成系AI導入がしやすくなりました

簡素化された
ワークフロー

PyTorchとの親和性向上

大規模モデルの最適化

最新のnotebook
サンプルプログラム

Optimum-Intel

2行書き換えるだけ!

```
- from transformers import AutoModelForCausalLM
+ from optimum.intel.openvino import OVModelForCausalLM

- model = AutoModelForCausalLM.from_pretrained(model_id)
+ ov_model = OVModelForCausalLM.from_pretrained(model_id)

generate_ids = ov_model.generate(input_ids)
```

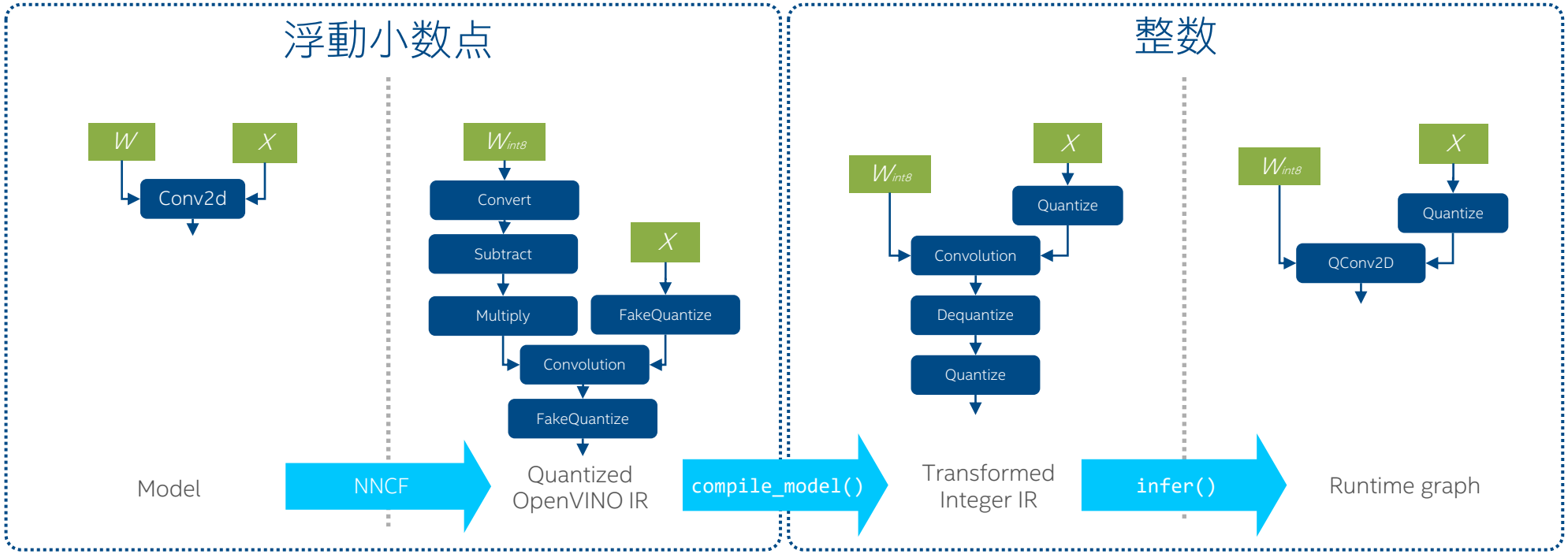
- bart,
- blenderbot,
- blenderbot-small
- bloom,
- codegen,
- gpt2,
- gpt_neo,
- gpt_neox,
- llama,
- marian,
- opt,
- pegasus,
- …

LangChainを使用したLLMの場合

統合されたOptimum-intel推論バックエンド

```
from langchain.llms import HuggingFacePipeline
from transformers import pipeline
- from transformers import AutoModelForCausalLM
+ from optimum.intel.openvino import OVModelForCausalLM

- model = AutoModelForCausalLM.from_pretrained(model_id)
+ ov_model = OVModelForCausalLM.from_pretrained(model_id)

pipe = pipeline("text-generation", model=ov_model, tokenizer=tokenizer,
max_new_tokens=128, pad_token_id=tokenizer.eos_token_id)

hf = HuggingFacePipeline(pipeline=pipe)

llm_chain = LLMChain(prompt=prompt, llm= hf)

output = llm_chain.run(question)
```

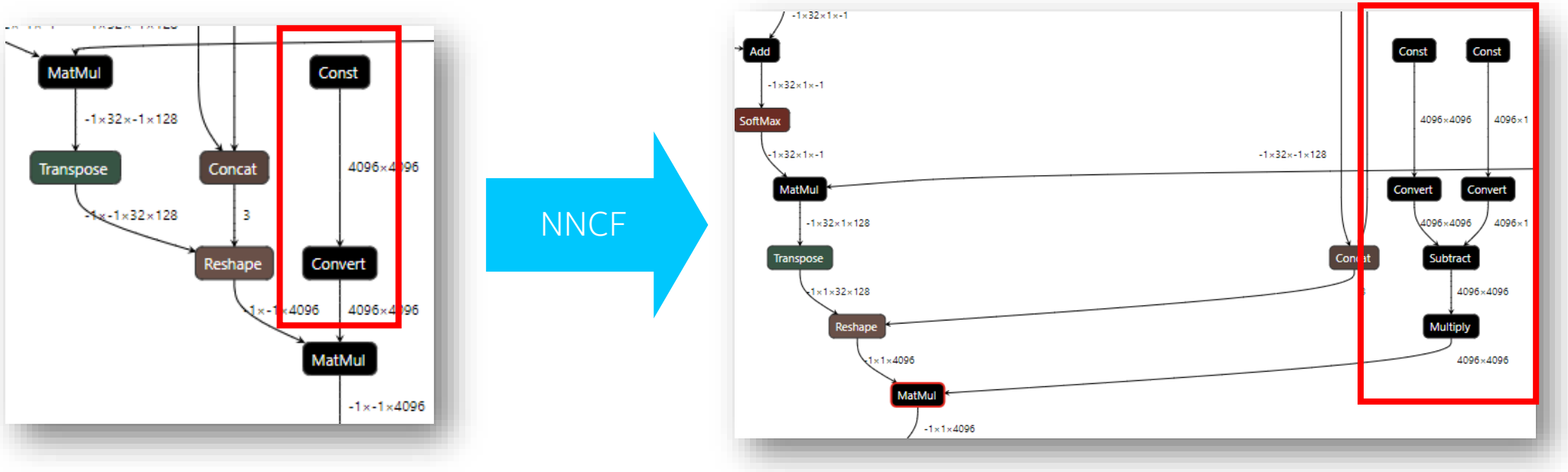
モデルの量子化 (OpenVINO + NNCF)



W = 重み　　X = 入力

重みの圧縮 - Weights Compression

https://github.com/openvinotoolkit/nncf/blob/develop/docs/compression_algorithms/CompressWeights.md




```
from nncf import compress_weights
compressed_model = compress_weights(model)
```

重みの圧縮 - Weights Compression

https://github.com/openvinotoolkit/nncf/blob/develop/docs/compression_algorithms/CompressWeights.md

Model	Size (GB) Reduction
`llama-2-7b-chat`	25 → 6
`open-llama-3b`	13 → 3
`dolly-v2-12b`	44 → 11
`gpt-neox-20b`	77 → 19
`llama-7b`	25 → 6
`gpt-j-6b`	23 → 6

CPU Xeon Gold 6338

OpenVINO 2023.1

生成系AI導入がしやすくなりました

簡素化された
ワークフロー

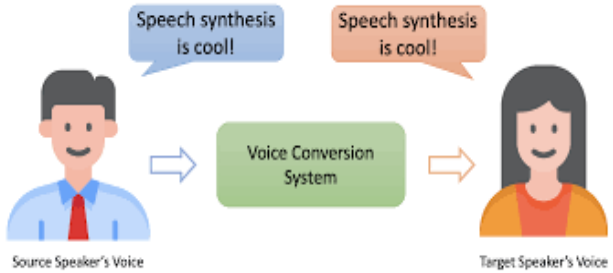
PyTorchとの親和性向上

大規模モデルの最適化

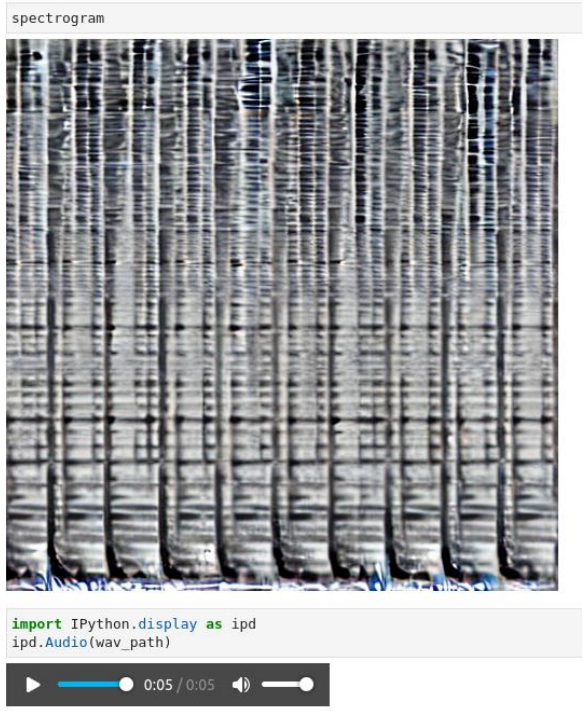
最新のnotebook
サンプルプログラム

オーディオ生成

FreeVC: text-free voice conversion



Riffusion: Text-to-audio spectrogram

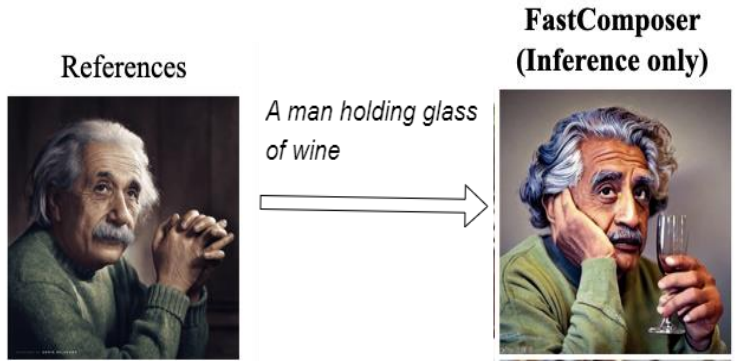


MusicGen: Text-to-Music

イメージ生成

Stable Diffusion XL

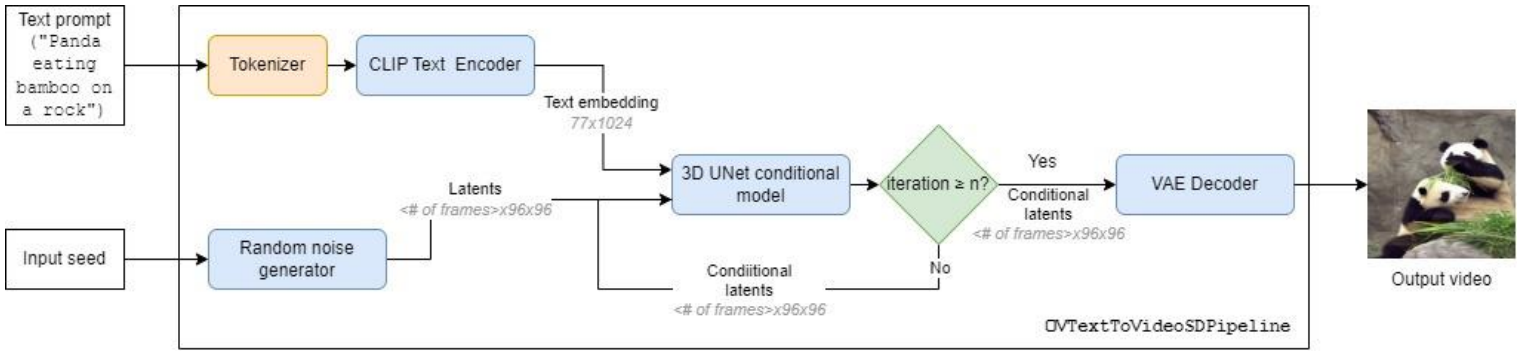
FastComposer: generation personalized images without model fine-tuning



TinySD

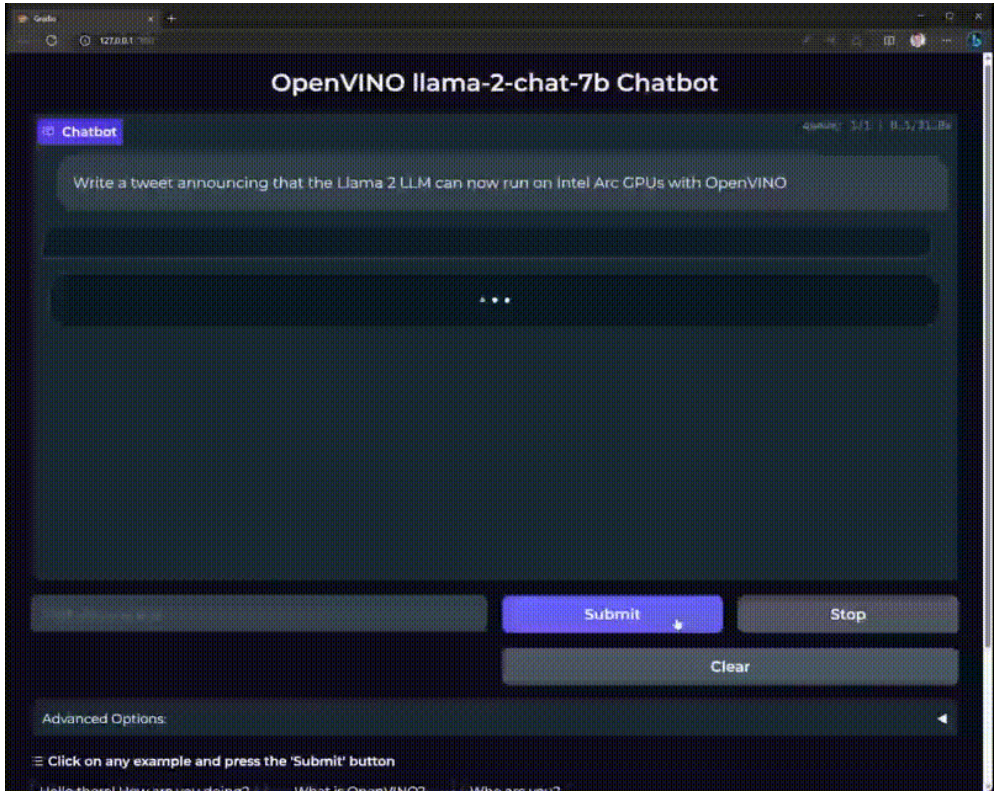
ムービー生成

ZeroScope: Text-to-Video generation

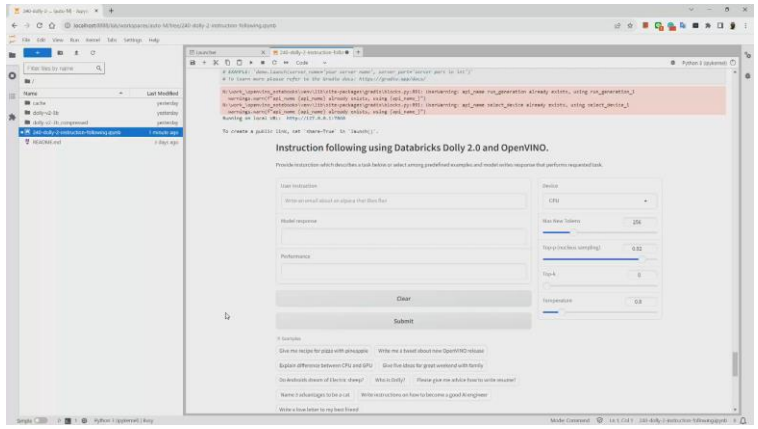


大規模言語モデル(LLM)チャットボット

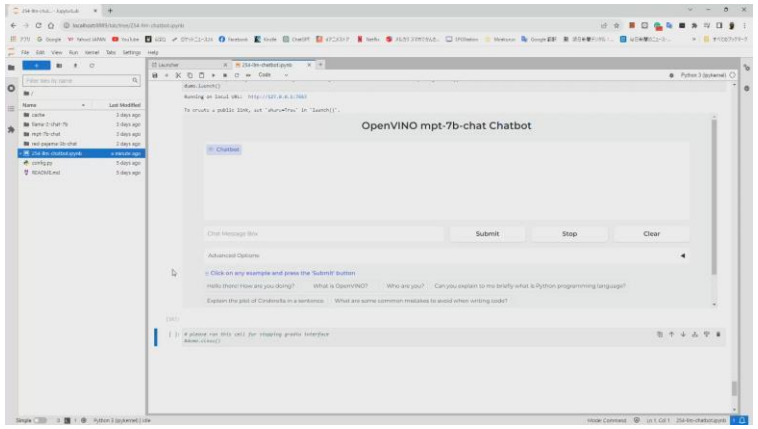
- LLAMA2を含む多くのLLMモデルをサポート
- CPUでもインテルGPUでも推論実行可能
- INT8重み圧縮の手順なども提示



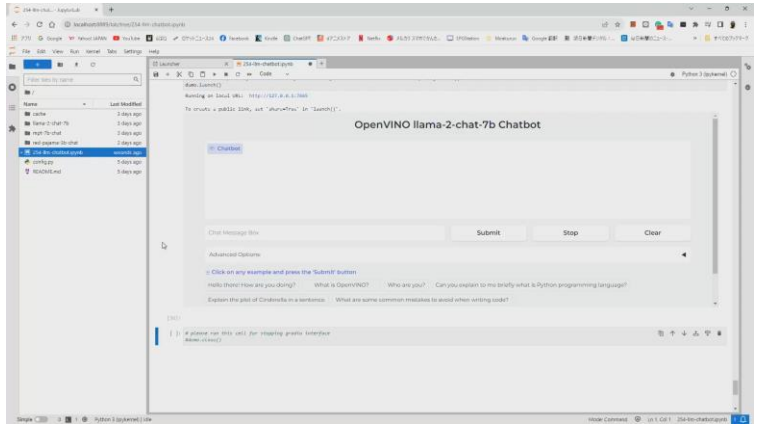
Demo



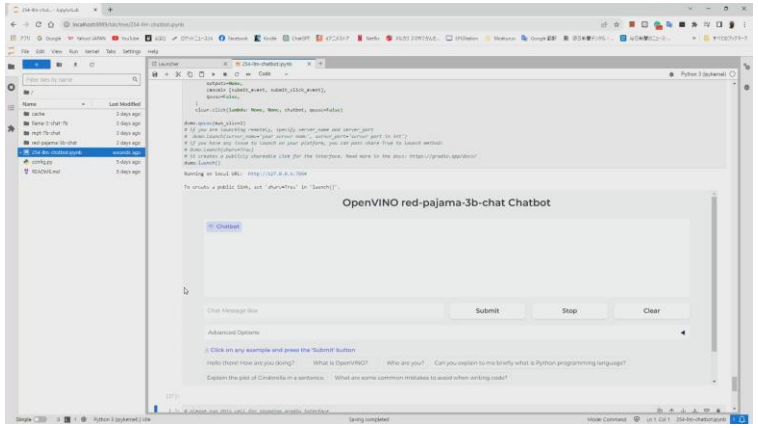
240-dolly-2-instruction-following
CPU i7-10700K, 64GB, ov23.1



254-llm-chatbot, mpt-7b-chat
CPU i7-10700K, 64GB, ov23.1



254-llm-chatbot, llama-2-7b-chat
CPU i7-10700K, 64GB, ov23.1



254-llm-chatbot, red-pajama-3b-chat
CPU i7-10700K, 64GB, ov23.1

HW: Core i7-10700K, RAM 64GB,
SW: Windows 11 23H2, OpenVINO 2023.1.0

OpenVINO™ Notebooks https://github.com/openvinotoolkit/openvino_notebooks

OPENVINO 2023.2.0 HAS JUST BEEN RELEASED!!

緊急追加!!
訳が適当で申し訳ありません

- Officiel Release Notes
  - https://www.intel.com/content/www/us/en/developer/articles/release-notes/openvino/2023-2.html
- **Summary of major features and improvements**

- **More Generative AI coverage and framework integrations to minimize code changes.**
  - **PyTorch直接サポートモデルの拡充 - torch.compile()**
  - **注目モデルサポートの拡充 –** LLaVA, chatGLM, Bark (text to audio), and LCM (Latent Consistency Models, an optimized version of Stable Diffusion).
  - **Huggingfaceモデルサポートの容易化 –** Huggingface CLIで簡単にOpenVINO IRに変換
  - **Conanパッケージマネージャーでの配布 –** for C and C++ developers.

- **Broader Large Language Model (LLM) support and more model compression techniques.**
  - **CPU, iGPU上でのint8 LLMモデルのパフォーマンス向上**
  - **GPUでのダイナミックシェイプサポートモデルの追加**
  - **[PREVIEW] CPU, iGPUでのint4重みモデルのサポート (Llama2, GML2など)**
  - **The following Int4 model compression formats are supported for inference in runtime**:
    - Generative Pre-training Transformer Quantization (GPTQ); with GPTQ-compressed models, you can access them through the Hugging Face repositories.
    - Native Int4 compression through Neural Network Compression Framework (NNCF).
- **More portability and performance to run AI at the edge, in the cloud, or locally.**
  - **ARMプロセッササポートの強化 - 最適化、パフォーマンス向上**

Notices and disclaimers

Intel is committed to respecting human rights and avoiding complicity in human rights abuses. See Intel's Global Human Rights Principles. Intel® products and software are intended only to be used in applications that do not cause or contribute to a violation of an internationally recognized human right.

Intel® technologies may require enabled hardware, software, or service activation. No product or component can be absolutely secure. Your costs and results may vary.