

# BigDL: Framework para Deep Learning Distribuido

Integrantes:

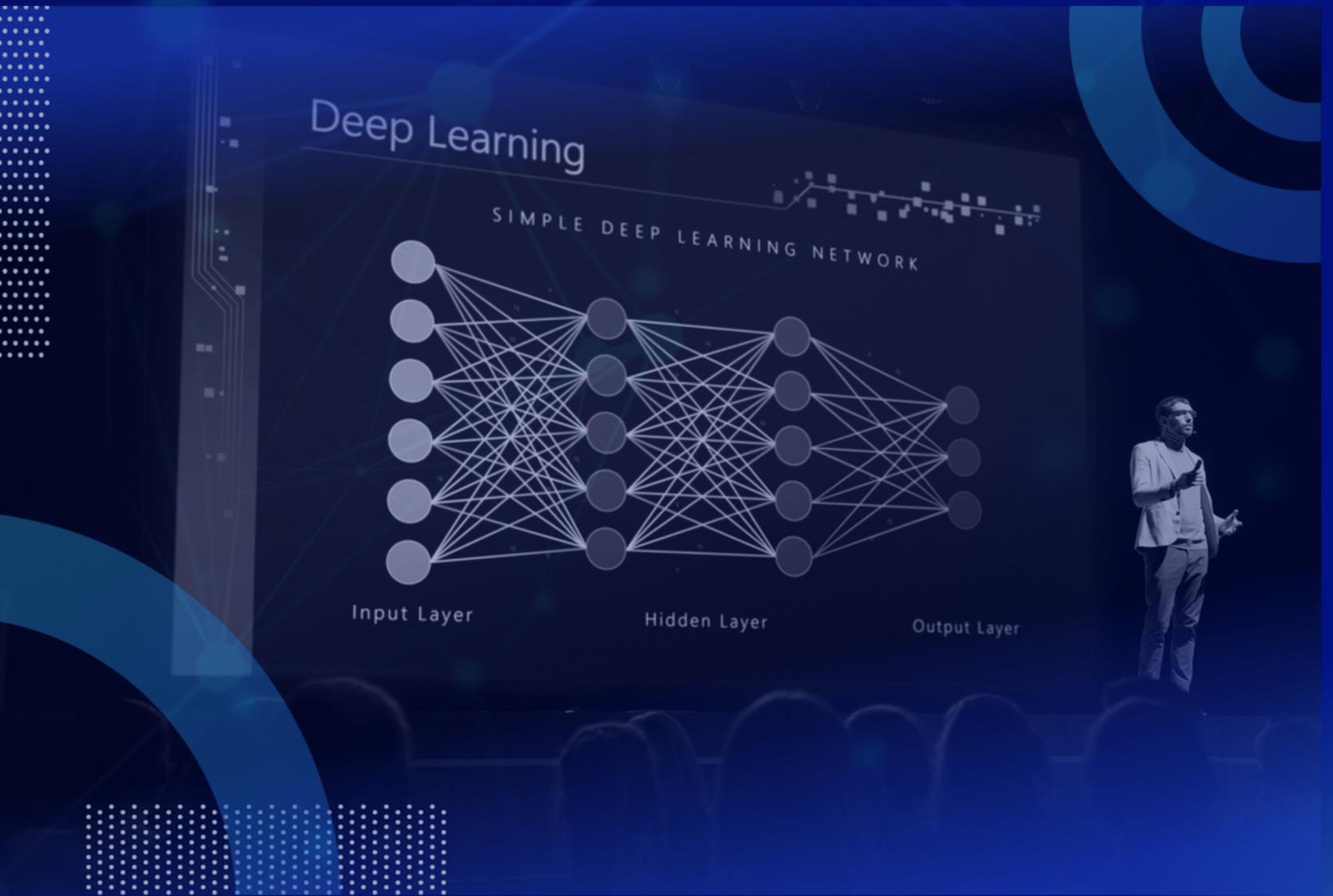
- Miguel Soto
- Nicolás Berenguela

# Contexto

- BigData dominaba el mercado los 2010's, mucha informacion pero no existian las herramientas para trabajarla.
  - Se crea Spark y Hadoop
  - Orientado a las CPU's
  - Estallido del Deep Learning en 2012 gracias a los avances en GPU's.
  - Se crean herramientas como TensorFlow y PyTorch
  - Orientado a las GPU's
- 
- Las diferencias de principios entre ambos hacen que el proceso de unificar una pipeline de datos fuera muy costoso
  - Propenso a errores, generaba el famoso "Impedence mismatch"

# Motivación

- En flujos tradicionales de trabajo en Big Data nos encontramos con el **Problema de Impedancia**.
  - Spark → Mover Datos → Entrenar Clusters de GPU's
- Los conectores de estos procesos sufren de ejecuciones imcompatibles
  - Spark: Tareas cortas, independientes y *stateless*
  - Frameworks de DL: Tareas largas, comunicativas y *stateful*
- Como consecuencia 1 fallo de un worker de Spark puede bloquear indefinidamente todo el entrenamiento de DL.



# BigDL

- BigDL es un framework para procesamiento de datos
- Creado por ingenieros de Intel, Alibaba y tencent
- Es un “Puente Nativo”
- Traer el algoritmo a los datos, no los datos al algoritmo



# ○ Funcionamiento

- ¿Como solucionamos nuestro problema?
  - Reimplementamos las operaciones de entrenamiento distribuido solamente utilizando logica nativa de Spark
- BigDL comienza distribuyendo los datos de entrenamiento (Sample RDD) y copias identicas del modelo de red neuronal (Model RDD) en todos los nodos disponibles (Fisicamente todo esto en el mismo nodo) → Co-Localización.
- Cada Nodo realiza un calculo “forward” y un “backward” para obtener un gradiente local (Proponer un cambio)
- Tomamos todas las “opiniones” de los nodos, se combinan y se actualiza el modelo global
- Logramos la sincronización de parámetros (AllReduce) tipica de DL y lo transformamos en operaciones “Shuffle” y “Broadcast”





# Aplicaciones

BigDL en Producción

JD.com:

- Pipeline de extracción de características de imágenes.
- 3.83x más rápido que su solución anterior con Caffe/GPU y mucho más simple de desarrollar.

Cray:

- Aplicación de pronóstico de precipitaciones con modelos Seq2Seq.
- Unificaron el pipeline de datos y entrenamiento,
- mejorando la productividad del desarrollador.

GigaSpaces:

- Clasificación de voz en tiempo real.
- Integración nativa con
- Kafka y Spark Streaming para inferencia distribuida



# ○ Ventajas

Ventajas de la Arquitectura Unificada

1) Elimina el Desajuste de Impedancia:

- \* Un solo modelo de ejecución para todo el pipeline, desde el ETL hasta la inferencia.

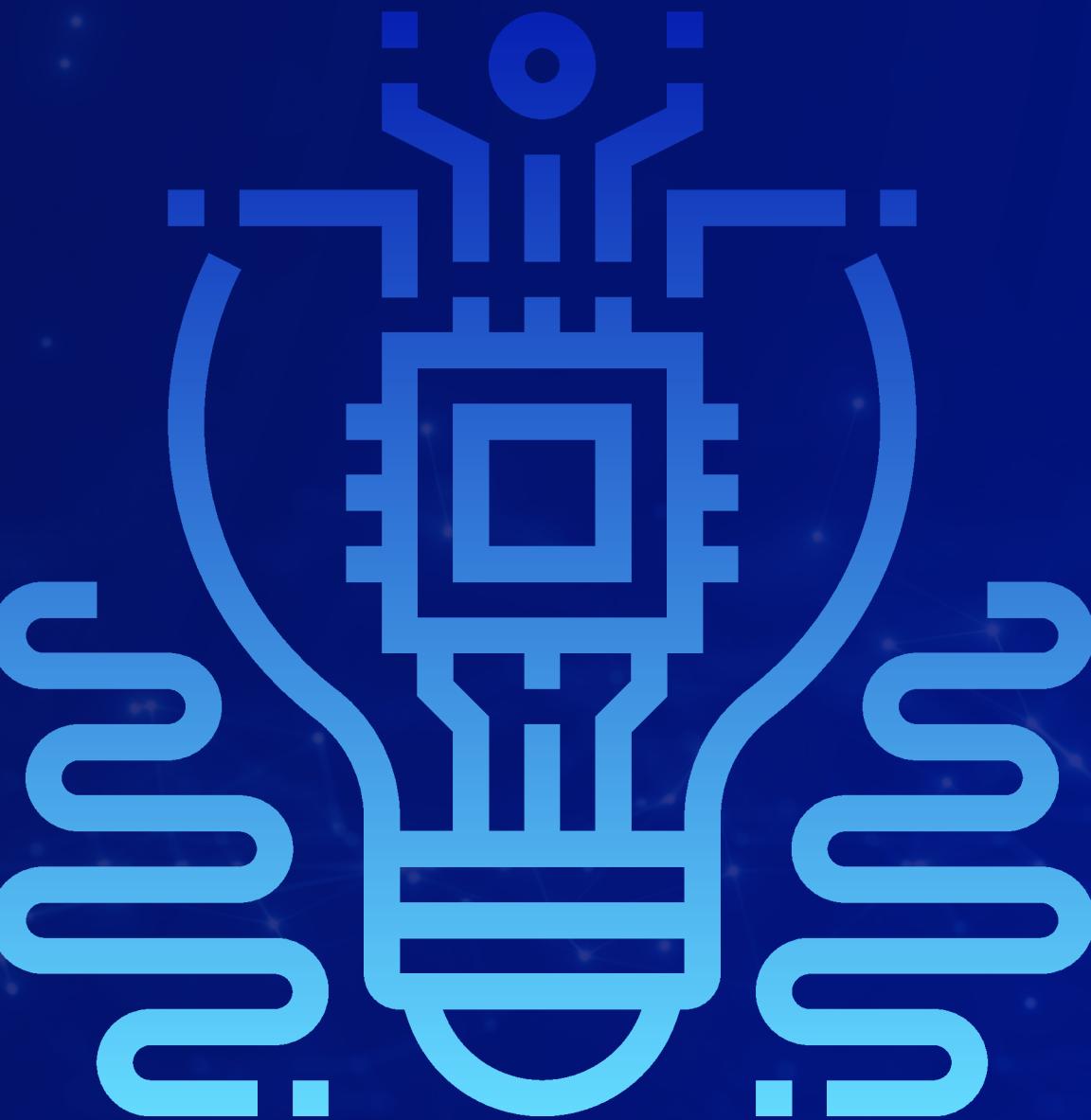
2) Tolerancia a Fallos de Grano Fino:

- \* Hereda el modelo de Spark: si una tarea falla, solo esa tarea se re-ejecuta, no todo el trabajo.

3) Simplicidad Operacional:

- \* Un solo clúster y un solo framework que gestionar.

Permite construir fácilmente pipelines de punta a punta.



# Desventajas

Muy nuevo y no tanto soporte

- El primer paper al respecto es del 2019 y el ultimo commit en el GitHub de Intel es de hace mas de un año. En comparacion, PyTorch tuvo un commit hace 20 minutos de escribir esta presentacion.

No tiene soporte para CUDA

- La aceleracion por hardware es critica para poder procesar volumenes grandes de datos de manera eficiente.

Tiene licencia Apache 2.0

- Por ende su uso es mas restrictivo en comparacion a otros frameworks mas permisivos con licencias MIT, por ejemplo.



The Open Language of AI

# Rendimiento

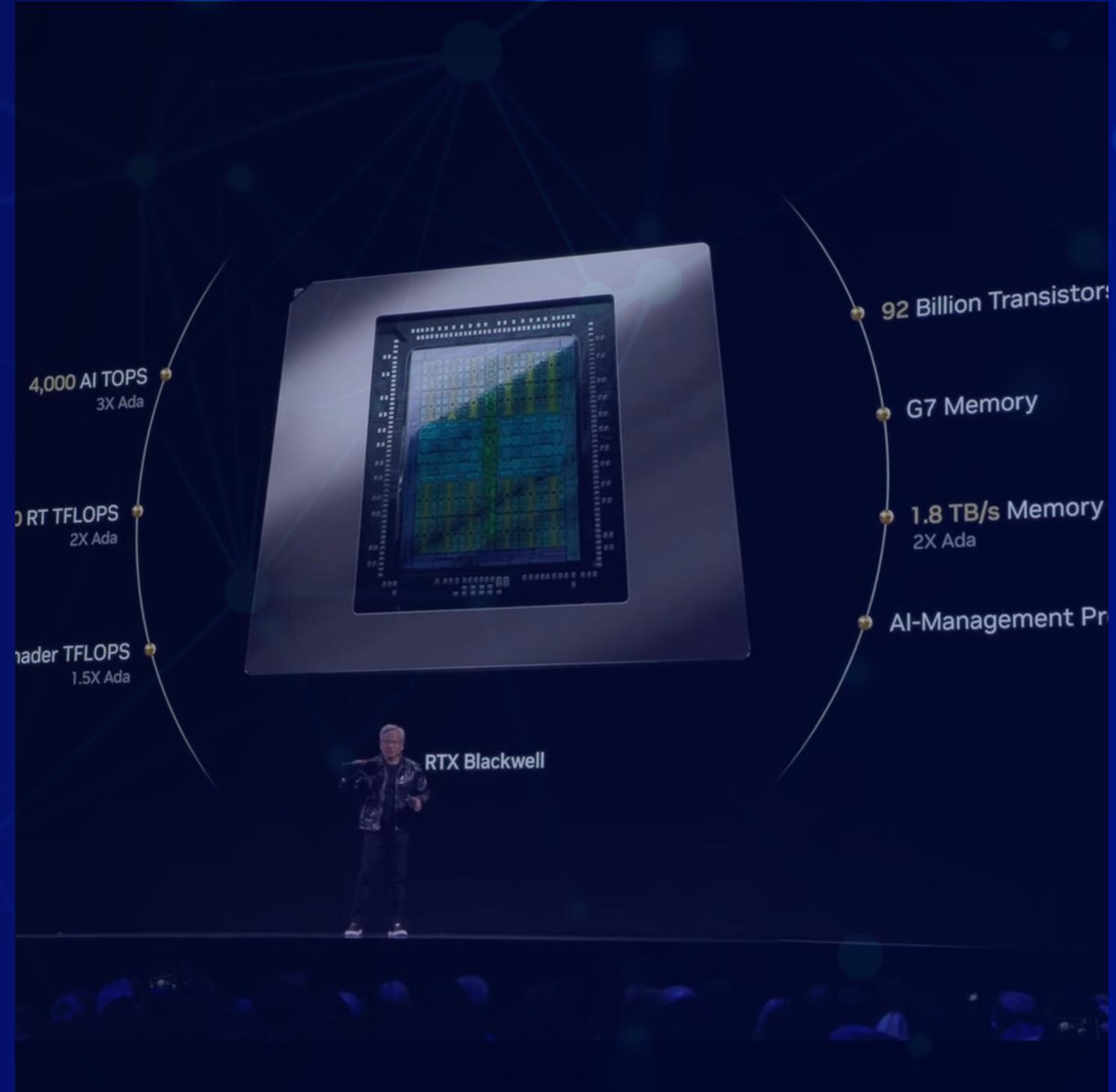
¿Es competitivo? Sí.

Rendimiento Computacional (vs. GPU):

- En un modelo de recomendación NCF, BigDL en un servidor Xeon fue 1.6x más rápido que la implementación de referencia en PyTorch sobre una GPU P100.

Escalabilidad del Entrenamiento Distribuido:

- El entrenamiento de ImageNet (Inception-v1) escala casi linealmente hasta 96 nodos y sigue escalando bien hasta 256 nodos.



# Estudios a partir de BigDL

- ***An Analysis of the Elevator Failure Prediction System with BigDL Time Series Forecasting Models:***

Estudio con enfoque en la industria de los elevadores comerciales, en donde se utiliza el “Forecaster” Seq2Seq para generar un modelo predictivo para accidentes relacionados.



- ***Executing Spark BigDL for Leukemia Detection from Microscopic Images using Transfer Learning:*** La enfermedad de Leucemia Aguda es un problema relacionado a la medula osea. A través de procesamiento de imágenes digitales este estudio se pretende explorar métodos para detectar esta enfermedad en etapas tempranas.



# Conclusiones

- BigDL es una alternativa viable y poderosa para el Deep Learning en Big Data.
- Su principal innovación es un entrenamiento distribuido eficiente sobre el modelo funcional de spark, algo que se creía ineficiente.
- Unificación de los pipelines de datos y DL, resolviendo el *impedance mismatch*.
- Su valor está comprobado por la industria con casos de uso en producción a gran escala