Frameworks

MindSpore (Huawei)

Prof. Carlos Moraes (Carlão)

Projeto Residência 8 (TPA N° 068/SOFTEX/UNIFEI)

MindSpore

Multi-domain expansion

Model Zoo (Model library)

Vision | NLP | Model Hub

Audio | Rec

Extend

Reinforcement learning, GNN, deep probabilistic programming...

Scientific computing

Electromagnetic simulation, molecular simulation, quantum computing

Insight

Network debugging

Performance tuning

Accuracy tuning

Developer-friendly

Expression (Unified APIs for all scenarios)

Function/OOP Programming Paradigm Fusion | AI+ numerical calculation and expression fusion | dynamic and static expression unification | single-server distributed expression unification

NN | Ops | Dataset | Train/Infer | Numpy

Third-party frontend

Cangjie | Julia

Efficient running

Data

Data loading

Data format

Heterogeneous acceleration

Data augmentation

AI compiler

Type derivation | Auto differentiation | Auto parallelism | Second-order optimization

Memory optimization | Graph kernel fusion | Pipeline execution

Quantization, pruning, distillation

Intermediate representation (IR) layer

Hardware optimization | Automatic operator generation

Armour

AI security

Encrypted AI

Trustworthy AI

All-scenario deployment

Runtime

Runtime system for all device-edge-cloud scenarios

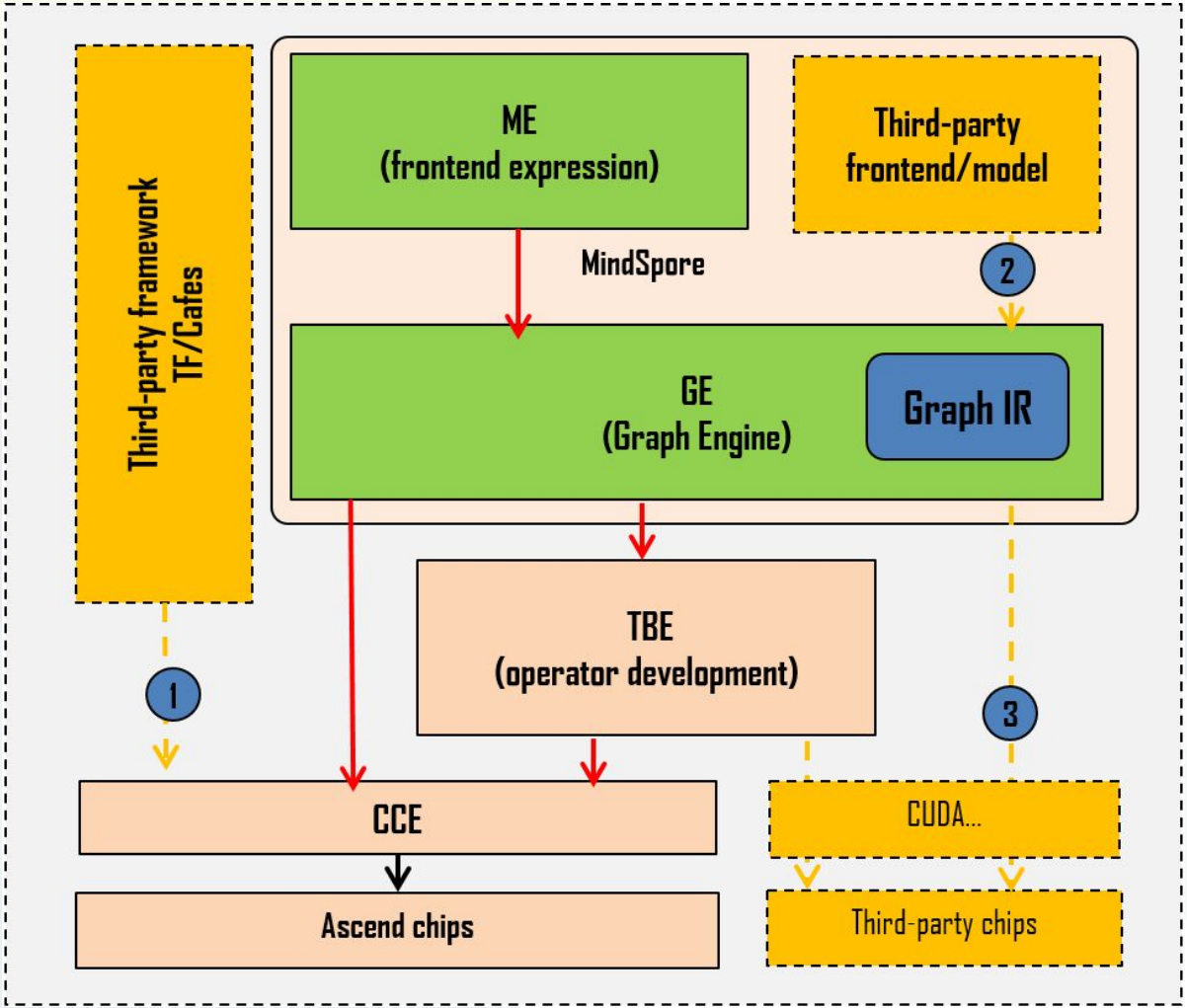
Diversified hardware

CANN(NPU) | CUDA(GPU) | Eigen(CPU) | oneDNN(CPU)

Cloud

Edge

Device


Third-party framework
TF/Cafes
ME
(frontend expression)
Third-party
frontend/model
MindSpore
2
GE
(Graph Engine)
Graph IR
TBE
(operator development)
1
3
CCE
CUDA...
Ascend chips
Third-party chips

MindSpore

Unified APIs for all scenarios

Auto differ | Auto parallelism | Auto tuning

MindSpore intermediate representation (IR) for computational graph

On-device execution | Pipeline parallelism | Deep graph optimization

Device-edge-cloud co-distributed architecture (deployment, scheduling, communications, etc.)

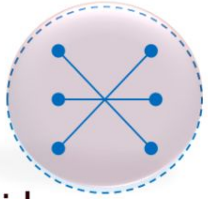
Easy development:
AI Algorithm As Code

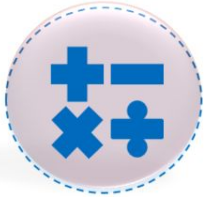
Efficient execution:
Optimized for Ascend
GPU support

Flexible deployment: on-demand cooperation across all scenarios

Processors: Ascend, GPU, and CPU

Diferenciação Automática





Technical path
of automatic differential

Graph: TensorFlow

- Non-Python programming based on graphs
- Complex representation of control flows and higher-order derivatives

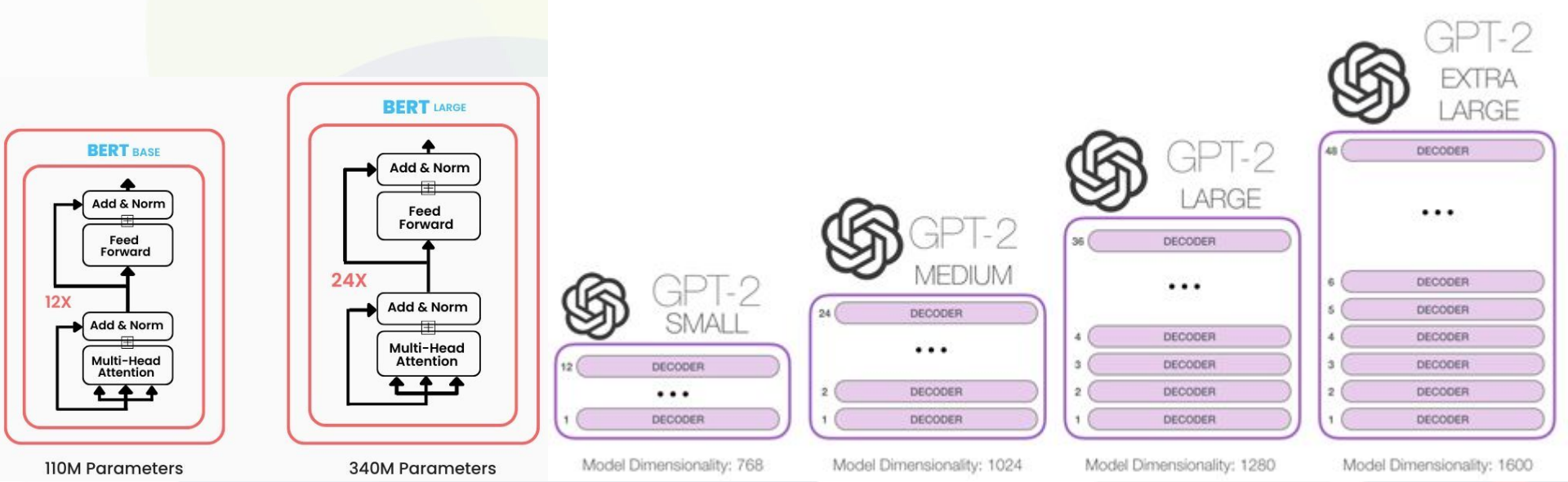
Operator overloading: PyTorch

- Runtime overhead
- Backward process performance is difficult to optimize.

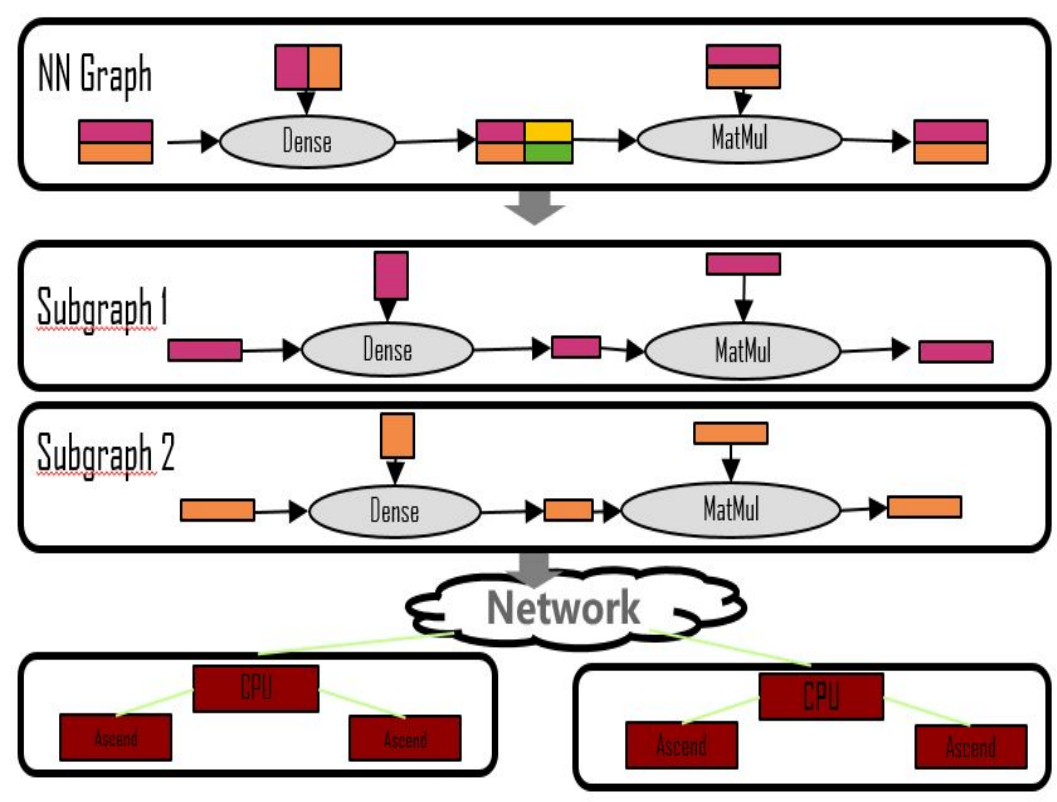
Source code transfer: MindSpore

- Python APIs for higher efficiency
- IR-based compilation optimization for better performance

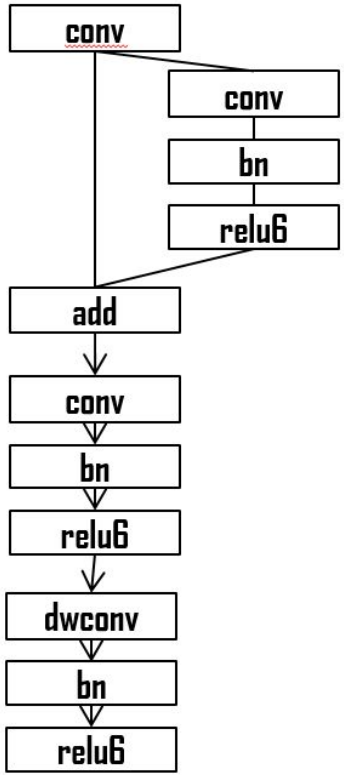
Paralelismo Automático - Desafios

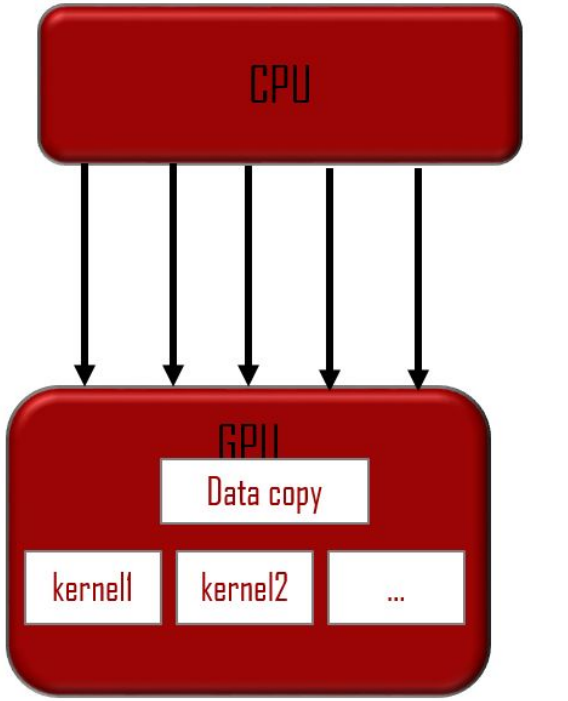


Paralelismo Automático - Tecnologias



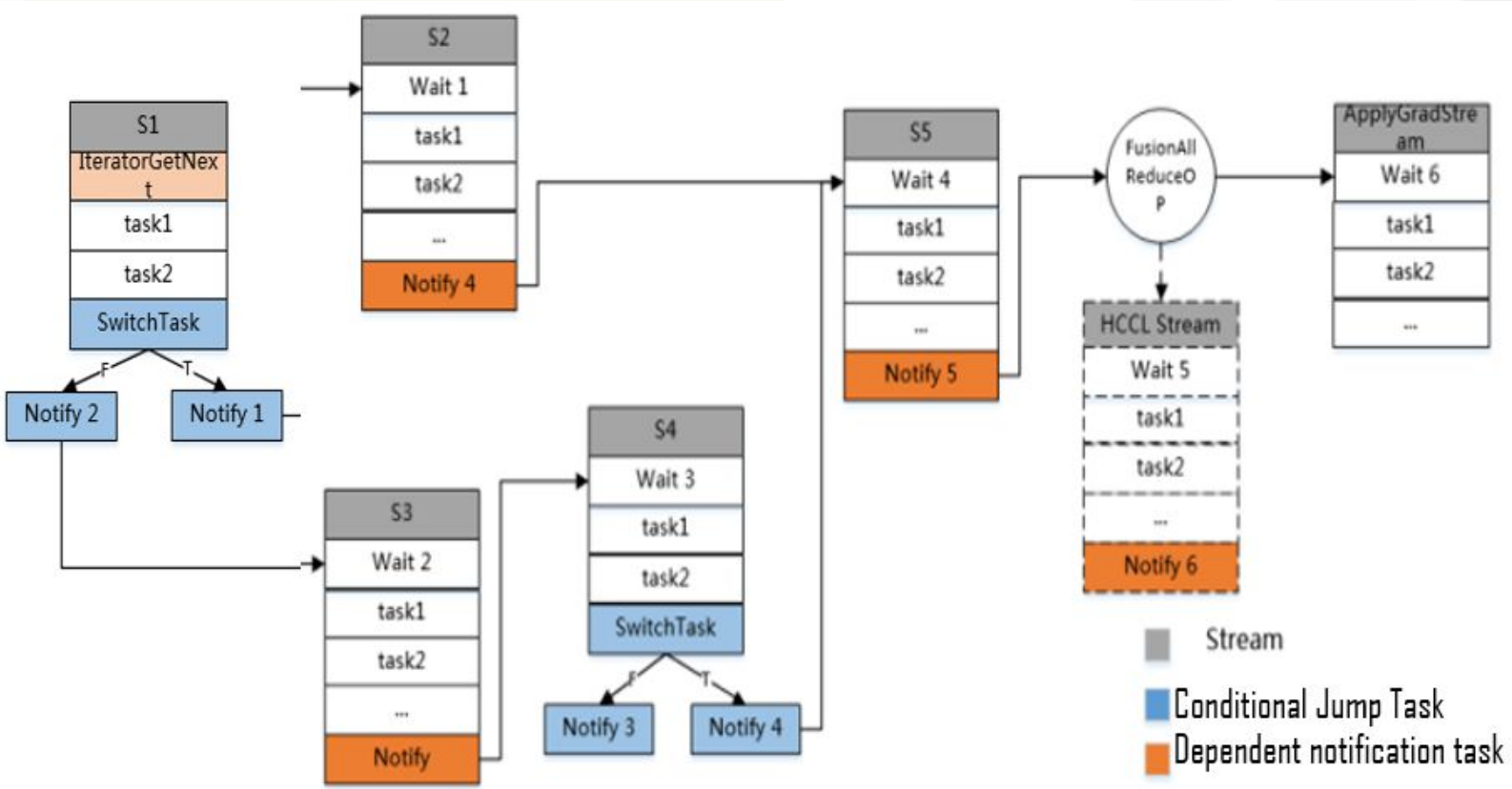
Execução no Dispositivo



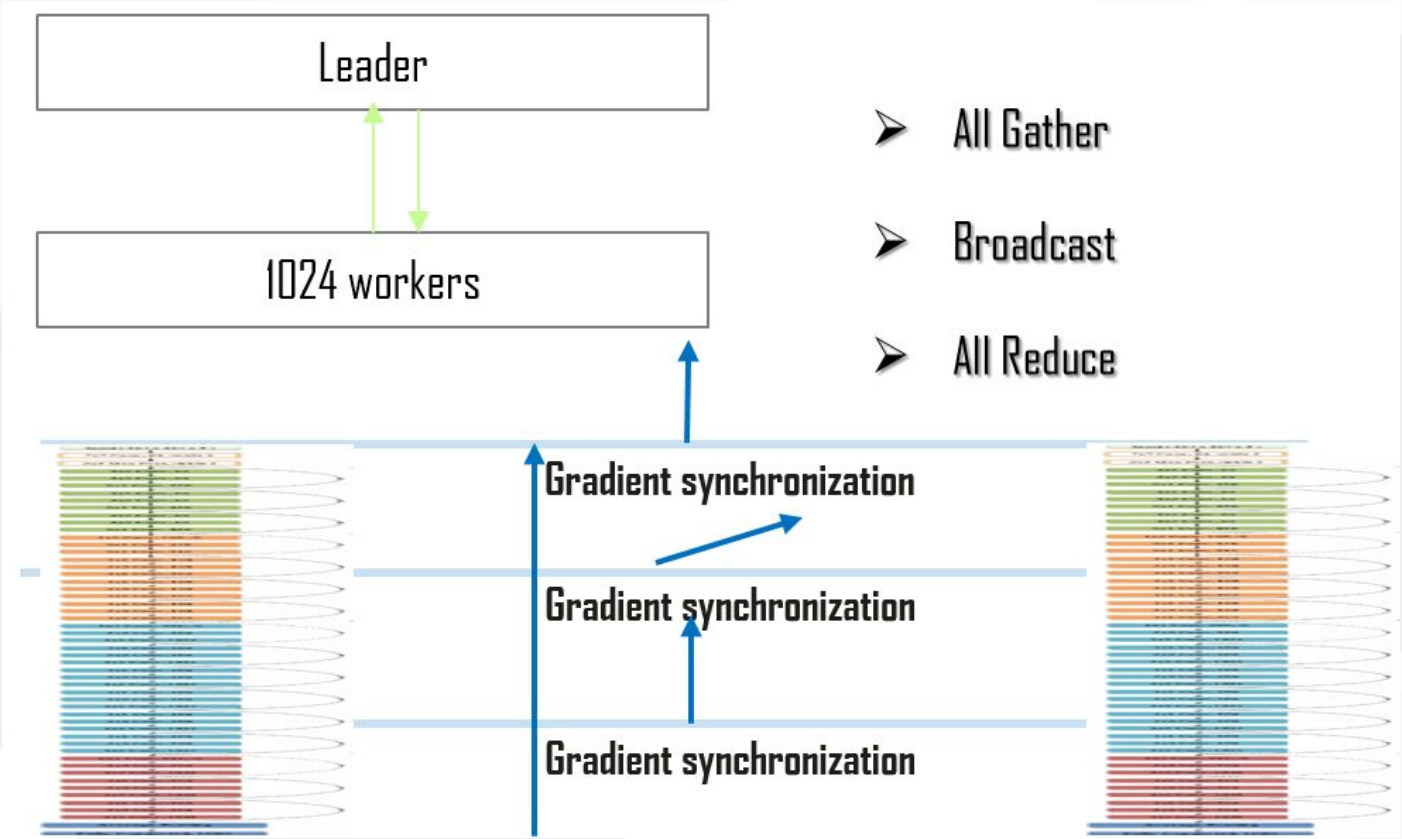


Large data interaction overhead and difficult data supply

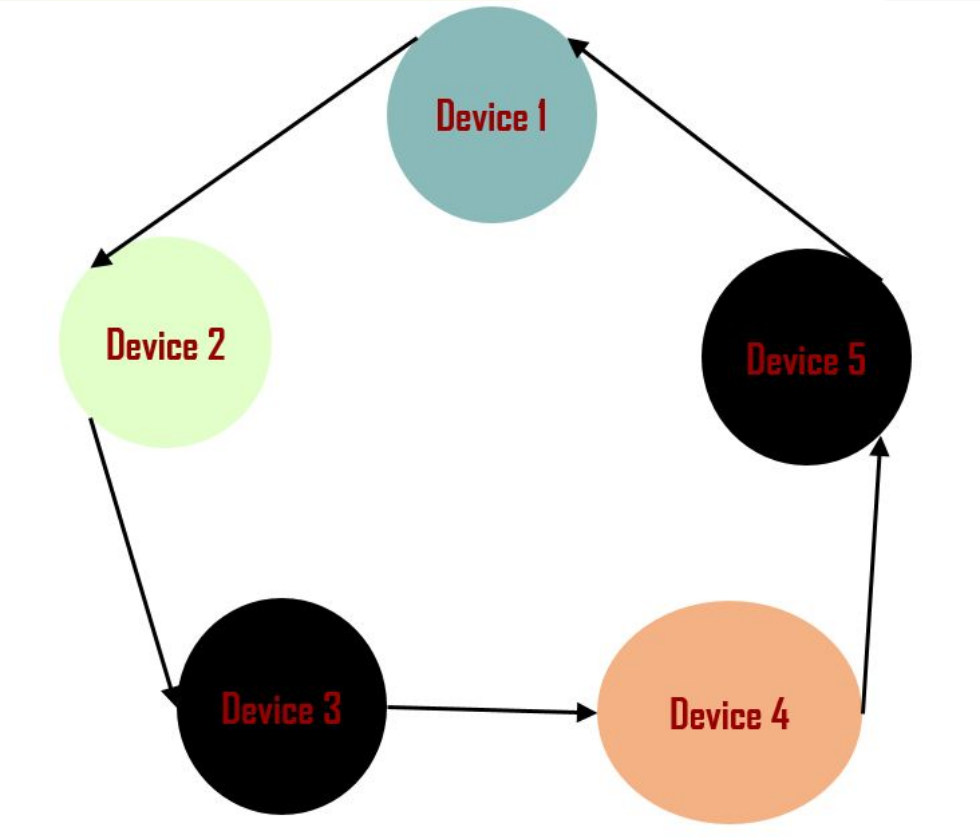
Execução no Dispositivo - Tecnologias



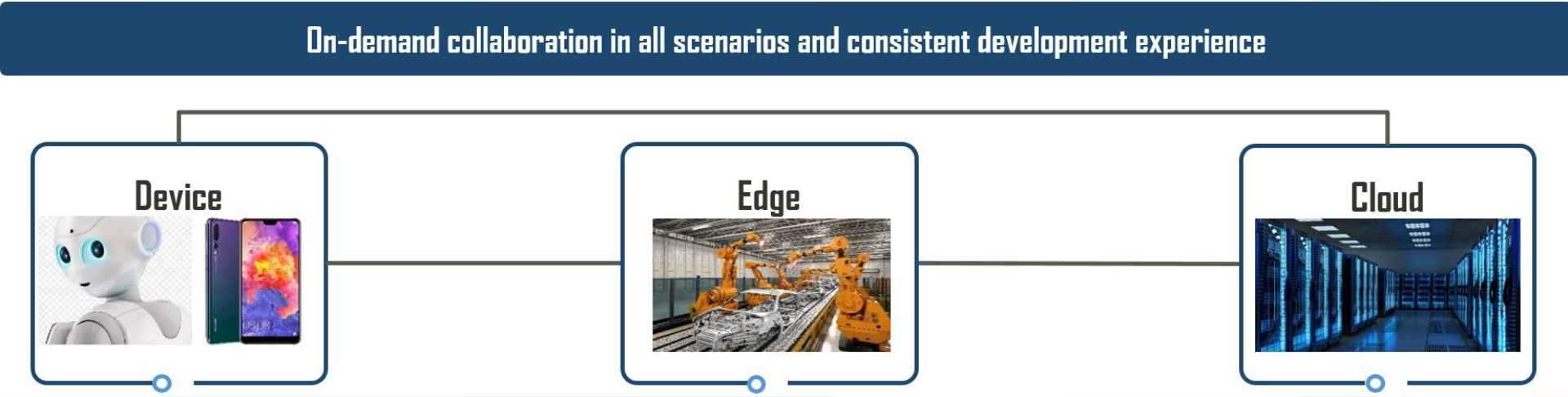
Desafios para Agregação de Gradiente Distribuído



Tecnologias para Agregação de Gradiente Distribuído



Arquitetura Distribuída Device-Edge-Cloud



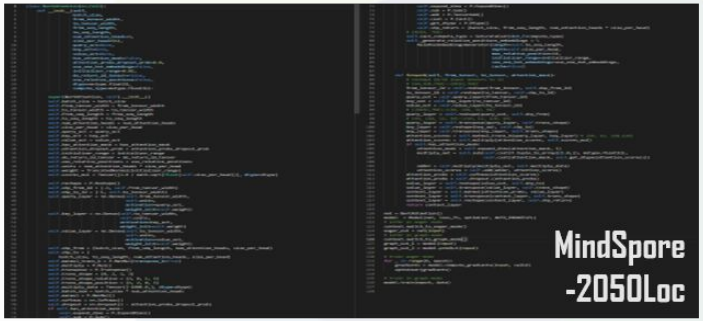
Características

AI Computing Framework: Desafios

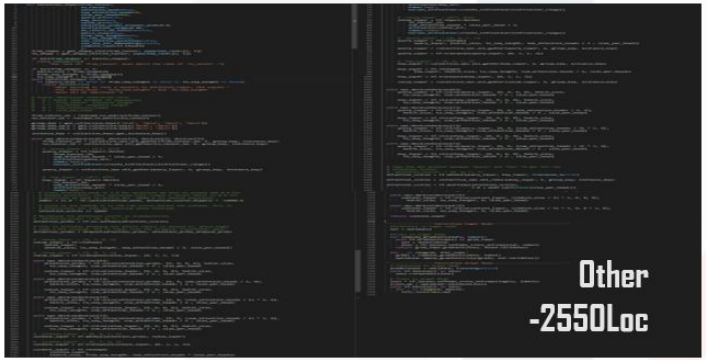
Desafios da Indústria	Inovação tecnológica
Uma enorme lacuna entre pesquisa da indústria e aplicação de IA em todos os cenários ● Barreiras de entrada altas ● Alto custo de execução ● Longa duração de implantação	**MindSpore facilita IA inclusiva em todas as aplicações** ● Novo modo de programação ● Novo modo de execução ● Novo modo de colaboração

Novo Paradigma de Programação

Algorithm scientist



Algorithm scientist
+
Experienced system developer



One-line
Automatic parallelism

Efficient automatic differential

One-line debug-mode switch

NLP Model: Transformer

TensorFlow vs MindSpore

```
import tensorflow as tf

model() {
    with tf.device("/device:0")
        token_type_table = tf.get_variable(
            name=token_type_embedding_name,
        shape=[token_type_vocab_size, width],
        initializer=create_initializer(initializer_range))
        flat_token_type_ids = tf.reshape(token_type_ids, [-1])
        one_hot_ids = tf.one_hot(flat_token_type_ids, depth=token_type_vocab_size)
        token_type_embeddings = tf.matmul(one_hot_ids, token_type_table)

    with tf.device("/device:1")
        query_layer = tf.layers.dense(
            from_tensor_2d,
            num_attention_heads * size_per_head,
            activation=query_act,
            name="query",
            kernel_initializer=create_initializer(initializer_range))

    with tf.device("/device:2")
        key_layer = tf.layers.dense(
            to_tensor_2d,
            num_attention_heads * size_per_head,
            activation=key_act,
            name="key",
            kernel_initializer=create_initializer(initializer_range))
```

```
class DenseMatMulNet(nn.Cell):
    def __init__(self):
        super(DenseMutMulNet, self).__init__()
        self.matmul1 = ops.MatMul.set_strategy({[4, 1], [1, 1]})
        self.matmul2 = ops.MatMul.set_strategy({[1, 1], [1, 4]})
    def construct(self, x, w, v):
        y = self.matmul1(x, w)
        z = self.matmul2(y, v)
        retorno s
```

Desafios de Execução

On-device execution

Offloads graphs to devices, maximizing the computing power of Ascend

Framework optimization

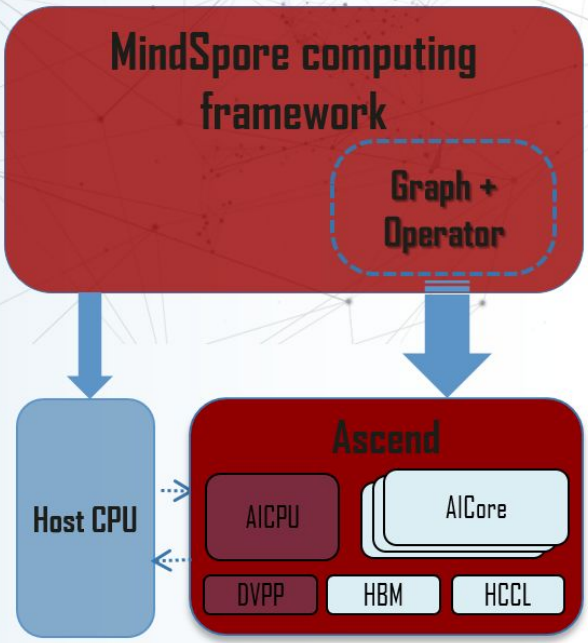
Pipeline parallelism

Cross-layer memory overcommitment

Soft/hardware co-optimization

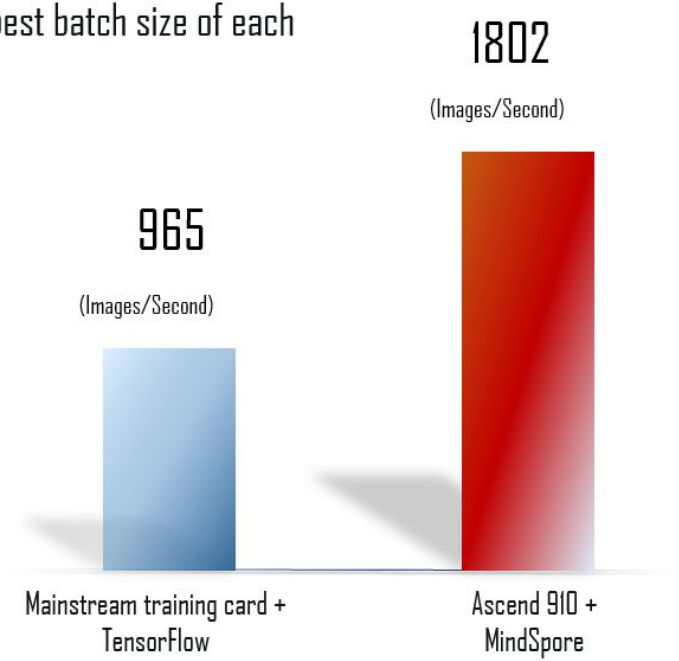
On-device execution

Deep graph optimization



O desempenho do ResNet-50 é dobrado

- ResNet 50 V1.5
- ImageNet 2012
- With the best batch size of each card



Reconhecimento de vários objetos em tempo real

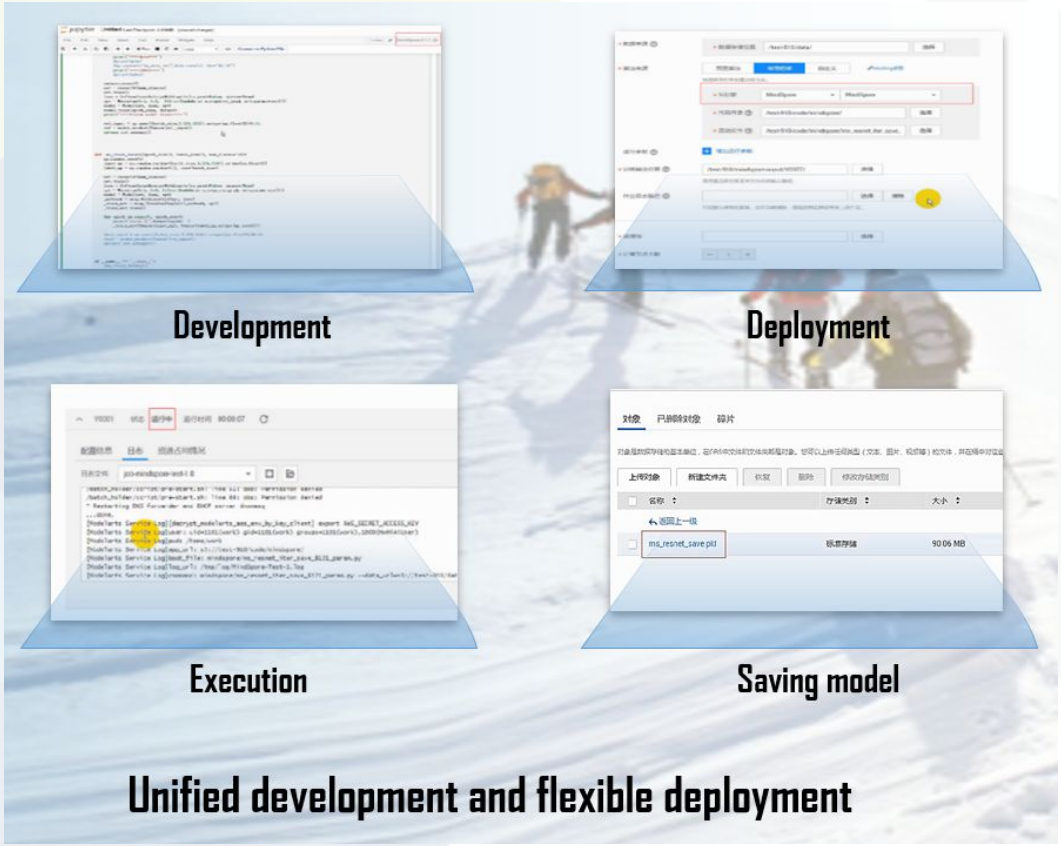
Detecting objects in 60 ms

Novo modo de colaboração

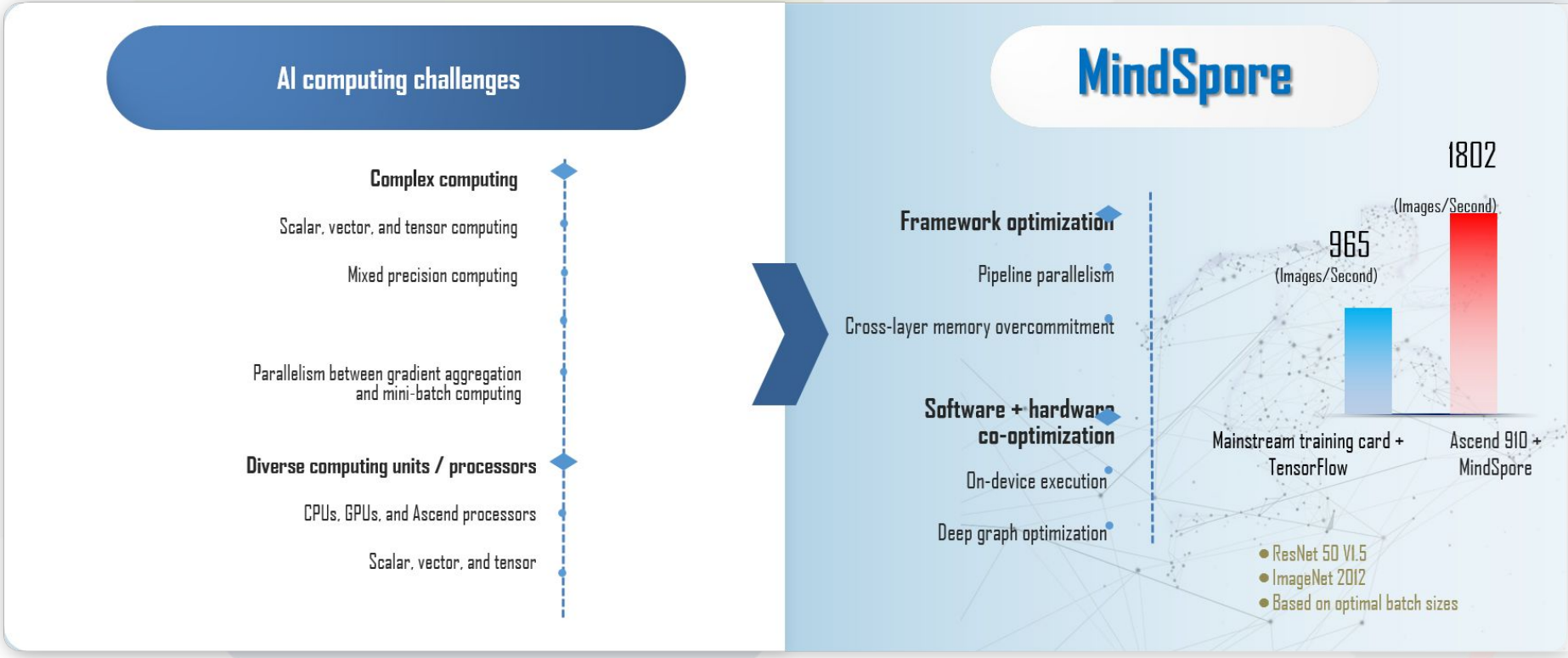
V.S.

Diferentes precisão e velocidade de hardware

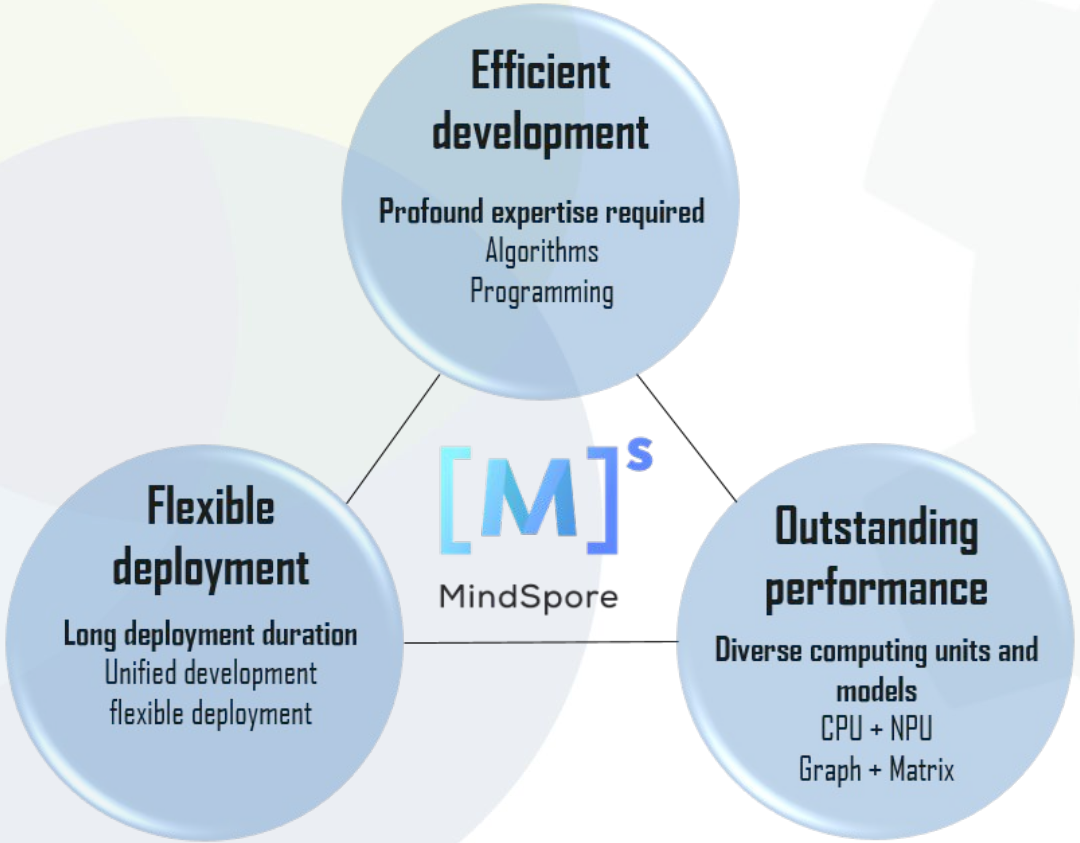
Desenvolvimento unificado e Implantação flexível



Alta performance

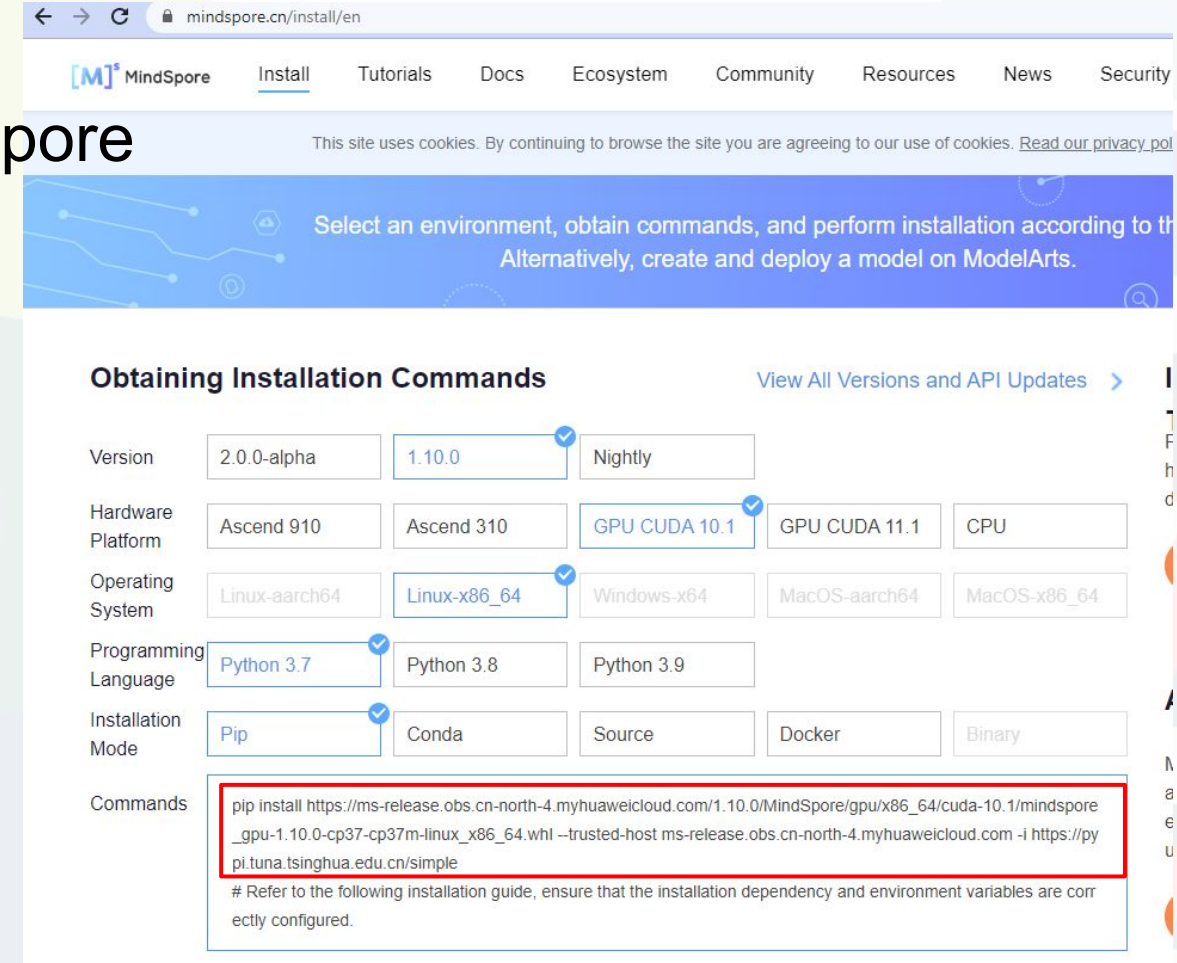


Visão e valor da MindSpore



Desenvolvimento e Aplicação

Instalando o MindSpore



Módulos

model_zoo	Define modelos de rede comuns
communication	Módulo de carregamento de dados, que define o dataloader e dataset e processa dados como imagens e textos.
dataset	Módulo de processamento de dataset, que lê e pré-processar dados.
common	Define tensor, parâmetro, dtype e inicializador.
context	Define a classe de contexto e define os parâmetros de execução do modelo, como grafos e PyNative modos de comutação.

Módulos

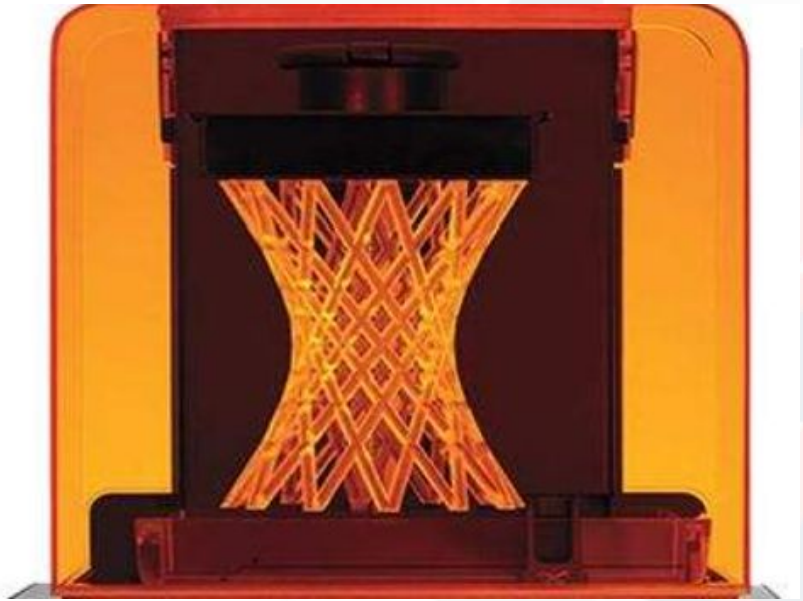
akg	Diferencial automático e biblioteca de operador personalizado.
nn	Define células MindSpore (unidades de rede neural), funções de perda e otimizadores.
ops	Define operadores básicos e registra operadores reversos.
train	Modelo de treinamento e módulos de função de resumo.
utils	Utilitários, que verificam os parâmetros. Este parâmetro é usado na estrutura.

Operações com Tensores

- asnumpy()
- size()
- dim ()
- dtype()
- set_dtype()
- tensor_add(Tensor)
- tensor_mul(Tensor)
- shape()
- __Str__ # (conversão em strings)

Célula MindSpore

```
class Net(nn.Cell):
    def __init__():
        ......
    def construct(self, x):
        ......
```

Pergunta

Which of the following is a math operator in MindSpore?

- Mul
- Select
- Callback
- ControlDepend

Pergunta

Automatic graph segmentation is one of the key technologies of MindSpore.

- Operator-level automatic differentiation
- Automatic parallelization
- Automatic operator generation
- Semi-automatic data labeling

Pergunta

Facing industry research and full scenarios AI The huge gap between applications, MindSpore Bridging the application gap to help inclusiveness AI of technological innovation does not include which of the following?

- New programming language
- New ways of collaboration
- New programming paradigm
- New execution mode

Pergunta

In MindSpore, if you want to define a function for collecting information after each step is complete, you need to inherit the [?] class.

Pergunta

The [?] subsystem of MindSpore provides Mind IR-oriented, graph-level, real-time compilation.

Apoio

Este projeto é apoiado pelo Ministério da Ciência, Tecnologia e Inovações, com recursos da Lei nº 8.248, de 23 de outubro de 1991, no âmbito do [PPI-Softex| PNM-Design], coordenado pela Softex.

Carlos Henrique Valério de Moraes
Universidade Federal de Itajubá
e-mail: valerio@unifei.edu.br

UNIFEI Softex MCTI FUTURO GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO