

HUGECTR - 端到端点击率预估训练 解决方案介绍（一）

15 Nov 2019



AGENDA

Click-Through Rate Prediction

Challenges in CTR Training

HugeCTR Introduction

CLICK-THROUGH RATE PREDICTION

WHAT IS CTR

Wikipedia:

“**Click-through rate (CTR)** is the ratio of users who click on a specific link to the number of total users who view a page, email, or advertisement.”

Relatives:

Data Mining, Learning to Rank, NLP, CV

APPLICATIONS

Search Advertising

Recommend based on input query && advs && user information

Baidu 百度 韩语自学  **百度一下**

网页 资讯 视频 图片 知道 文库 贴吧 采购 地图 更多»

百度为您找到相关结果约14,800,000个  搜索工具

[学韩语入门_沪江韩语_从零基础开始学韩语](#)
随时随地在线学韩语,采用延世韩国语,新标准韩国语等全新权威教材,中韩老师联合授课,从零基础入门直达中高级水平,更多线上优惠课程进行中!

[韩语特惠] [沪江网校韩语TOP课程加惠 学霸班实付5.1折起](#)

[韩语机构] [有哪些比较好的学习韩语的机构推荐](#)


[韩语咨询] [零基础学韩语要多少钱 容易学吗 多久学会](#)

[class.hujiang.com](#) 2019-11  - 评价 广告

[零基础如何自学围棋 常用中韩围棋术语整理\(一\)_棋牌_新浪竞技风暴...](#)
 4天前 - 零基础如何自学围棋 常用中韩围棋术语整理(一)2019年11月10日 15:43 ...一、压-밀다(韩语读音:mil da) 定义: 1、双方两子斜向2、落子...
[sports.sina.com.cn/go/...](#) - 百度快照

[我想自学韩语 没有一点基础 要如何学?_百度知道](#)
19个回答 - 回答时间: 2018年8月21日 - 302人觉得有用
最佳答案: 1.理解韩语字结构。对于初学者而言,首先要理解韩语的字结构。所有的韩语字都是A+B或A+B+C的结构,这里的A=辅音,B=元音,C=收音(韵尾)书写的时候都...
[更多关于韩语自学的问题>>](#)
 [百度知道](#) - 百度快照



[韩语自学教程视频_韩语自学教材_韩语自学网](#)
 2017年5月20日 - 沪江专区 提供了韩语自学相关的学习资讯、在线课程等丰富内容;沪江是专业的互联网学习平台,致力于提供便捷优质的网络学习产品和服务。
 [沪江网](#) - 百度快照

淘宝网 Taobao.com 宝贝 羽绒服男  **搜索**

细分风格: 潮 青春活力 商务休闲 基础大众 精致韩风 欧美简约 工装军旅 美式休闲 日韩


筛选条件: 充绒量 含绒量 领型 适用场景 相关分类




您是不是想找: 北面羽绒服 羽绒服女 外套男 羽绒服女中长款 儿童羽绒服 轻薄羽绒服女 羽绒服男阿迪达斯 卫衣男 加拿大

综合排序 销量 信用 价格  


☐ 包邮 ☐ 赠送运费险 ☐ 货到付款 ☐ 新品 ☐ 公益宝贝 ☐ 二手 ☐ 天猫 ☐ 正品保障 ☐ 7+天内退货 更多»



SELECTED




¥1499.00 6人付款
SELECTED思莱德男士抽绳连帽保暖中长款羽绒服外套SI419412559
 selected官方旗舰店 天津
 掌柜热卖 



波司登



¥380.00 1876人付款
波司登轻薄羽绒服男短款2019新款秋冬立领修身潮流外套B90131011
 波司登官方旗舰店 江苏 苏州


波司登



¥999.00 265人付款
波司登羽绒服2019新款男士中长款可脱卸帽冬季保暖休闲B90141005
 波司登官方旗舰店 江苏 苏州


APPLICATIONS

Recommended Ads

Recommend based on advs && user information

 猿辅导小学数学课

广告

图中共有 15 条线段?



图中共有 个三角形?



图中共有 个长方形?



图中共有 个正方形?



小学数学: 49元14次课

仅限11月, 猿辅导寒假特训班, 特训孩子速算技巧, 49元14节, 速抢

小程序

根据浏览, 猜我喜欢

换一组

聚划算 好品质选得力



单店热销 41000+台

送100张塑封膜

全国联保 黑白可选

提前享

¥118.00 ¥196.00

月销: 2487

家用办公都得力 幼儿园老师推荐



优惠价

送塑封膜100张

¥185 A4/A3通用

¥185.00 ¥398.00

月销: 1006

易佰丰 送100张塑封膜



冷热双裱 过塑平整

品牌直营 售后无忧

¥68.00 ¥118.00

月销: 935

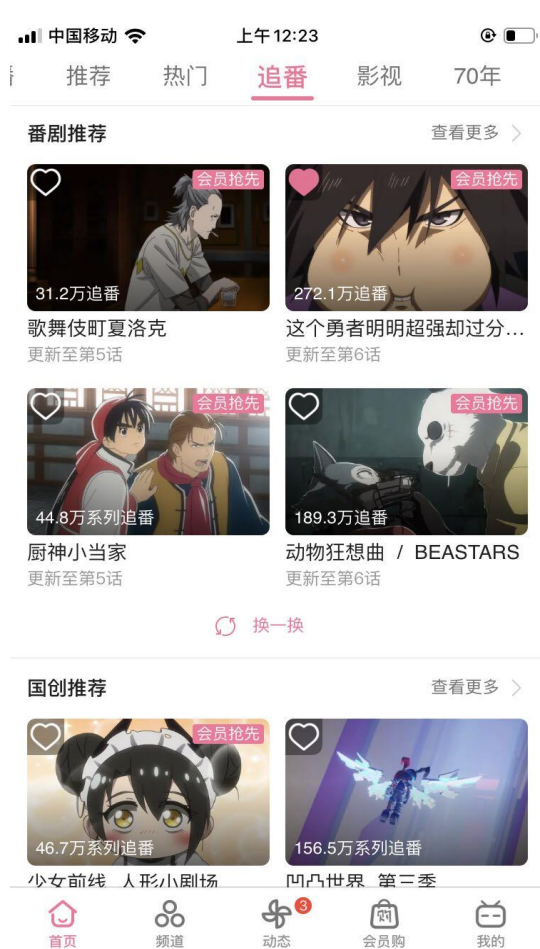
APPLICATIONS

Content Recommendation: UGC

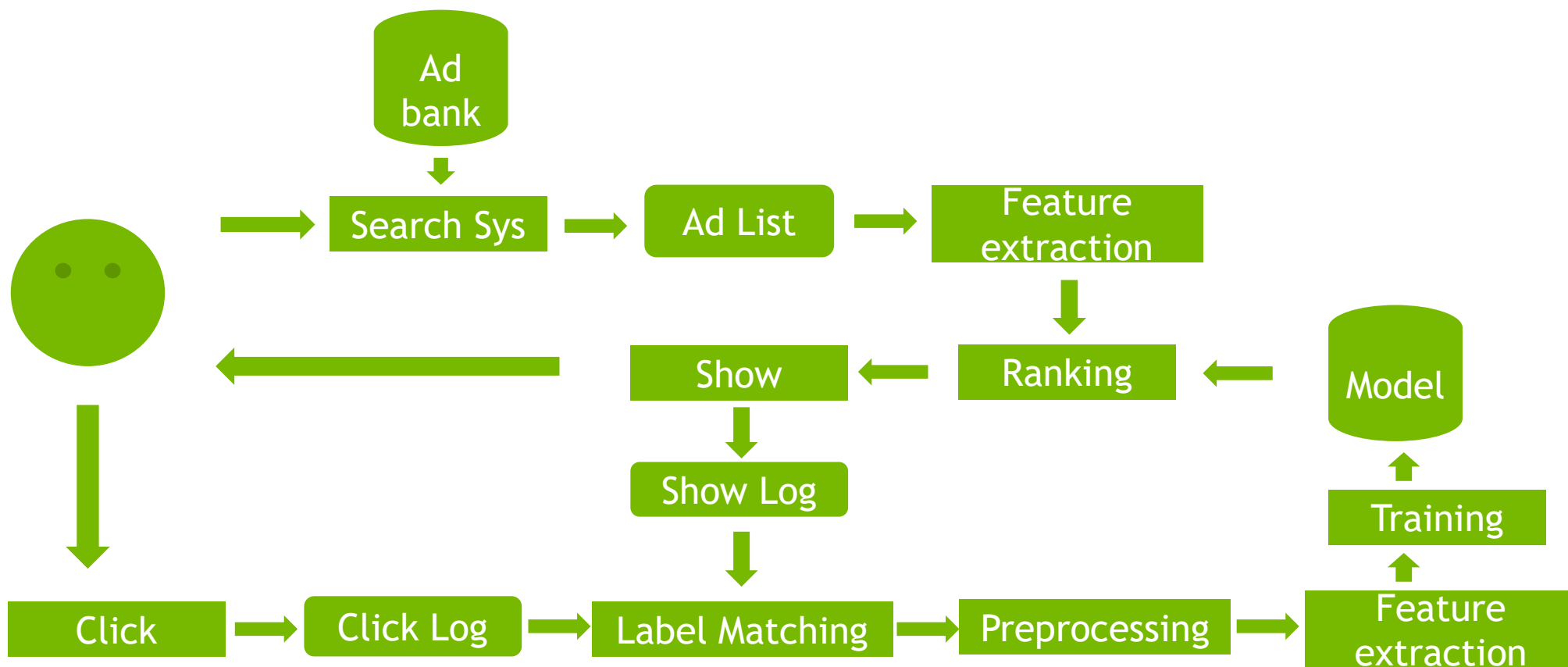


APPLICATIONS

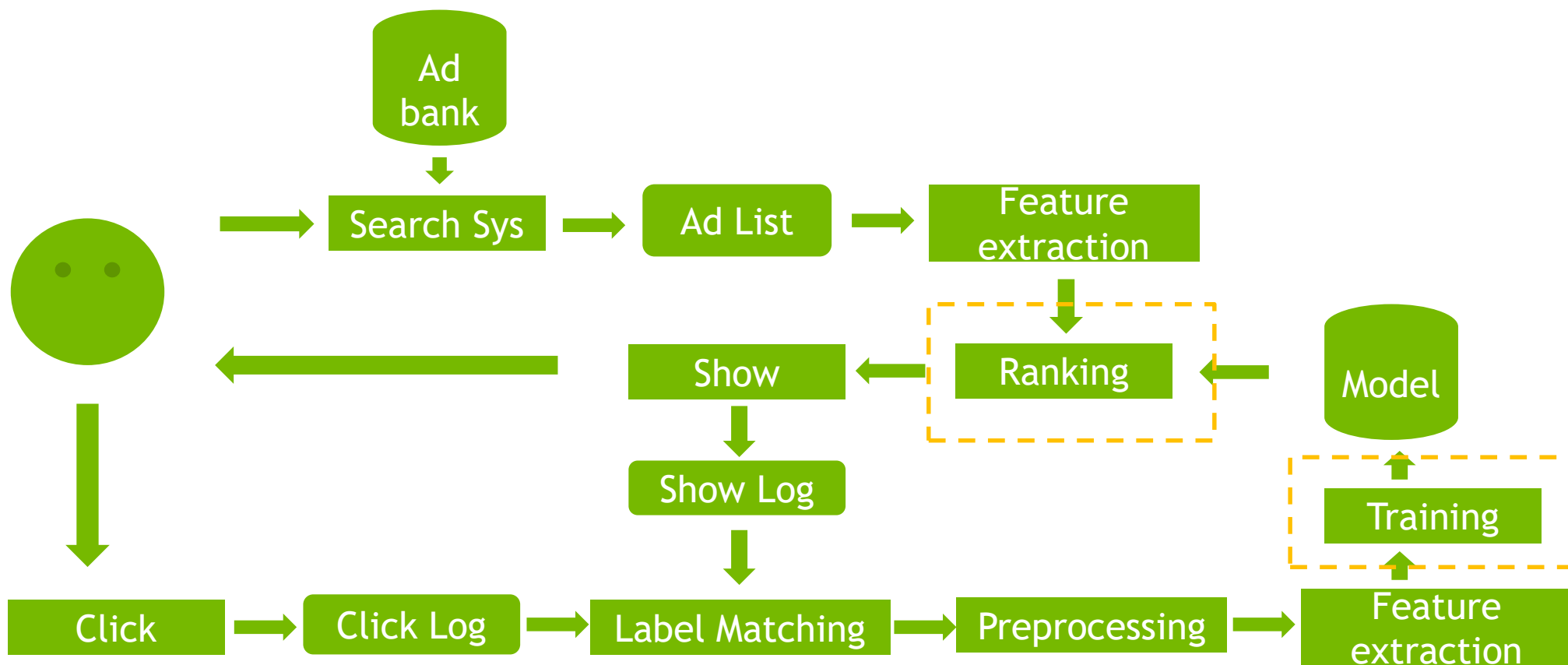
Content Recommendation: PGC



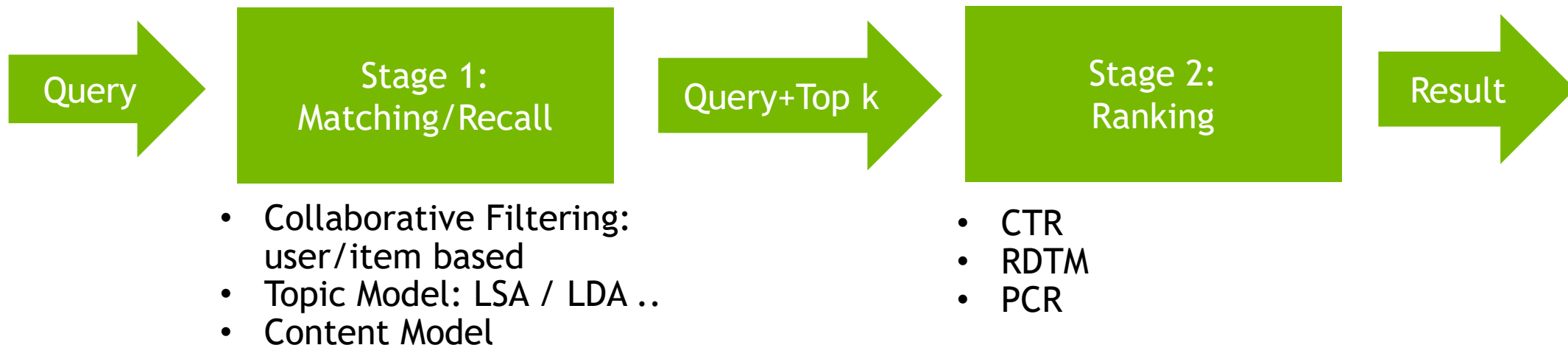
SEARCH ADVERTISING DISTRIBUTION SYSTEM



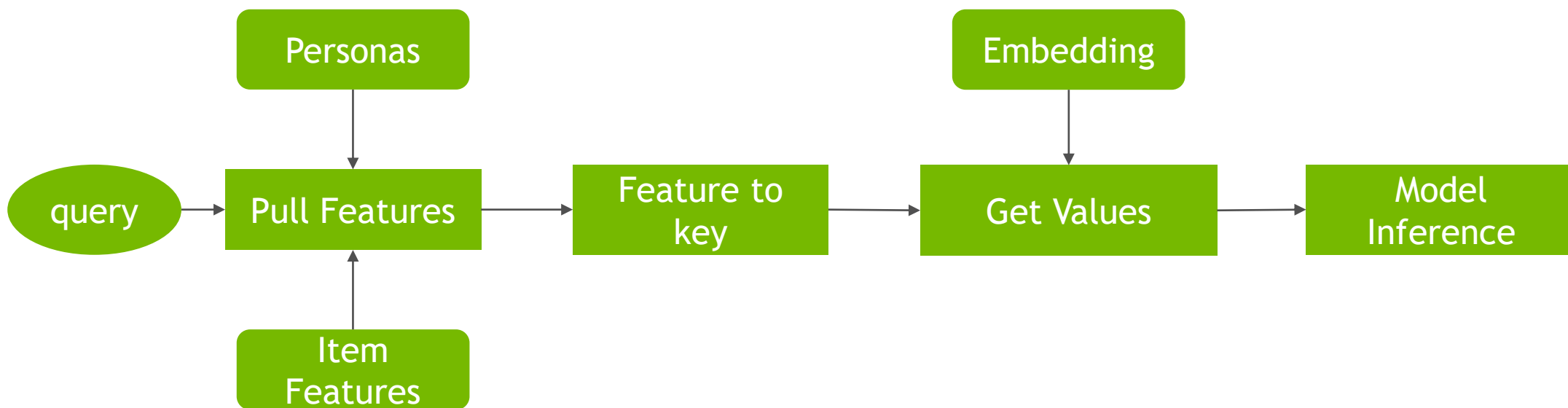
SEARCH ADVERTISING DISTRIBUTION SYSTEM



TWO STAGES RANKING

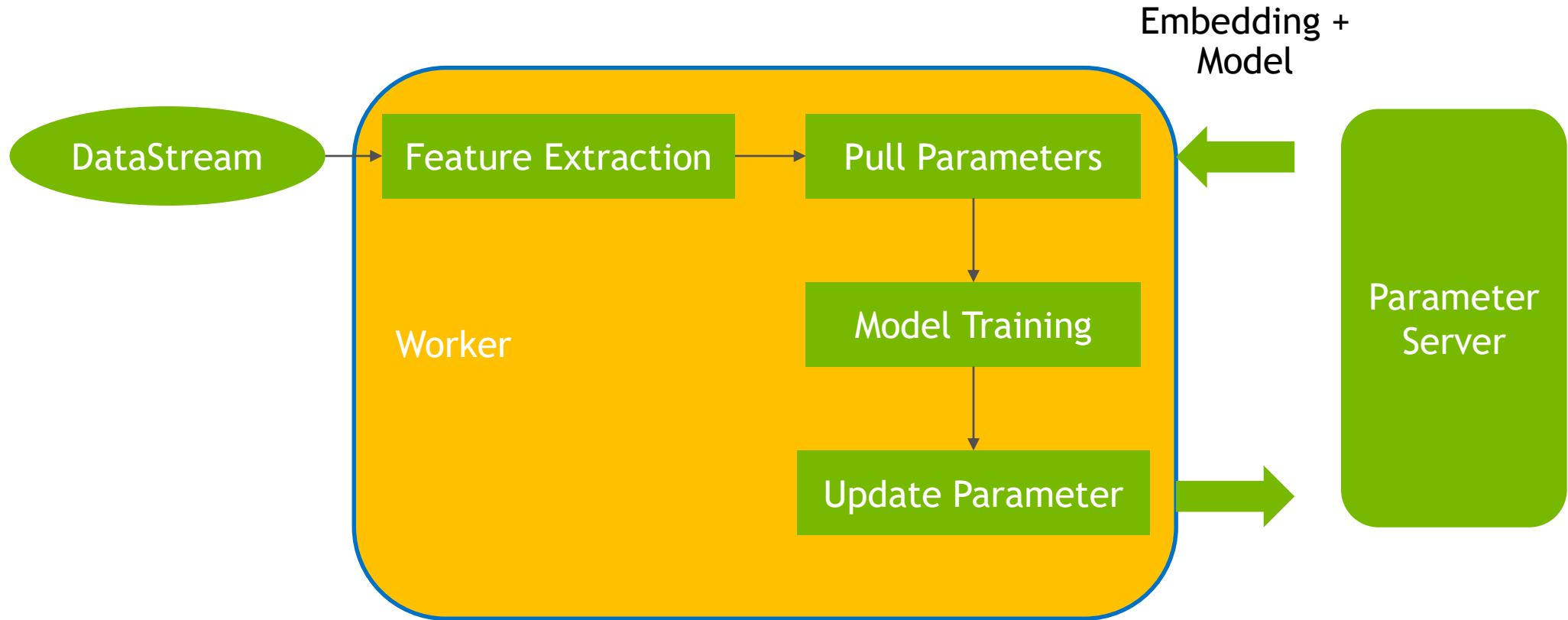


CTR INFERENCE WORKFLOW



CTR TRAINING WORKFLOW

Parameter Server Based



MODEL

Without DNN: Logistic Regression / Factor Machine

With DNN: Embedding+MLP / Wide Deep Learning / DeepFM / DCN / DIN / DIEN

CHALLENGES IN CTR TRAINING

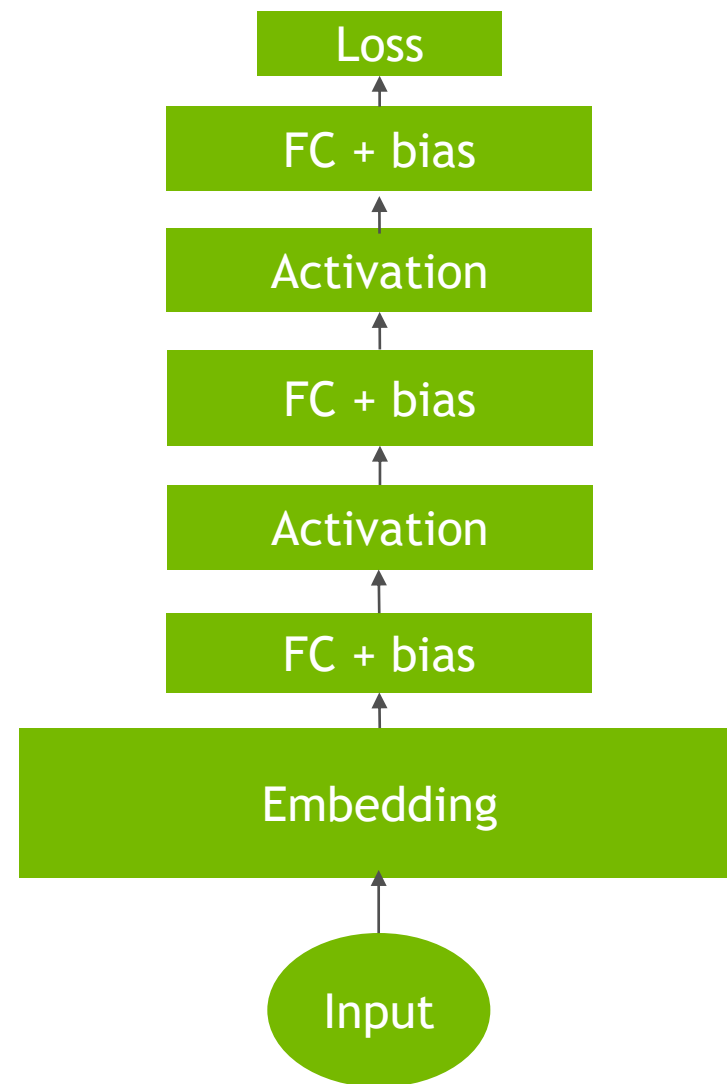
EMBEDDING + MLP

Standard Network

Large Embedding table: E_MEM = GBs to TBs

Small FC layers:

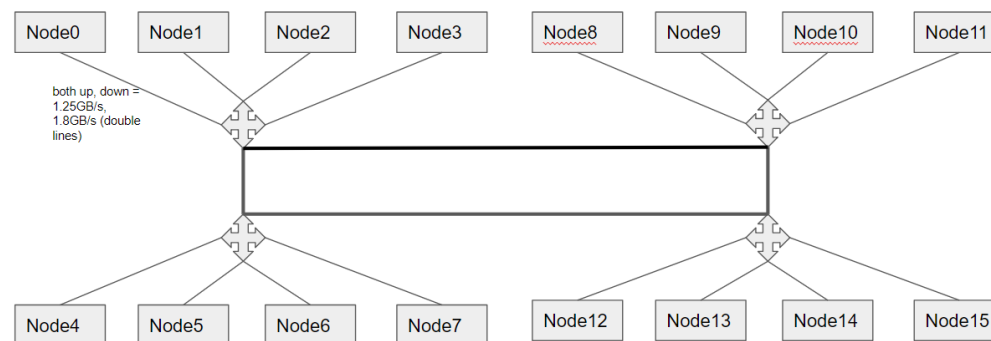
$\text{FC_MEM} = \# \text{Layers} * 100\text{s} * 100\text{s}$
(Suppose $5 * 500 * 500 * 4\text{B} = 5\text{MB}$)



CTR SOLUTION

CPU

- ▶ 100 Nodes, connected with Ethernet (1.25-1.8GB/s)
- ▶ Each forward/backward exchange whole the dense model ~10MB per node: **5.6ms***
- ▶ Compute time = ~2ms (BS=2000)
- ▶ Overall time = compute + data exchange = 7.6ms

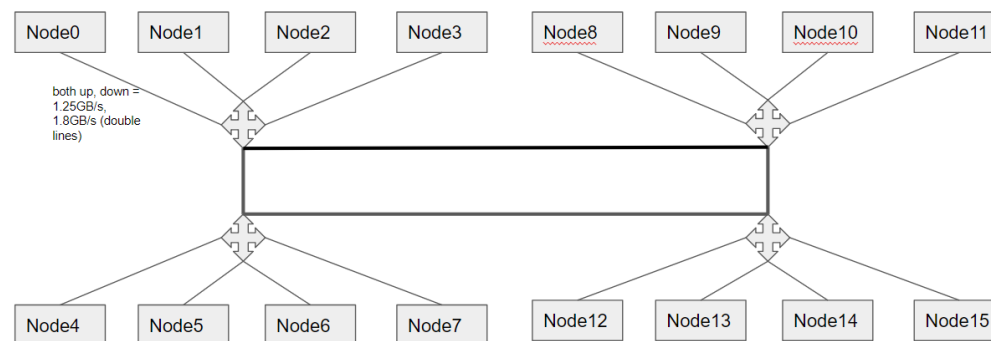


* Suppose 1.8GB/s Ethernet and CPU with 6TFlops per node

CTR SOLUTION

CPU

- ▶ 100 Nodes, connected with Ethernet (1.25-1.8GB/s)
- ▶ Each forward/backward exchange whole the dense model ~10MB per node: **5.6ms**
- ▶ Compute time = ~2ms (BS=2000)
- ▶ Overall time = compute + data exchange = 7.6ms

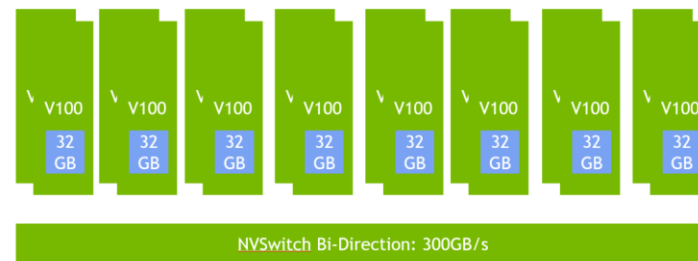


Bottle Neck is Network

CTR SOLUTION

Single GPU Node

- ▶ Single Node
 - ▶ Within GPU server: model exchange is >83x faster (0.067ms)
 - ▶ Compute Time: **6ms** (batchsize= 2×10^5)
 - ▶ Total Time = 6ms (1.26x 100 CPU Nodes)

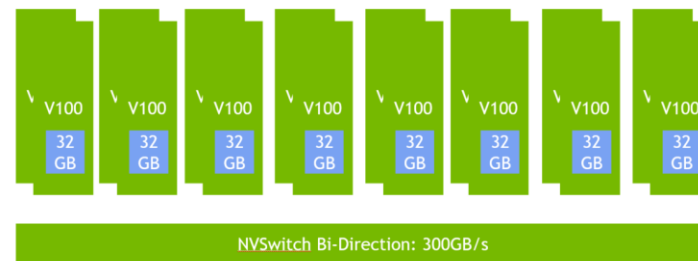


CTR SOLUTION

Single GPU Node

- ▶ Single Node

- ▶ Within GPU server: model exchange is >83x faster (0.067ms)
- ▶ Compute Time: **6ms** (batchsize= 2×10^5)
- ▶ Total Time = 6ms (1.26x 100 CPU Nodes)



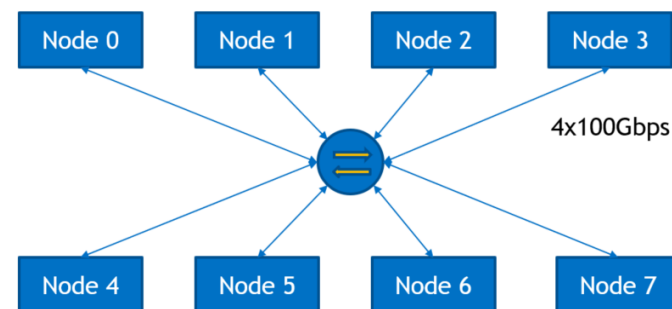
Bottle Neck is Compute

CTR SOLUTION

Multi GPU Nodes

► Multi Node

- Within GPU server: model exchange is 27.8x faster than CPU
- Compute Time: $6\text{ms}/\text{\#Node}$
(batchsize= $2 \times 10^5 / \text{\#Node}$)
- Total Time = $6\text{ms}/\text{\#Node} + 0.2\text{ms}$
(linear scale if Nodes < 10)



CHALLENGES FOR GPU SOLUTION

Streaming Training: Dynamic Hashtable Insertion

Very big hashtable (GBs~TBs)

Large data I/O for data reading

Very shallow networks (3~20 layers)

Not a typical DNN training can be handled by current frameworks like pytorch
TensorFlow

CHALLENGES FOR GPU SOLUTION

Challenges:

- ▶ Streaming Training: Dynamic Hashtable Insertion
- ▶ Very big hashtable (GBs~TBs)
- ▶ Large data I/O for data reading
- ▶ Very shallow networks (3~20 layers)

HugeCTR:

- ▶ Flexible GPU Hashtable
- ▶ Multi-Node training
- ▶ Efficient Three Stage Pipeline

HUGECTR INTRODUCTION

WHAT IS HUGECTR

HugeCTR is a high efficiency GPU framework designed for Click-Through-Rate (CTR) estimating training.

Key Features in 2.0:

- GPU Hashtable and dynamic insertion
- Multi-node training and enabling very large embedding
- Mixed precision training

HOW HUGECTR HELP

1. Prototype: Showing performance and possibility on GPUs. (v1.0)
2. Reference Design: Developers and NV can work together to modify HugeCTR according to the specific requirements (v2.0 current stage)
3. Framework: Developers can train their model easily on HugeCTR (v3.0)

NETWORK SUPPORTED

Embedding + MLP

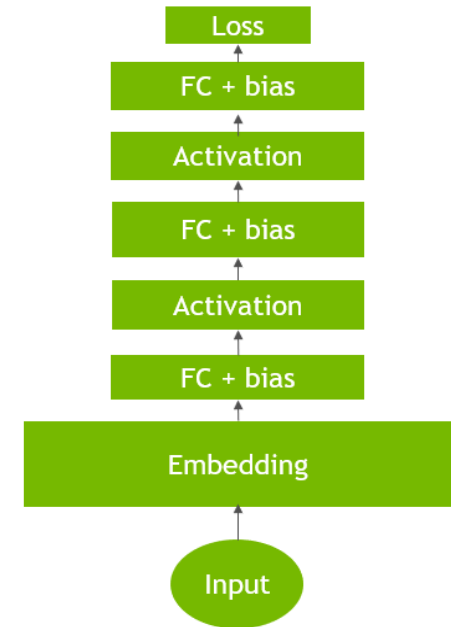
Multi slot embedding: Sum / Mean

Layers: Concat / Fully Connected / Relu / BatchNorm / elu

Optimizer: Adam/ Momentum SGD/ Nesterov

Loss: CrossEntropy/ BinaryCrossEntropy

* Supporting multiple labels and each label will have a unique weight



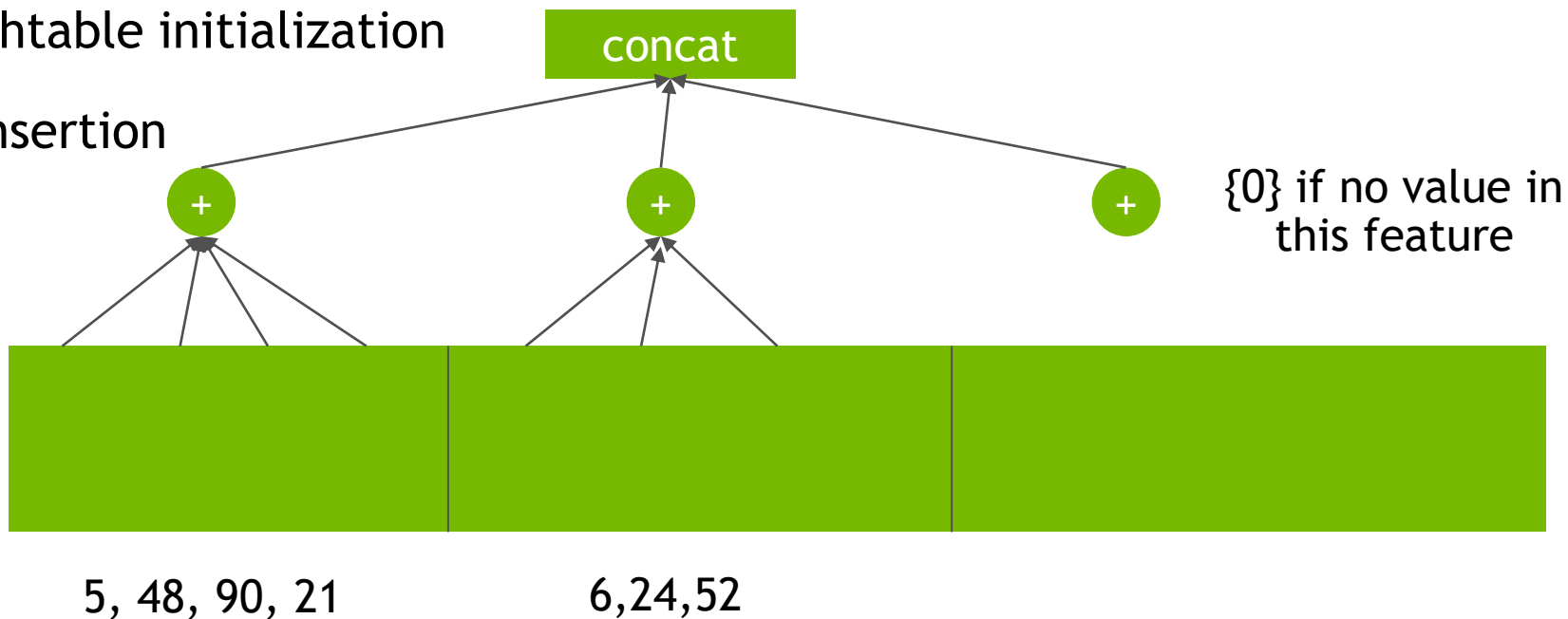
NETWORK SUPPORTED

Sparse Model

Supported reduce '+': sum / mean

Empty Hashtable initialization

Dynamic insertion



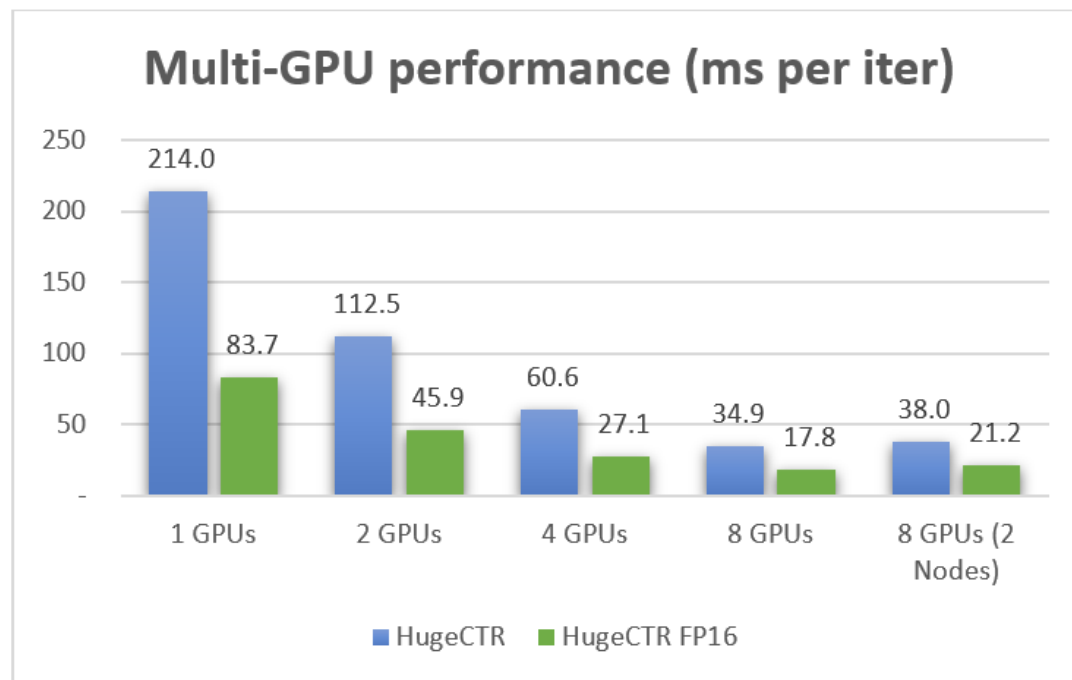
PERFORMANCE

Good Scalability

NCCL 2.0

Three stages pipeline:

- reading from file
- host to device data transaction (inter / intra nodes)
- GPU training

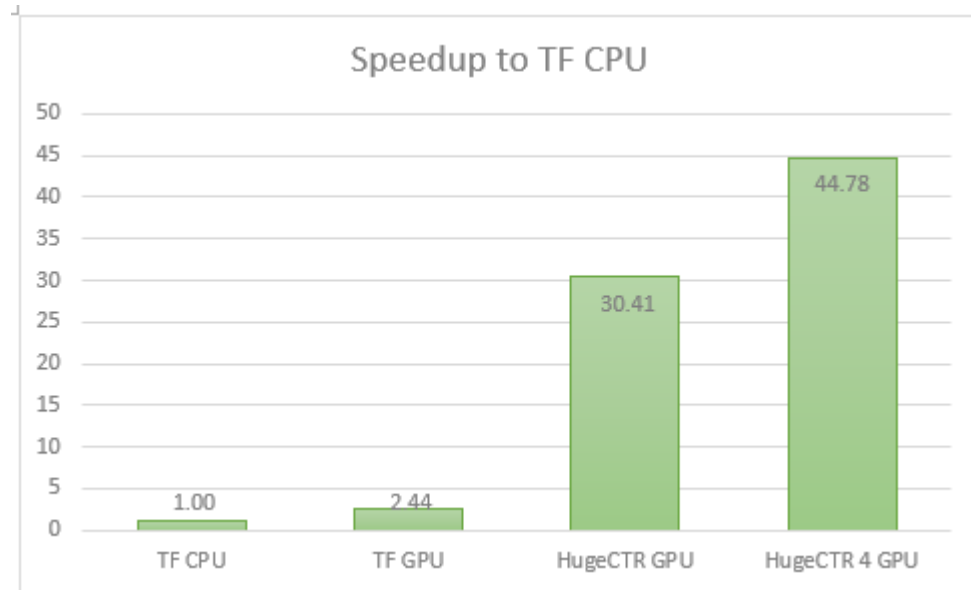
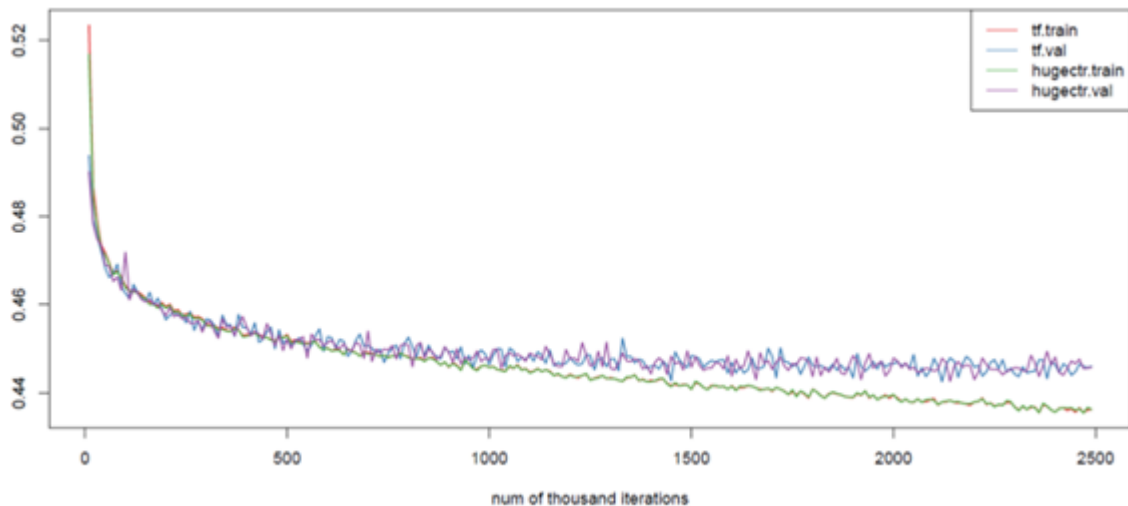


*MLP Layers: 12 / MLP Output: 1024 / Embedding Vector: 64 / Table Number: 1

PERFORMANCE

TensorFlow

44x Speedup to CPU TF and same loss curve



PERFORMANCE

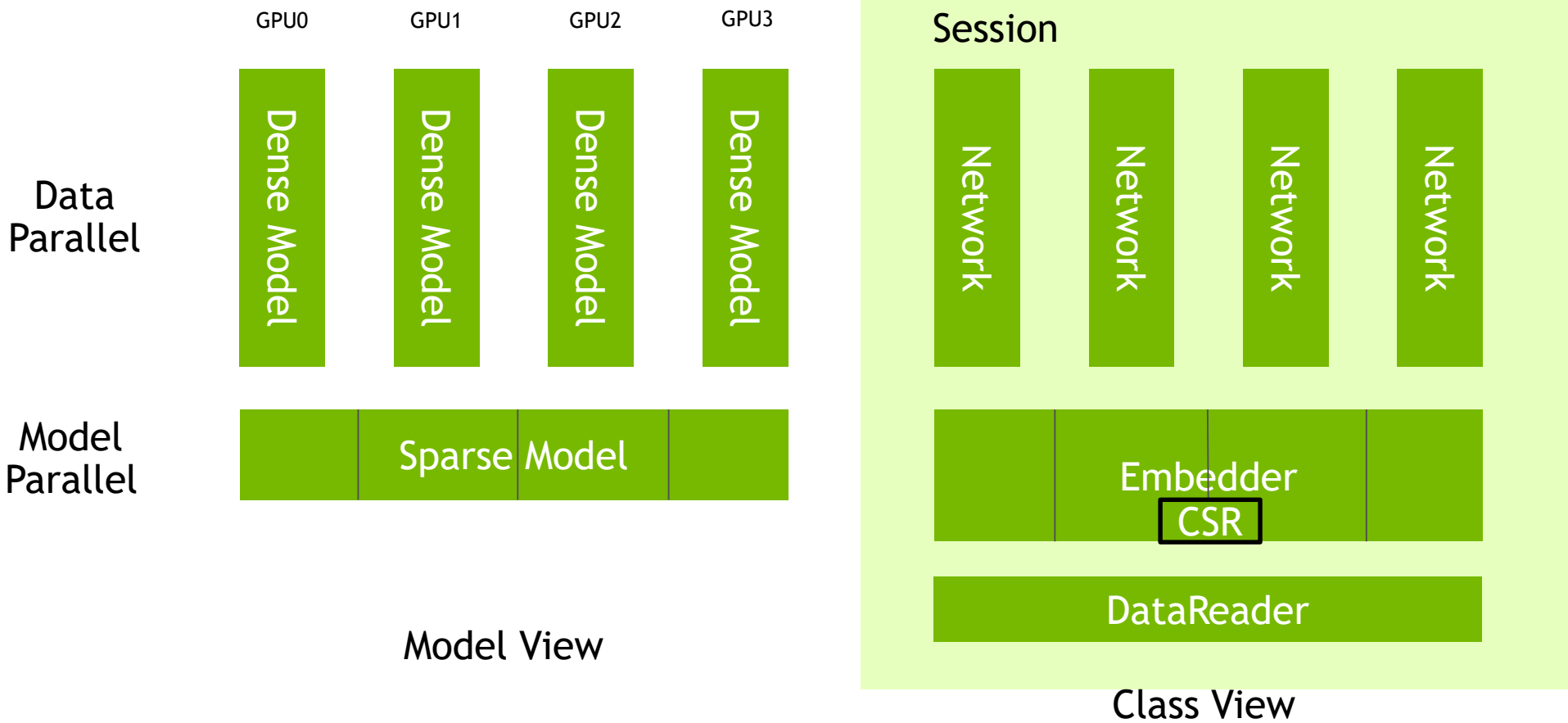
Pytorch DLRM

HugeCTR	slot_num	embedding_vec	num_layers	output of MLP
	64	64	4	512

GPUs	Batchsize	HugeCTR Time (s per 200iters)	DLRM (200iters)	Speedup
1	40960	13.5	17.7	131%
2	40960	10.3	19.4	188%
4	40960	6.3	17.3	275%
8	40960	4.3	33.8	786%
1	4096	1.6	4.5	281%
2	4096	1.34	6.5	485%
4	4096	0.9	8.4	933%
8	4096	0.75	13.7	1827%

Embedding Vector: 64 / Layers: 4 / MLP Output: 512 / Table number: 64

SYSTEM



HOW TO USE

A Simplified Framework For Ranking or Retrieval

Weight initialization: generate a file with initialized weight according to the name in config file

```
$ huge_ctr --init config.json
```

Training:

```
$ huge_ctr --train config.json
```

All the network, solver and dataset is configured under config.json

HOW TO USE

Config.json

Configuration file is in json format, and has four parts:

Solver

Optimizer

Data

Network

```
{
  "solver": {
    "lr_policy": "fixed",
    "display": 200,
    "max_iter": 50000,
    "gpu": [[0],[0]],
    "batchsize": 40960,
    "snapshot": 10000,
    "snapshot_prefix": "./",
    "eval_interval": 1000,
    "eval_batches": 100,
    "model_file": "./criteo.model"
  },
```

```
  "optimizer": {
    "type": "Adam",
    "adam_hparam": {
      "alpha": 0.005,
      "beta1": 0.9,
      "beta2": 0.999,
      "epsilon": 0.000001
    }
  },
```

```
  "data": {
    "source": "./file_list.txt",
    "eval_source": "./file_list_test.txt",
    "max_feature_num_per_sample": 100,
    "label_dim": 1,
    "slot_num": 1
  },
```

```
  "layers": [
    {
      "name": "sparse_embedding1",
      "type": "SparseEmbeddingHash",
      "top": "sparse_embedding1",
      "sparse_embedding_hparam": {
        "vocabulary_size": 1603616,
        "load_factor": 0.75,
        "slot_num": 10,
        "embedding_vec_size": 64,
        "combiner": 0
      }
    },
    {
      "name": "concat1",
      "type": "Concat",
      "bottom": "sparse_embedding1",
      "top": "concat1"
    },
    {
      "name": "fcl",
      "type": "InnerProduct",
      "bottom": "concat1",
      "top": "fcl",
      "fc_param": {
        "num_output": 200
      }
    },
    {
      "name": "relu1",
      "type": "ReLU",
      "bottom": "fcl",
      "top": "relu1"
    }
  ],
```

HOW TO USE

Dataset

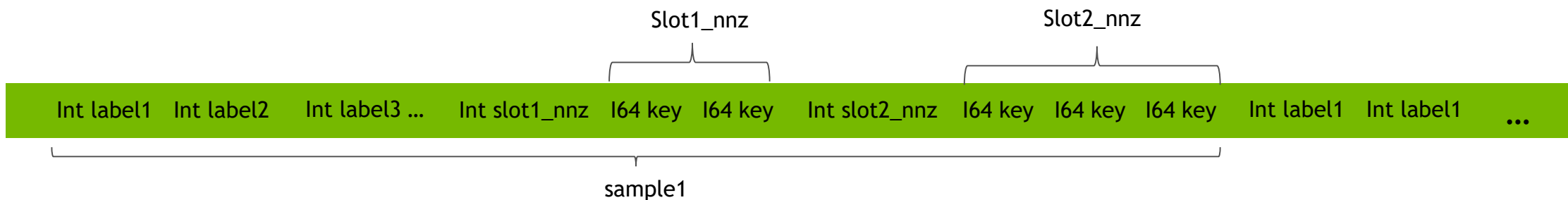
Dataset contains two kinds of files:

1. File list: includes the number of files and file name list with text format.
A file name could be either of a relative address or absolute address. The file names are separated with '\n'
2. Data files: includes a bunch of files with binary format.

HOW TO USE

Data File

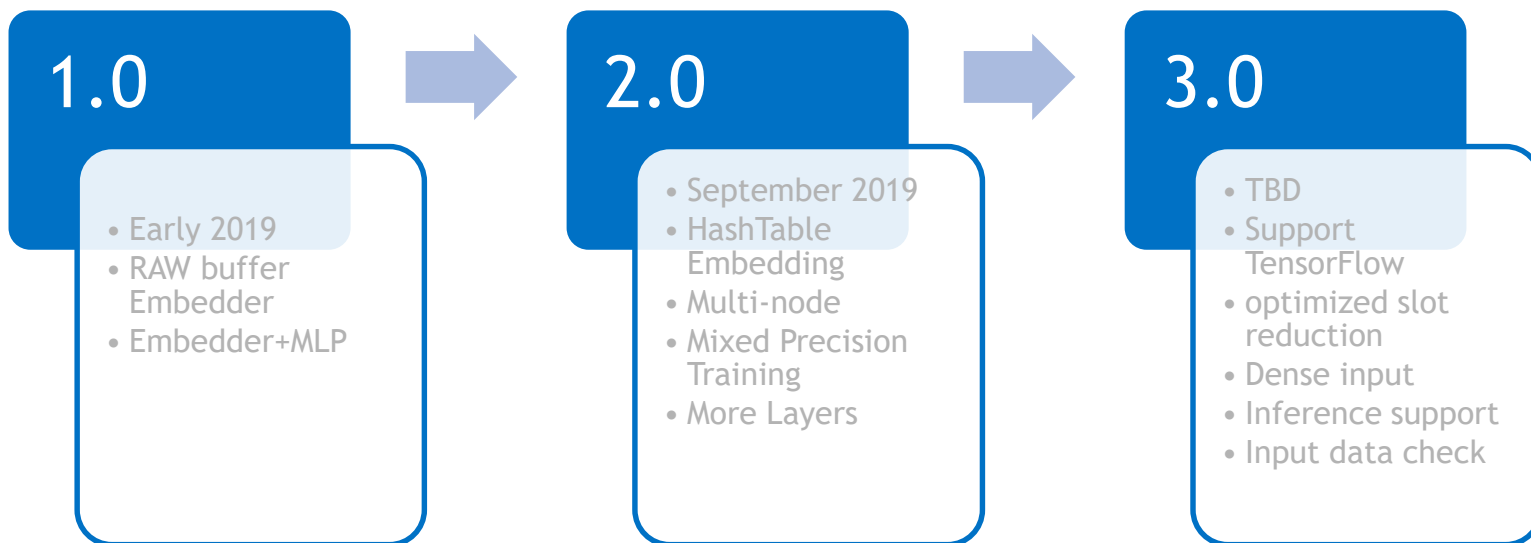
Training Set Format (RAW data with header):



Header:

```
typedef struct DataSetHeader_{  
    long long number_of_records; //the number of samples in this data file  
    long long label_dim; //dimension of label  
    long long slot_num; //the number of slots in each sample  
    long long reserved; //reserved for future use  
} DataSetHeader;
```

ROADMAP



RESOURCES

源码:

<https://github.com/NVIDIA/HugeCTR>

公众号文章:

<https://mp.weixin.qq.com/s/Oieuhvt2vzFEfKklTHiuOg>

KEY CONTRIBUTORS



Fan Yu
Hashtable



Yong Wang
Algorithm
Advisor



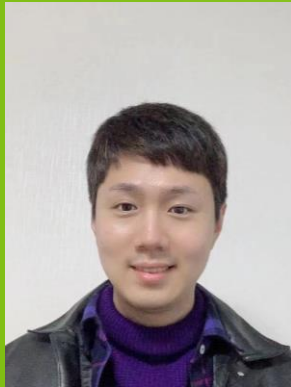
Ryan Jeng
Competitive
Study



Joey Wang
Project
Management



Xiaoying Jia
Mixed
Precision



Minseok
Lee
Multi-Node



David Wu
Embedding



沟通

与来自 NVIDIA 和其他业界领先组织的技术专家互动。



学习

通过百余场讲座、动手实验和研究海报获取宝贵见解和实践培训。



发现

了解 GPU 技术如何为深度学习等重要领域带来重大突破, 描绘最新 AI 世界观。



创新

共同探索改变世界的颠覆性创新, 定义未来。

立即注册, 扫码立享 75 折邀请优惠购票
或使用我的优惠邀请码: NVZEHUANW
前往 www.nvidia.cn/gtc/ 完成报名



CUDA PYTHON

探讨如何使用 Numba (即时，专用类型的 Python 函数编译器) 在 NVIDIA 大规模并行运算的 GPU 上加速 Python 应用程序。

您将学习如何：

- 使用 Numba 从 NumPy ufuncs 编译 CUDA 内核
- 使用 Numba 创建和启动自定义 CUDA 内核
- 应用关键的 GPU 内存管理技术

完成本课程后，您将能够使用 Numba 编译并启动 CUDA 内核，以加速 NVIDIA GPU 上的 Python 应用程序。



RYAN JENG
NVIDIA 高級工程師



扫码注册, 经典课程+全新主题,
AI 实践经验升级



zehuanw@nvidia.com

