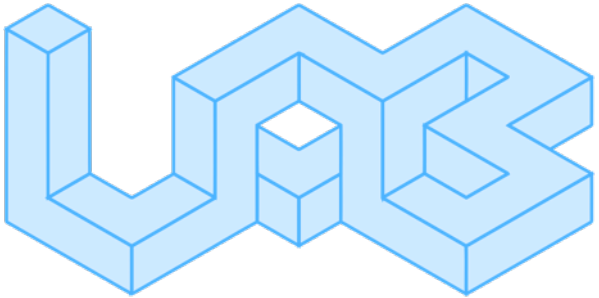
오픈테크넷 2018

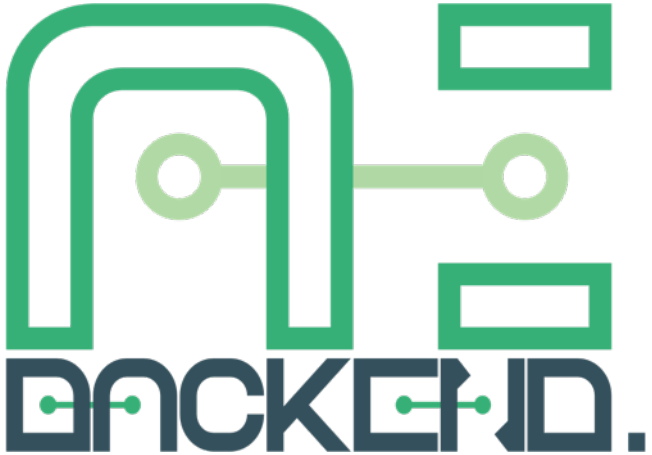
초 대규모 연산 자원 기반의 AI 자동화 시대 대응 전략

신정규 / 래블업 주식회사



Lablup Inc:

Make AI Accessible



기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 오픈소스 플랫폼

Backend.AI 포지셔닝

개발자를 위한
사용자 인터페이스

GPU를 활용하는
다양한 도구

관리 및 오케스트레이션
미들웨어 계층

가상화 솔루션
및 클라우드 서비스

5대 핵심목표

쉽게 **함께**

빠르게 **어디서나**

값싸게

- 보안 격리된 컨테이너 기반 가상화로 서버 수준 고밀도 자원 공유
- 오토스케일링 및 컨테이너 버전 관리
- 컨테이너와 GPU 연결 기능 제공
- 사전 정의된 목적 특화 컨테이너 제공
- 사용자별 자원 사용량 추적
- Jupyter, VSCode, Atom, CLI/IDE 등 다양한 사용자 개발환경 플러그인 지원
- 완전관리형 클라우드 / 설치형 오픈소스

왜?

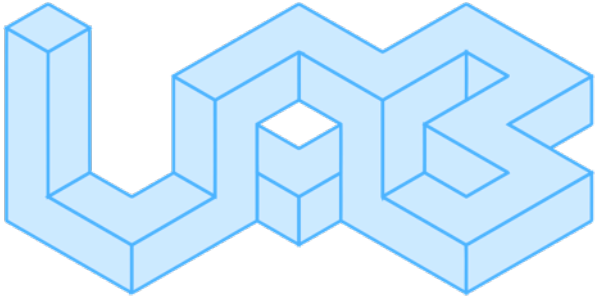
이런 플랫폼을 만들고 있는가!

오늘의 이야기

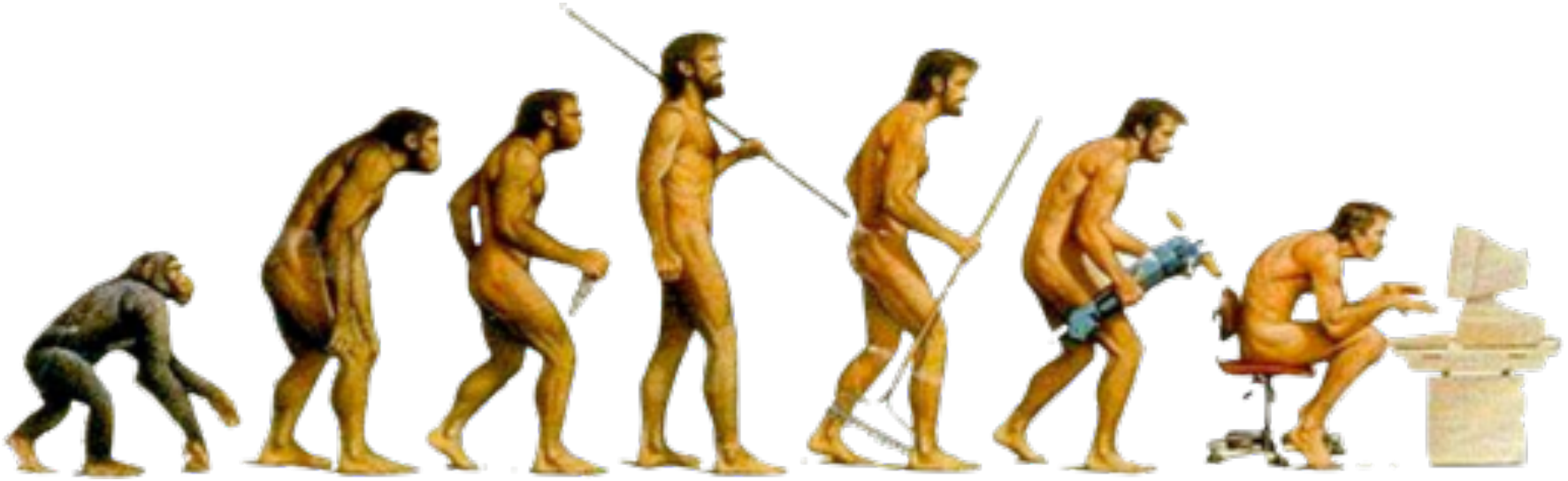
- 왜?

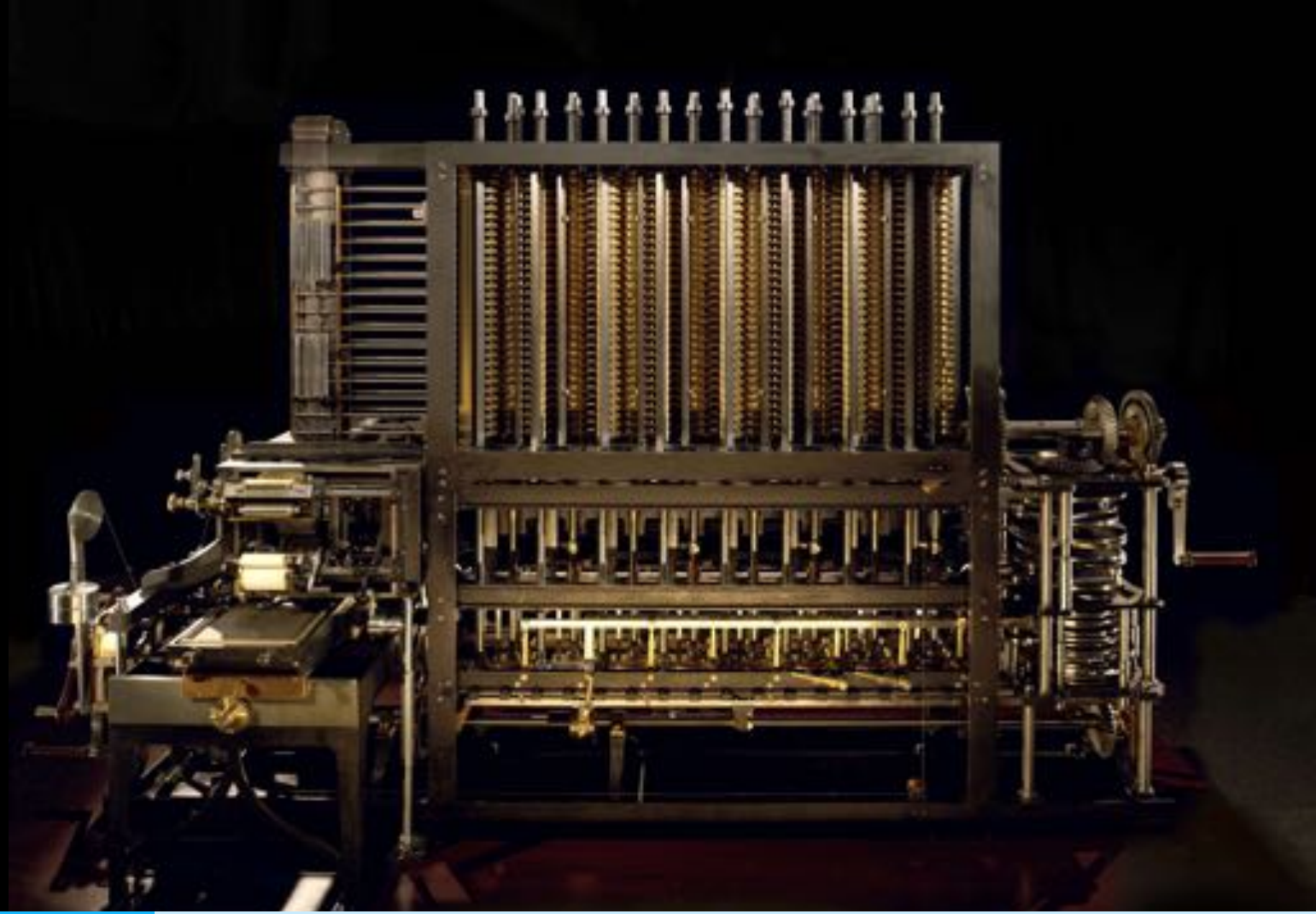
- 놀고 싶은 인간
- 인공지능 구축의 3요소
- GPU 기반의 초대규모 (ultra-large scale) 행렬 연산 환경
- 초대규모 연산 기반의 AI 환경 변화 전망

- 그래서



1. 놀고 싶은 인간





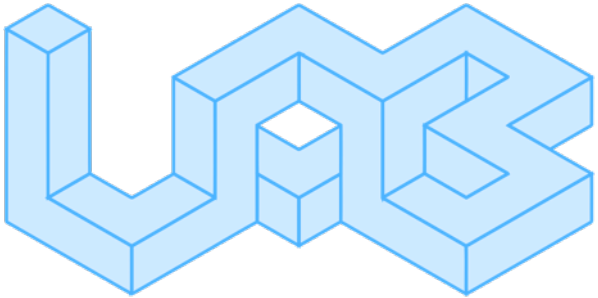
지성 =

계산 능력 + 지능 + 통찰

"앞으로 10년동안,
우리는 거의 모든 것이 디지털화되는 시점에
도달할 것이다."

–Satya Nadella, Microsoft CEO

…이 언급 이후 이미 5년 지났습니다.

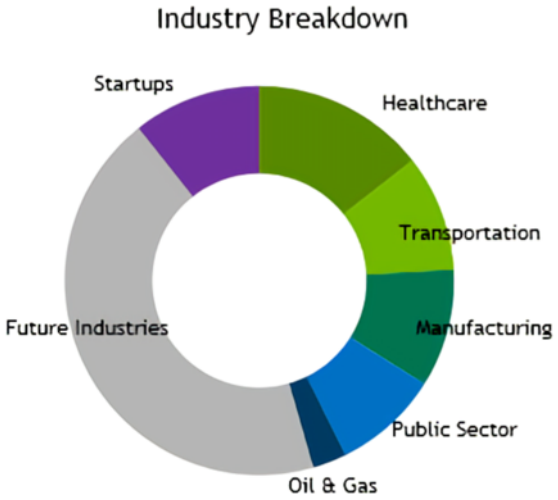


2. 인공지능 구축의 3요소

클라우드

- 개념의 변천
  - 메인 프레임(1980s) - Network Computer (1990s) – 데이터 서버 (2000s)– 클라우드 (2010s)
- 필요한 만큼의 양을 필요할 때 사용
  - 스토리지에서 연산 자원까지
- 시작
  - 대규모 연산 자원 이용의 특징: 이용 패턴이 busty 함
  - 그렇다면 나머지 유휴 시간에는?
- 아마존 웹서비스
  - 자체적인 서비스 백엔드를 외부에 공개하는 서비스로 시작
  - 현재는 아마존의 주요 서비스로 성장

클라우드 컴퓨팅과 산업



US $20B

TAM, 세계 시장 전체 규모*

Alibaba Cloud · aws · Baidu Cloud · Google Cloud · ORACLE · Tencent Cloud · IBM Cloud

KOMATSU · Ford · LOCKHEED MARTIN · BAKER HUGHES · KLA-Tencor

클라우드 & 산업 선도 업체

클라우드 최상위 산업분야

- 헬스케어
- 운송
- AI 도시 & 스마트 팩토리

성장 중인 개발자 생태계

2,000곳 이상의 기업이
GPU 클라우드를 사용중
(전세계)

Reference: TAM US $20B at 2016, Traderscommunity.com “NVIDIA Analyst day gaming A.I. trends growing”

클라우드로의 이전 가속

"2015년에는 20%의 데이터가
클라우드에 존재할 것으로 예상된다"

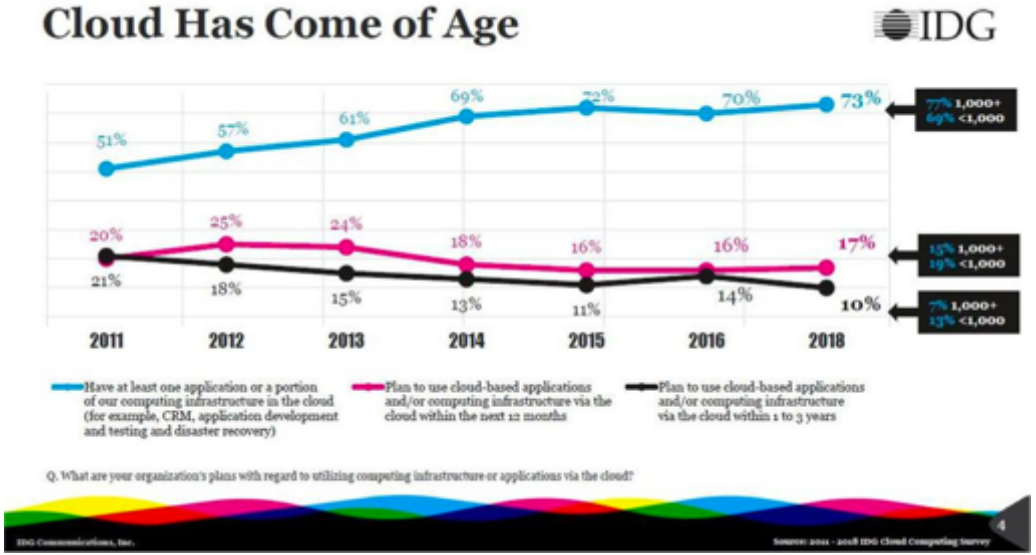
-2011년

"하드웨어 및 소프트웨어의
단위당 비용은
2005년 대비 1/6로 감소…"

-2012년

"현재 약 76%의 서비스가
클라우드에서 운영…"

-2018년



2018 IDG Cloud Computing Study

클라우드 시장 예측

- 급격한 속도로 인프라 구조의 교체 진행중
- 2021년까지의 예측

	2017	2018	2019	2020	2021
Cloud Business Process Services (BPaaS)	42.6	46.4	50.1	54.1	58.4
Cloud Application Infrastructure Services (PaaS)	11.9	15.0	18.6	22.7	27.3
Cloud Application Services (SaaS)	60.2	73.6	87.2	101.9	117.1
Cloud Management and Security Services	8.7	10.5	12.3	14.1	16.1
Cloud System Infrastructure Services (IaaS)	30.0	40.8	52.9	67.4	83.5
Total Market	153.5	186.4	221.1	260.2	302.5

Worldwide Public Cloud Service Revenue Forecast (Billions of U.S. Dollars)

http://www.gartner.com/newsroom/id/3871416

빅데이터

- “큰 데이터?”
  - 데이터의 입력 속도가 데이터의 처리 속도를 웃도는 상황
  - 기존 방법론으로는 실시간 처리가 불가능한 데이터
  - “빅데이터 방법론” 의 등장

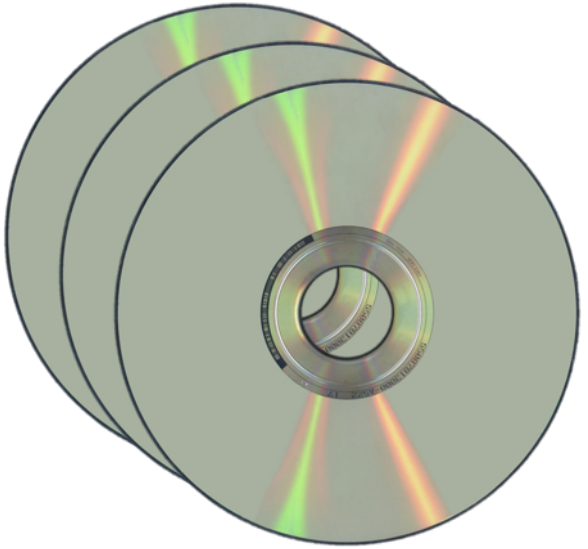
- 클라우드와 빅데이터
  - 빅데이터 방법론의 특징
  - 공간을 더 써서 속도를 올림 (CRUD에서 R,U,D 제거)
  - 분산처리에 적합하게 데이터를 가공하고 병렬화함
  - 클라우드에 적합

- 클라우드 + 빅데이터
  - 표준 빅데이터 처리 방법
  - 서비스 시간 – 유휴 시간에 따라 동작으로 데이터 수집/처리

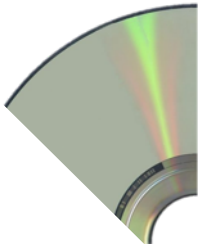
NYSE

1994년

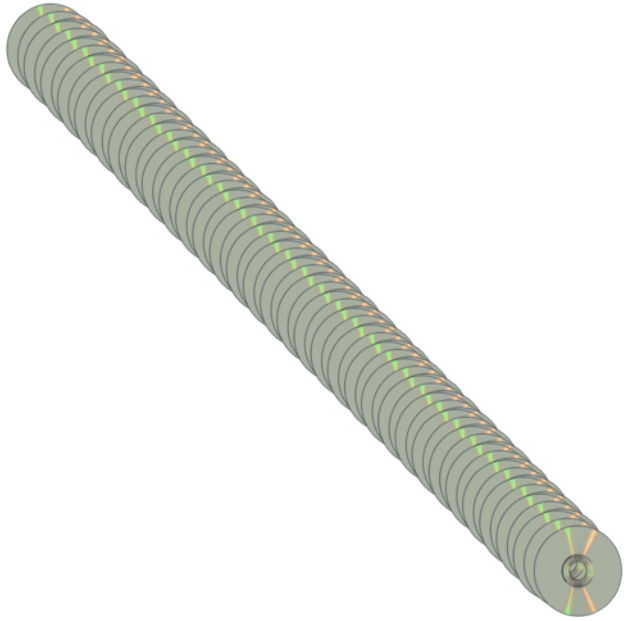
2004년



1994년



2004년

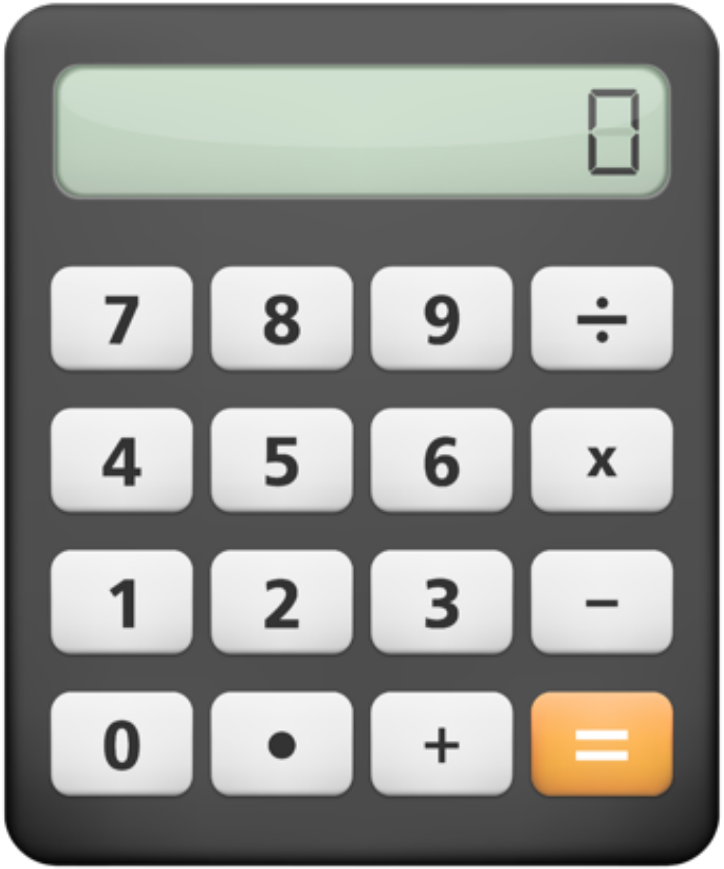


“…위와 같은 이유로 웹의 총 용량은
2 페타바이트에 달한다.”

–2001년

“…2012년의 경우 하루에 생성되는 데이터의 양은
약 7.5 엑사바이트…”

–2012년


0
7
8
9
÷
4
5
6
x
1
2
3
–
0
•
+
=

100만 트랜지스터당 비용

1992	222$
2012	0.06$

=27/1000000

1 GB
저장장치
비용

1985	80,000$
1995	686$
2005	0.75$
2015	0.02$

=25/100000000

=

계산 능력 + 지능 + 통찰

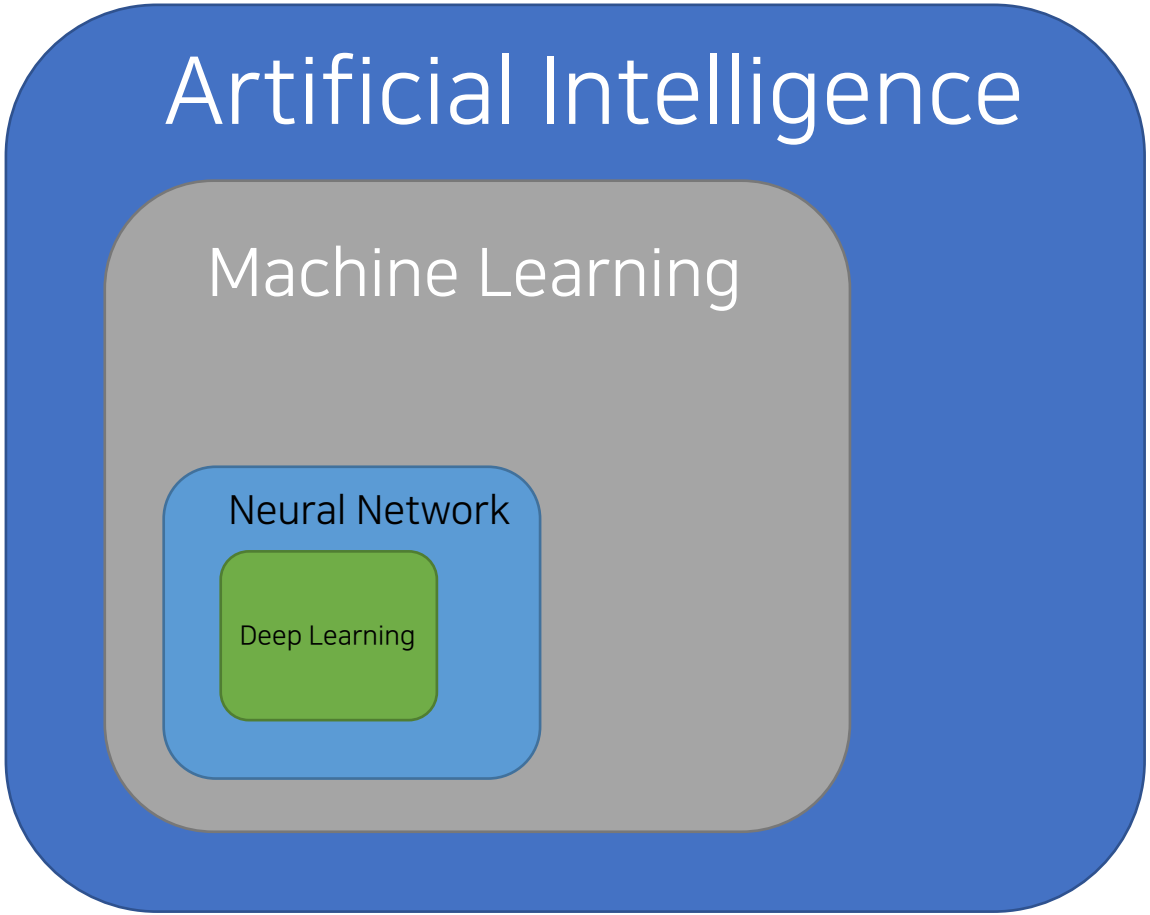
클라우드  빅데이터

딥 러닝: 황금시대의 꽃

- 만약 (일반적으로) 미쳤다고 생각될 실험을 할 정도로 많은 돈이 있다면? 예를 들어
  - 은닉층의 수를 계속 늘리거나
  - 무한정에 가까운 데이터를 부어 넣거나

- 딥러닝의 돌파구
  - Geoffrey Hinton (2005)
  - Andrew Ng (2012)

- Convolution Neural Network
  - Pooling layer + weight
- Recurrent Neural Network
  - Feedforward routine with (long/short) term memory
- Deep disbelief Network
  - Multipartite neural network with generative model
- Reinforcement Learning
  - Deep Q-Network: Using deep learning for reinforcement learning

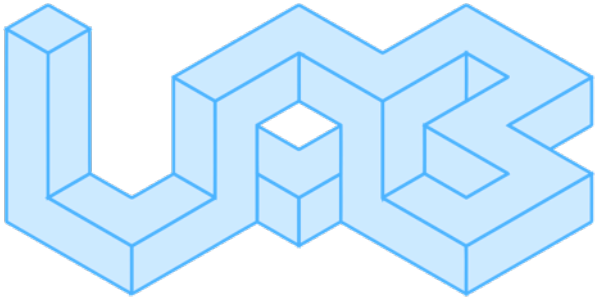

Artificial Intelligence
Machine Learning
Neural Network
Deep Learning

클라우드 인프라의 세대 교체

- 대부분의 인프라스트럭처가 클라우드화 됨
  - 더이상 유휴자원의 재판매가 아님
  - 클라우드 전문 사업

- 머신러닝 시대의 도래
  - 유휴자원 사용 이상의 자원이 요구되는 시점. 왜?
  - 기존에는 필요하지 않았던 리소스가 중요해졌음
  - GPU, special-purpose ASIC (e.g., TPU)

- GPU 클라우드
  - 폭발적인 수요 증가 추세 (2017~)
  - 전성비 측면 / 실사용 측면에서 기술적으로 엄청난 개선 진행 중

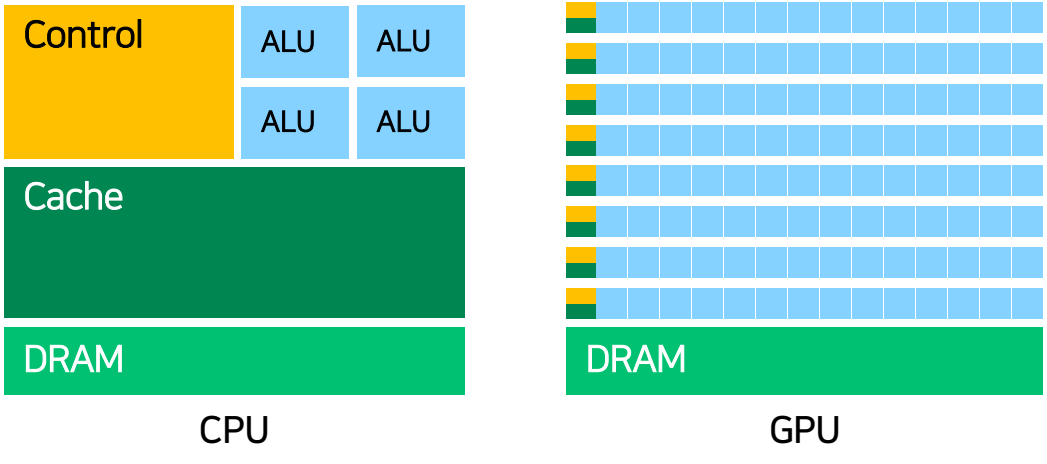


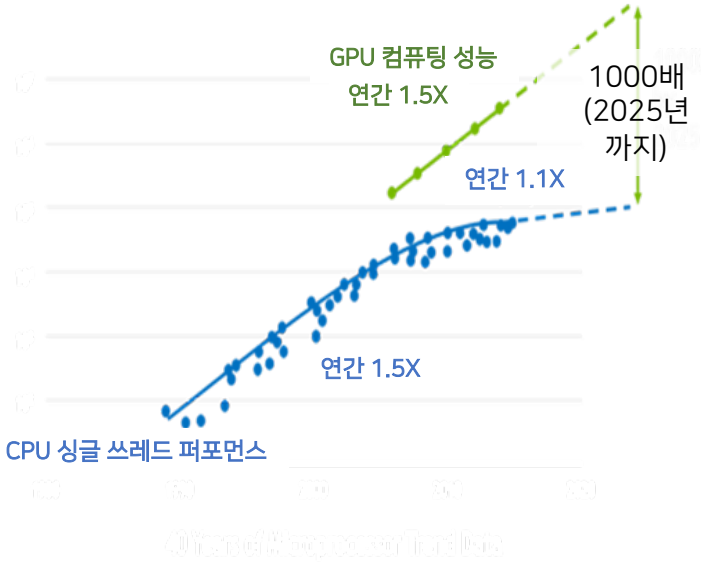
3. GPU 기반의 초대규모 행렬 연산 환경

GPU의 특징

- 대규모 병렬 연산(SPMD†) 방식에 특화된 칩 구조

†SPMD: Single Program Multiple Data





- 2007년 NVIDIA CUDA, 2009년 OpenCL 기술의 등장으로
  일반 C/C++ 개발환경을 그대로 쓰면서 GPU에서 돌아가는 코드를 쉽게 작성할 수 있게 됨
- 이후 수치 연산, 머신 러닝, 딥 러닝 등에서 활용도가 높아지고 있음

GPU 컴퓨팅 분야의 성장

고성능 컴퓨팅

10조원

하이퍼스케일 기반
소비자 시장

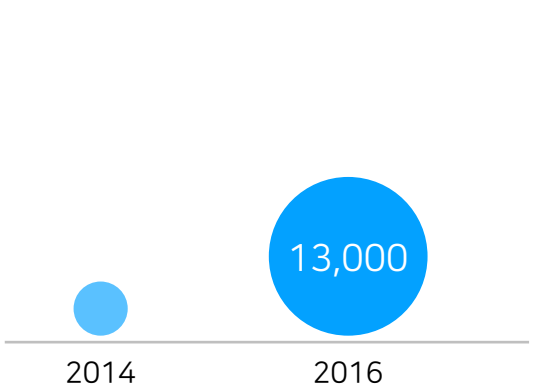
20조원

클라우드 컴퓨팅 및
산업

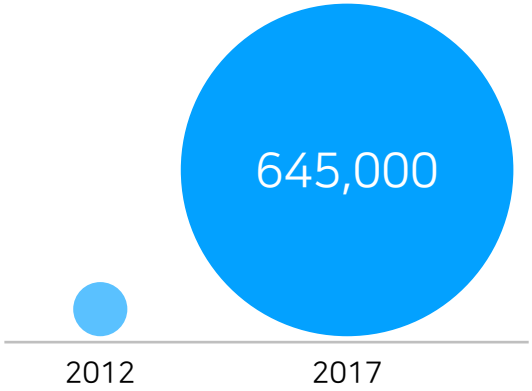
20조원

Reference: NVIDIA ANALYSTIC DAY 2018

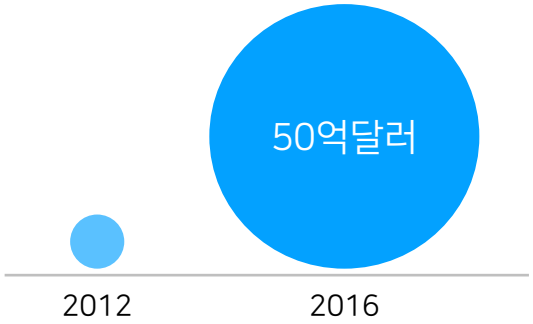
인공지능 시장의 급성장



인공지능 대표 학회
(NIPS, ICML, CVPR, ICLR) 참가자 수
2년사이 2배



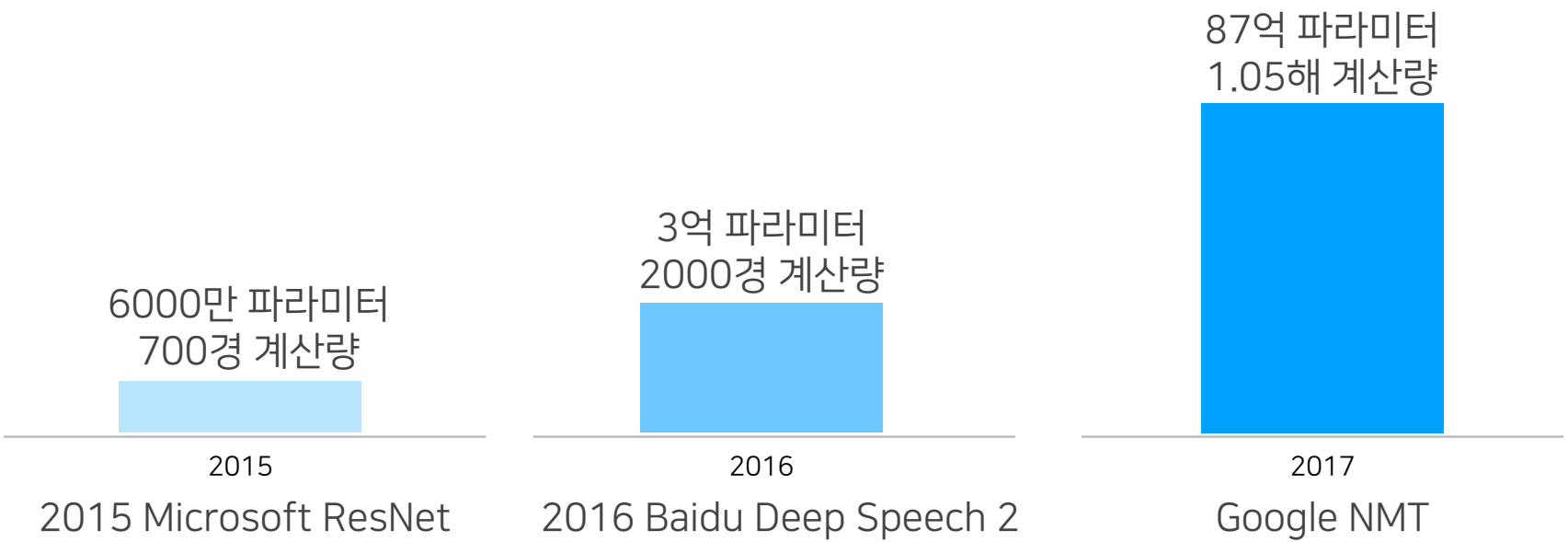
GPU Developers
CUDA 개발자 등록
5년사이 15배



인공지능 스타트업
전체 투자금액
4년사이 9배

Reference: NVIDIA 2017 “A NET COMPUTING ERA”

인공지능 모델의 복잡도 증가



Reference: NVIDIA 2017 “A NET COMPUTING ERA”

계산량 산정: GOPS * bandwidth

클라우드 컴퓨팅: GPU 가속기 데이터센터

GPU를 더 많이 도입할수록 더 절약되는 분야들

Workload	Baseline (CPU-Only)	HPC (Amber,LAMPS)	AI Training (TensorFlow)	AI Inference (Image, Speech)
Speed Up	1x	20x	>100x	60x
Servers	5,000	250	<50	84
Capex	$45M	$11M	$7.5M	$7M
3 Year Opex (Power+Cooling)	$19.5M	$2.5M	$1M	$1.5M
TCO Saving	N/A	79%	86%	86%

Note(s): CPU Baselined to 5000 Servers for each workload | Capex Costs: CPU node with 2x Skylake CPU's ~$9K; GPU node with 4x V100 GPU's ~$45K | Opex Costs: Power & cooling is $180/kW/month | Power: CPU server + n/w = 0.6 KW; GPU server + n/w = 1.6 KW; DGX-1V/HGX-1 Server = 3.2KW | HPC: GPU node with 4xV100 compared to 2xCPU Server | DL Training: DGX-1V compared to a 2xCPU server | DL Inference: HGX-1 based server (8xV100) Compared to 2x CPU Server |numbers rounded to nearest $0.5M

머신러닝과 GPU

- 뉴럴넷의 특징
  - 기본적으로 행렬 연산: GPU 에 적합
  - 성능차: 동일 전성비 기준 10^2 배 이상
- 환경의 변화로 이어짐
  - 머신러닝의 발전 → GPU 수요 증가 → GPU 클라우드 서비스 등장 → 대규모 GPU/ASIC 자원의 등장
- AI Accelerator: ASIC 시장의 활성화
  - Google TPU (2016~), Intel Nervana (2017) and Movidius (2017)
- GPU를 왕창 모아 놓으면 무엇이 가능해지는가?
  - **초대규모 행렬 연산자원** Ultra-large scale calculation resources 의 등장

딥러닝 알고리즘과 자동화

- 딥러닝 알고리즘
  - 최근 가장 각광받는 머신러닝 방법론
  - 연산자원의 증가 혜택

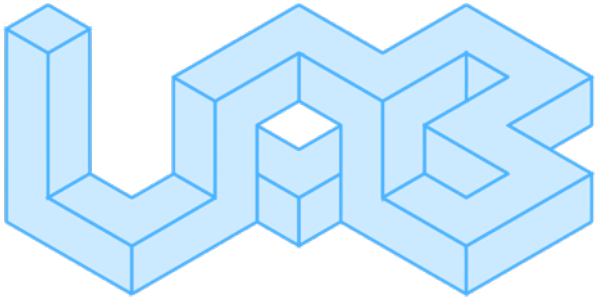
- 딥러닝 알고리즘 자동화
  - 아직 휴리스틱함
  - 자동화하기 어려운 부분 다수 존재
  - 수동 설계, 하이퍼 파라미터 튜닝, 데이터 최적화 등…
  - 도메인 전문가 + 머신러닝 전문가 + 연산 리소스 관리자가 필요

초대규모 연산 자원의 역할

- 가용 GPU가 엄청나게 많으면 어떻게 될까?
  - Trial and error 식의 접근이 가능해짐

- Genetic Algorithm / Genetic Programming
  - 대표적인 오토매틱 프로그래밍 기법
  - 단순 연산 단위의 조합 및 조합 최적화를 통해 문제를 해결하는 알고리즘을 개발하는 머신러닝 방법

- 뉴럴넷과의 결합
  - 뉴럴넷의 연산 단위: 퍼셉트론, 층, 가중치, 뉴런 그룹, 버킷, 숏컷 등
  - 진화 알고리즘 방법론을 임의의 뉴럴넷을 설계할 때 사용할 수 있지 않을까?



4. 초대규모 연산 기반의 AI 전망

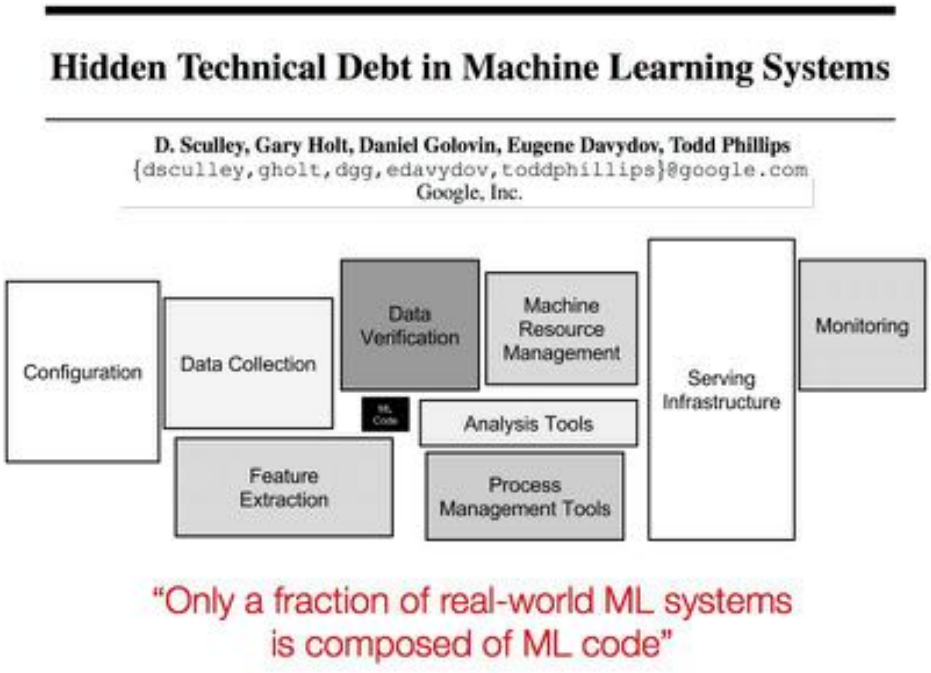
AI 자동화

- AutoML (Google, 2017~)
  - 대규모 연산 자원을 사용한 뉴럴넷 설계
  - 기본 네트워크를 바탕으로 끊임없이 세대를 거치며 최적화를 거치는 방법 채택
  - 비전, 자연어, 번역용 AutoML 공개 (2018. 7, Google CloudML)

- 오픈소스 시도
  - Auto-sklearn : 자동화된 estimator 제공 (2015)
  - AutoKeras : 가우시안 기반의 최적화 (2018)

- 공통점
  - 기존 클라우드로는 감당할 수가 없음 (엄청난 연산량이 필요함)
  - 일반 뉴럴넷 학습 자원의 10^3~ 10^4 배

초대규모 인프라 관리의 자동화



- 비자동화된 부분이 상당히 많음
  - 설정, 데이터 수집, 검증, 특징 추출, 리소스 관리, 인프라스트럭처 관리 등
- 각 부분별 새로운 자동화 도입
  - 클라우드 벤더들의 서비스 제공
  - AWS
  - Google
  - Microsoft

초대규모 GPU 인프라 기반의 AI 모델 최적화

- 개인화
  - 각 분야마다 인공 신경망 템플릿을 미리 작성해 두고, 이를 기반으로 자동 최적화 도구 실행
  - Azure Cognition Toolkit (2017) / Google CloudML (2018)
  - 추가 레이어를 별도로 두는 방식 / 모델 자체의 가중치를 조절하는 방식이 혼재

- 데이터 기반 최적화
  - 공개된 신경망을 기반으로 자신의 용도에 알맞는 신경망이 되도록 데이터를 추가로 집어넣어 학습
  - 실시간 데이터 파이프라인을 통해 동적 평형 상태에 있는 시스템에서 항상 최적화된 결과를 내놓을 수 있음

초대규모 GPU 인프라 기반의 인공신경망 자동 설계 (예측)

- 인공신경망 모델 개발
  - 기본 설계는 여전히 연구진이 진행
  - 최적화는 완전히 AI 자동화에 위탁
  - 현재 타 분야에서 일어나는 일들과 유사함: 건축, 기기 설계 분야

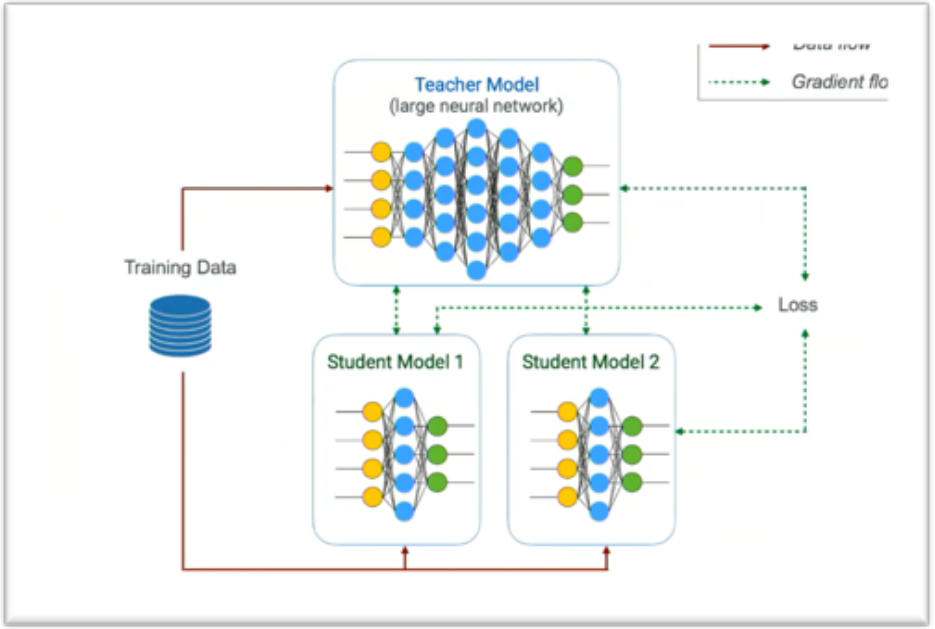
- 장점
  - 미시 최적화 부분을 자동화할 수 있음
  - 지속적으로 정확도를 올려나갈 수 있음

- 단점
  - 최종 결과물의 크기가 필요 이상으로 커질 수 있음
  - 최적화한 모델에서 오버피팅 문제가 발생했을 경우, 모델 구조의 복잡성으로 인하여 디버그하기가 까다로움

서비스용 인공신경망 최적화

- 모델 압축
  - 추론 과정에 쓰일 수 있도록 다양한 방식을 사용하여 신경망 압축
  - 정규화, 반정밀도, bfloat16 도입 등
  - TensorFlow Lite (2016~)

- 학생 모델 교육
  - Google (2018 가을)
  - 거대 신경망 모델을 판단자로 사용하여, 훨씬 단순한 신경망으로 비슷한 성능을 내는 모델 훈련

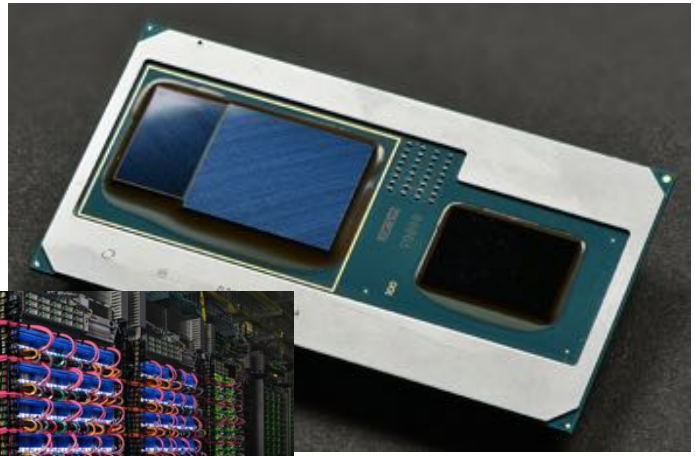


Google I/O 2018, ML Kit for Firebase

하드웨어

- 모바일: On-device 기계학습 (엣지 디바이스)
  - ASIC + GPU + CPU 형태의 AP
  - 모바일을 대상으로 공급 시작 (Apple NeuralEngine (2017), Huawei NPU (2017))
  - 데스크탑 도입 (Intel+AMD i7-8809g (2018))

- 서버: GPU 클라우드용 하드웨어
  - TPU (Tensor Processing Unit), Google (2016~)
  - Intel Nervana + Amazon AWS (2017~)
  - DGX-Family (DGX-1/2, DGX-station), Nvidia (2017~)

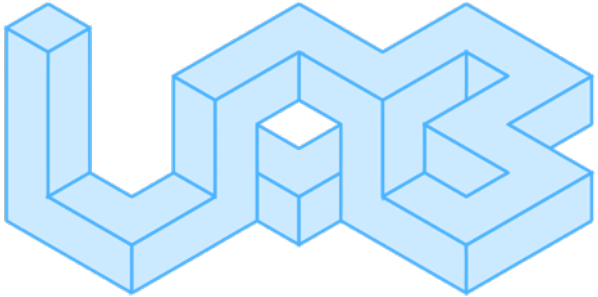


오늘의 이야기 정리

- 왜?

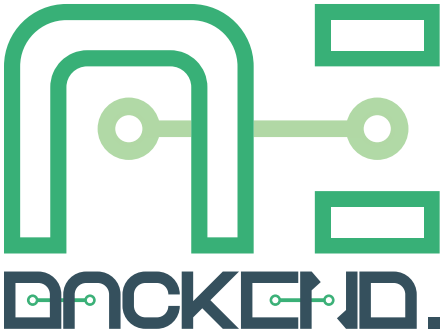
- 놀고 싶은 인간
- 인공지능 구축의 3요소
- GPU 기반의 초대규모 (ultra-large scale) 행렬 연산 환경
- 초대규모 연산 기반의 AI 환경 변화 전망

- **그래서**



5. Backend.AI

: 머신러닝 작업 환경과 리소스를 가장 쉽게 사용하는 방법



Backend.AI 개념도

기계학습 및 과학연산 코드

```
import tensorflow as tf
import matplotlib
v1 = tf.Variable(…, name="v1")
v2 = tf.Variable(…, name="v2")
...
plot(...)
```

개인·조직 사용자

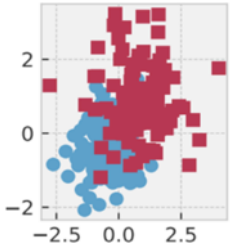
REST/GraphQL API

응용 서비스 및 애플리케이션

클라우드 서비스

웹 서비스

모바일 및 IoT 장비



실행 결과

Backend.AI 플랫폼

오토스케일링

보안격리

버전관리

언어별 SDK

자원할당

모니터링

클라우드 인프라 (IaaS)

온-프레미스 클러스터

Backend.AI 사용 시나리오

🤔 GPU가 없는 랩탑을 들고다니며 딥러닝 AI를 개발하고 싶을 때

자신의 GPU 워크스테이션을 cloud.backend.ai에 등록 또는 클라우드 요금제 구입 → 갖고다닐 개인 랩탑에 Backend.AI Client 설치 → 😎

🤔 사내 고성능 GPU 서버 1대를 여러 연구원이 공유하면서 쓰고 싶을 때

사내 서버에 Backend.AI 설치
(오픈소스 버전 무료 사용 가능) → 각 연구원 PC에 Backend.AI Client 설치 → 😎

🤔 망분리 적용된 사내 GPU 서버를 조직 단위로 유연하게 공유하고 싶을 때

사내 서버에 Backend.AI 설치
(엔터프라이즈 구입 후 오프라인 설치, SSO 연동, 보안 로깅 추가기능 사용) → 각 연구원 PC에 Backend.AI Client 설치 → 😎

Backend.AI 프로젝트 https://www.backend.ai

- https://github.com/lablup/backend.ai
  - 2015년 8월 첫 공개
  - 2016년 11월 v0.1 릴리즈
  - 2017년 10월 v1.0 릴리즈 / 17개의 컴포넌트 저장소로 구성
    - ✓ 개발 매뉴얼 제공
  - 2018년 1월 v1.1 / 2월 v1.2 / 3월 v1.3 / 9월 v1.4 릴리즈
    - ✓ 코드 안정화 (1.1)
    - ✓ 설치 매뉴얼 및 싱글 클러스터용 자동 설치 지원 (1.2)
    - ✓ 플러그인 아키텍처 도입 (1.3)
    - ✓ GPU 자원 고급 할당 및 스케줄러 기능 강화 (1.4)
- 로드맵
  - 대규모 HPC 클러스터 오케스트레이션 기능 강화 (1.5)
  - VPN 없는 다중 클라우드 연결을 통한 쉬운 Hybrid cloud 및 on-premise 연동 (1.4 / 1.5)
  - 오토스케일링 기능 강화
    - ✓ 장시간 연산 세션을 위한 scale-in protection
    - ✓ cpu/memory/gpu slot 여유 용량에 따라 cold/hot instance group 운영

Backend.AI 오픈소스

- Backend.AI Ground
  - 오픈소스 (코어 + 라이브러리 + 플러그인)
  - 설치형 소프트웨어 / 프레임워크
  - 듀얼 라이센스 채택
    - ✓ 비상업용: LGPLv3
    - ✓ 상업용 및 재배포: 별도 commercial license 및 사용 계약

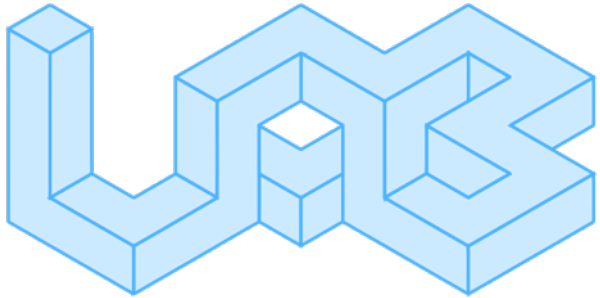
- Backend.AI Cloud
  - Backend.AI Ground 기반으로 Lablup에서 직접 운영하는 클라우드

- Backend.AI Enterprise
  - 상업 용도의 별도 계약에 따른 엔터프라이즈 솔루션
  - 온프레미스 / 서포트 플랜

감사합니다!

질문은 contact@lablup.com 으로!

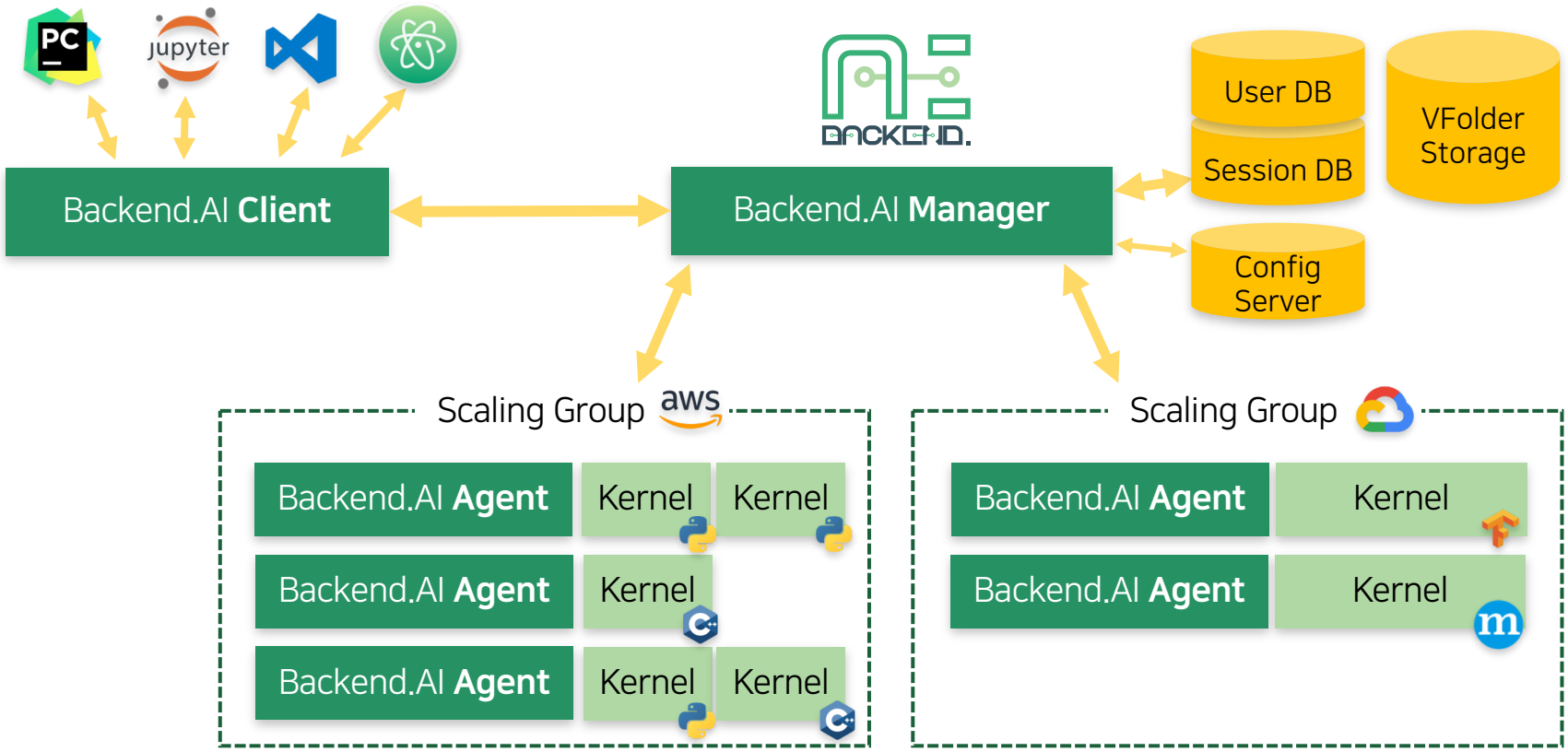
Lablup Inc.	https://www.lablup.com
Backend.AI	https://www.backend.ai
Backend.AI GitHub	https://github.com/lablup/backend.ai
Backend.AI Cloud	https://cloud.backend.ai
CodeOnWeb	https://www.codeonweb.com



Appendix

Backend.AI 소개

Backend.AI 컴포넌트 구성



Backend.AI 작동 예시

실행하고자 하는 소스 코드

```
import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

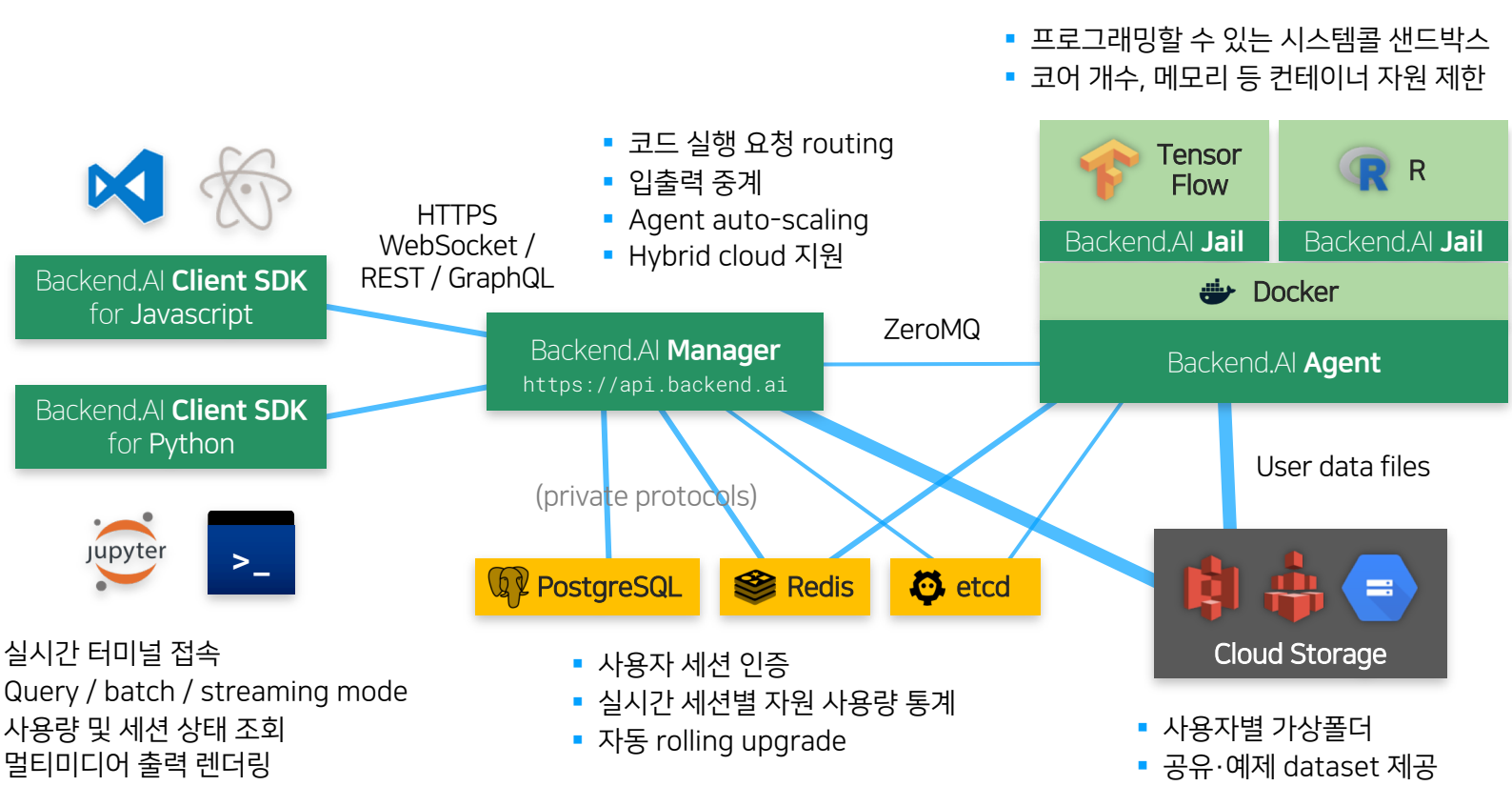
자신의 컴퓨터에는 개발환경이 없음

```
(backend.ai-client-py) ❯ python main.py
Traceback (most recent call last):
  File "main.py", line 1, in <module>
    import tensorflow as tf
ImportError: No module named 'tensorflow'
```

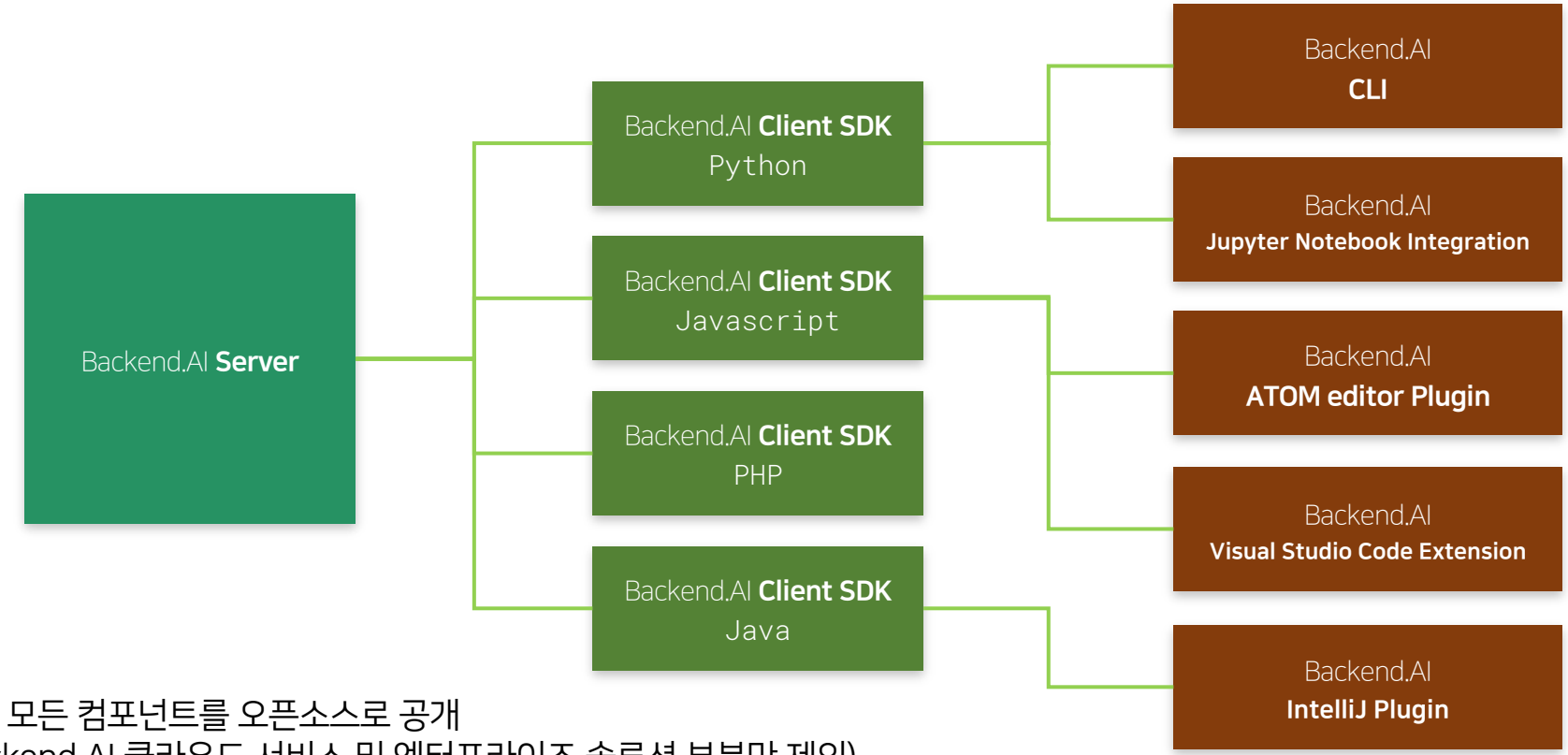
Backend.AI Cloud를 통해 실행하면 OK!

```
(backend.ai-client-py) ❯ backend.ai run python-tensorflow:latest main.py
✔ Session eae5e62d532b3dd94d96bf5099446021 is ready.
Uploading files: 100%|                                  | 109/109 [00:00<00:00, 1.33kbytes/s, file=main.py]
✔ Uploading done.
python-kernel: skipping build phase due to missing "setup.py" file
✔ Build finished. (exit code = 0)
b'Hello, TensorFlow!'
✔ Finished. (exit code = 0)
```

Backend.AI 상세

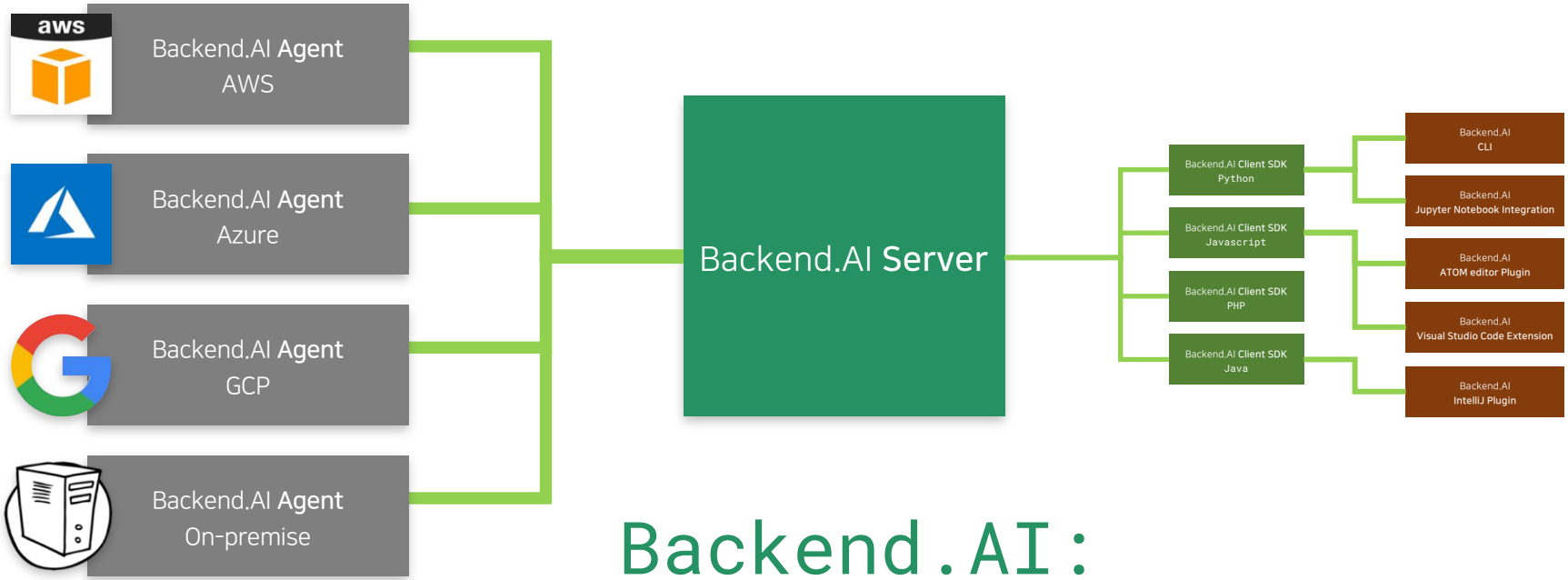


Backend.AI: 개발, 확장, 통합



거의 모든 컴포넌트를 오픈소스로 공개
(backend.AI 클라우드 서비스 및 엔터프라이즈 솔루션 부분만 제외)

Backend.AI: 하이퍼 클라우드



Backend.AI:

The **only All-in-one framework** for
Machine Learning Training PaaS

Backend.AI 주요 기능

- 빠른 시작
  - 클라우드 가입만으로 바로 사용 가능
  - 사용자의 요청 즉시 가상 프로그래밍 환경 생성
- 다양한 요구사항 충족
  - 모든 주요 프로그래밍 언어와 런타임 지원
    Python, R, Julia, Octave, PHP, Go, C/C++, Java, NodeJS, Lua, Haskell, Rust
  - 동일한 기계학습 라이브러리의 여러 버전 동시 지원
    TensorFlow, Caffe, PyTorch, Keras
- 친숙한 사용자 경험 + 개발자 친화적 프레임워크
  - 기존 연구·개발자들에게 익숙한 환경과의 통합 지원 (코드편집기, 웹기반 연구노트)
    Jupyter Notebook, Visual Studio Code, Atom Editor, IntelliJ beta
  - `$ backend.ai run` 명령줄 및 클라우드 인터프리터·컴파일러 지원
  - 개발자를 위한 HTTP 기반 공용 API 및 언어별 SDK 제공
    Python, Javascript (Node.js), JAVA beta ,PHP beta

Backend.AI

기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 플랫폼

- 다양한 기계학습 라이브러리 지원
  - TensorFlow, PyTorch, Caffe 등
- 여러 버전의 라이브러리 동시 지원
  - 예) TensorFlow 1.2~1.10을 동일 서버팜에서 동시 운영 지원
  - 예) PyTorch와 TensorFlow를 간섭 없이 동일 팜에서 지원
- 기계학습 라이브러리 자동 업데이트 지원
- CPU / GPU / RAM 동적 할당
  - 훈련마다 다른 연산 자원 할당 지원
    - ✓ 예) 4CPU + 2GPU + 64GB
  - 멀티 GPU 지원
  - GPU 컴퓨트코어 및 램 파티셔닝 지원 (Nvidia)
- GPU 지원
  - Nvidia CUDA 기반 가속 (TensorFlow / PyTorch+Caffe)
  - AMD ROCm 기반 가속 (TensorFlow) *

*베타 테스트 중

Backend.AI

기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 플랫폼

- 온라인 모델 사용 (1.5*)
  - use.backend.AI 기반의 모델 서빙
  - 함수처럼 딥러닝 모델 사용
  - Backend.AI SDK를 통한 다양한 언어 지원
    - ✓ Python / JAVA BETA / Node.js / JavaScript (on browser) / Microsoft Excel / PHP BETA
- 개발한 모델의 서빙 지원 (1.6†)
  - 직접 개발한 모델의 서빙 지원
  - 기존 모델의 데이터 기반 커스텀 트레이닝 지원

*베타 테스트 중

† 개발중 (2018년 하반기)

Backend.AI

기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 플랫폼

- 온프레미스 서버 설치
  - 물리 서버 및 VM 설치 모두 지원
  - OpenStack 설치 지원*
- 다양한 클라우드 지원
  - Amazon, Microsoft, Google 클라우드 지원
- 이기종 클라우드 통합 지원
  - 예) Amazon + Microsoft
  - 예) On-premise + Amazon
  - 클라우드 통합을 위한 편의 지원

*베타 테스트 중

Backend.AI

기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 플랫폼

- 연구자 및 개발자를 위한 IDE 통합
  - IntelliJ IDEA (PyCharm 포함), Visual Studio Code, ATOM editor, Jupyter Notebook 통합
  - TensorBoard 등의 모니터링 툴 설치 및 접근 지원
- 프로토타이핑 스케일링 시나리오
  - 로컬에서 테스트 후 스케일
  - 모든 워크로드를 서버에서
- 확장을 위한 Backend.AI SDK
  - Python 3 / Node.js / JAVA / PHP SDK 지원

Backend.AI

기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 플랫폼

- API 기반
  - 완전한 문서 지원 (영어)
  - 온/오프라인 컴포넌트 설치
    - ✓ pip / npm / composer 기반의 설치 지원
  - 오픈소스 생태계
- 완전 비동기 I/O 기반의 코드
  - 짧은 지연시간
  - 각 컴포넌트 동작의 독립성 향상
- 모던 언어 및 컨테이너 환경 기반
  - Python 3.6 + aiodocker
  - Docker 기반의 컨테이너 가상화

지원 프로그래밍 언어

- 현재
  - Python 3, R, PHP 5/7, node.js, JavaScript, Julia, Lua, Octave, Go, C/C++, Rust, Java, Haskell
  - 딥러닝 환경: TensorFlow, **PyTorch**, Keras, Caffe
- 테스트중
  - Swift, TypeScript, C# (.NET Core)

기술적 강점

- Backend.AI Cloud / Open Source
  - API 중심 구현
    - ✓ API 기반으로 확장가능하고 스케일러블하며 재사용가능하고 타 솔루션과 유연하게 결합 가능
  - 다양한 프로그래밍 언어 지원
    - ✓ 다양한 프로그래밍 언어 및 환경을 지원하므로 거의 모든 사용 사례 지원
- 머신러닝용 컨테이너 관련 기술들을 단일 프레임웍으로 제공하는 유일한 솔루션
  - 짧은 지연시간과 고밀도의 컨테이너 풀링
    - ✓ 스케일시 초단위의 컨테이너 스포닝
  - 멀티테넌트 환경에서의 GPU 가속 지원
    - ✓ Faster native GPU performance compared to VM-based solutions
  - 동적 샌드박싱: 프로그래밍 및 재작성 가능한 syscall 필터
    - ✓ Apparmor/seccomp 등 대비 풍부한 프로그래밍 가능한 정책 지원
  - Docker 기반의 레거시 앱 리소스 제한
    - ✓ 예) 현재 Docker의 경우 다양한 머신러닝 컴포넌트들의 CPU 코어 제한을 강제할 수 없음 (OpenBLAS (matrix calculation library) 등).

Backend.AI와 기존 유사 오픈소스 비교

- 기존 오픈소스 구현체들의 한계 해결
  - 특정 프로그래밍 언어 지원 → 범용 언어 지원
  - REPL 구현체들의 취약한 보안 → 보안 중심의 원천적 설계로 실서비스 구축이 가능
  - 기계학습 개발자를 위한 편의 기능 제공 (가상폴더 등)
  - 연산 가속을 위한 보조프로세서 지원 (GPU, TPU, ROCm 등)
  - 연산에 필요한 자원의 자동 스케일링
  - 온디맨드 대규모 코드 연산 기능 제공

Backend.AI 의 특징 비교 (1/2)

- vs. Anaconda Cluster
  - Anaconda Cluster는 고정된 수의 서버를 가진 클러스터에 ssh 기반으로 각종 데이터 분석용 Python/Java 프로그램들을 자동 설치하고 마스터 서버를 통해 통합 실행하는 기능 제공
  - Backend.AI는 클라우드 기반의 서버 자동 스케일링과 컨테이너 기반의 보안 격리된 실행 환경을 제공하여 여러 사용자가 클러스터뿐만 아니라 서버 수준에서도 고밀도 자원 공유 가능
  - 예) Anaconda Cluster를 Backend.AI 에서 생성해서 쓰고, 다 쓴 후 쉽게 없앨 수 있음.

- vs. Google Colaboratory / AWS SageMaker / Azure MLStudio / 각종 MLaaS
  - 클라우드 플랫폼 기반의 관리형 기계학습 실습 환경 제공
  - Backend.AI는 자신의 서버나 벤더 상관 없이 자가 소유한 클라우드 인프라에 직접 설치 가능하여, 퍼블릭 클라우드 벤더들이 제공하는 편의성을 프라이빗 클라우드에서도 제공

Backend.AI 의 특징 비교 (2/2)

- vs. Kubernetes (Google Borg) / Apache Mesos / Apache Aurora
  - Mesos와 k8s 모두 클러스터 자원 할당 및 관리를 위해 일반화된 오픈소스 솔루션
  - Aurora는 Mesos를 기반으로 service 및 job pipeline 관리 추가 제공
  - Backend.AI는 오케스트레이션 계층 뿐만 아니라, 머신러닝 개발자들을 위해 특화되어 가상폴더, AI 프레임워크 버전 관리, 분산 학습을 통한 적정 모델 찾기 등을 지원
  - 오토스케일링 등 클라우드 환경에 적합한 기능과, 스케쥴링 등 온프레미스 환경에 적합한 기능을 추가 제공
  - 연구자 / 개발자 등을 위한 다양한 엔드유저 편의성을 제공함

- vs. Ansible / Chef / Puppet / Terraform / ...
  - 대규모 서버 및 컨테이너를 프로비저닝하고 인프라 구성을 코드화하여 관리할 수 있는 DevOps용 도구
  - Backend.AI는 이런 기술들을 종합적으로 활용하여, 기계학습 개발자들이 인프라에 대한 고민을 하지 않도록 한 단계 더 감싸주는 역할을 함

Backend.AI 트리비아

- 첫 버전 발표: 2015년 8월 (PyCon KR)
- 첫 공개 테스트 버전 배포: 2016년 11월
- 첫 정식 버전 배포: 2017년 11월
- 클라우드 등록 사용자 : ~9000명
- 실사용 기업, 연구소 및 대학: 5
- Backend.AI가 지원하는 가장 작은 단일 노드: 라즈베리 파이 3
- Backend.AI가 지원하는 가장 큰 단일 노드: NVidia DGX-1
- 단일 Backend.AI 클러스터가 동시 지원하는 노드 수: 2000 노드
- 멀티 Backend.AI 클러스터가 동시 관리하는 노드 수: 62500 노드
- Backend.AI 설치를 지원하는 클라우드 환경: Amazon Web Service, Google Cloud Computing, Microsoft Azure 및 OpenStack

Backend.AI 응용 예시

- GPU 클라우드 서비스의 손쉬운 구축
  - Backend.AI를 활용한 GPU 클라우드 사업화
- 의료기관 및 금융기관 등 고도보안 망분리 환경에서의 사설 GPU 클라우드 구축
  - (엔터프라이즈 에디션) 오프라인 설치, SSO, 보안로깅 등 추가 지원
- 직접 보유한 GPU 서버팜의 용량이 부족할 때 cloud.backend.ai로 동적 용량 확장
  - 또는 반대로 cloud.backend.ai에 자신의 GPU 워크스테이션을 등록하여 원격 관리
- 고성능 컴퓨팅(HPC) 및 과학시각화(HPCV) 지원
  - 클러스터의 서비스와 애플리케이션 배포 단순화로 최신 버전 유지 및 성능 향상

Backend.AI 기술 현황 / 로드맵

기술 특징		nvidia-docker	Docker Swarm	OpenStack	Kubernetes	Apache Mesos	Backend.AI (현재)	Backend.AI (목표)
GPU Support	GPU Assignment & Isolation	✔			✔	✔	✔	✔
	Heterogeneous Accelerators							✔
	Partial GPU Sharing						✔ *	✔
Security	Sandboxing via Hypervisor/Container	✔	✔	✔	✔	✔	✔	✔
	Programmable Sandboxing						✔	✔
Virtualization	VM (Hypervisor)			✔	✔ **	✔ **	✔ **	✔ **
	Docker Container	✔	✔	✔	✔	✔	✔	✔
Scheduling	Availability-slot based		✔	✔	✔	✔	✔	✔
	Advanced				✔ ***	✔		✔
Integration	Modern AI Farmeworks							✔

* TensorFlow 프레임워크에 한하여 기술 테스트 완료
** VM 관리를 직접 수행하지 않고 클라우드 벤더 API 또는 OpenStack에 의존
*** slot 기반이긴 하지만 label 기능으로 상세한 사용자화 가능