DEVIEW
2019

챗봇 1만 개의 모델 서빙하기
: AI 서비스 어디까지 해봤니

고석현
Clova AI Business

NAVER

CONTENTS

DEVIEW
2019

1. 모델의 성격 파악하기
2. Amdahl's law 를 기억하며
3. 조금만 더 줄어 주면 안될까?
4. 모델 정말 1만개를 서비스 했을까

DEVIEW
2019

1.모델의 성격 파악하기

1.1 텍스트 ? 이미지 ? 음성 ?

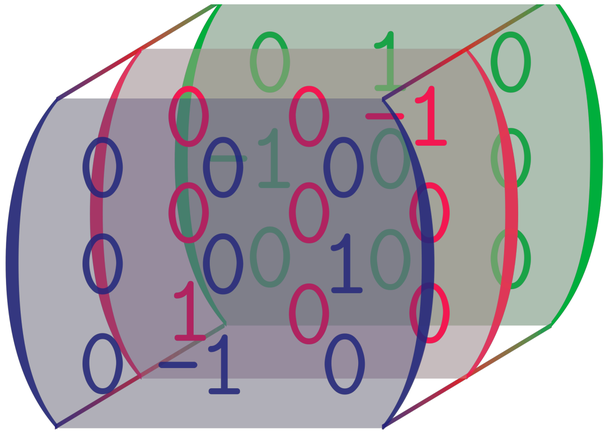
DEVIEW 2019

모델에 들어가는 인풋은 결국 어떤 텐서 형태의 피쳐 일것

text

image

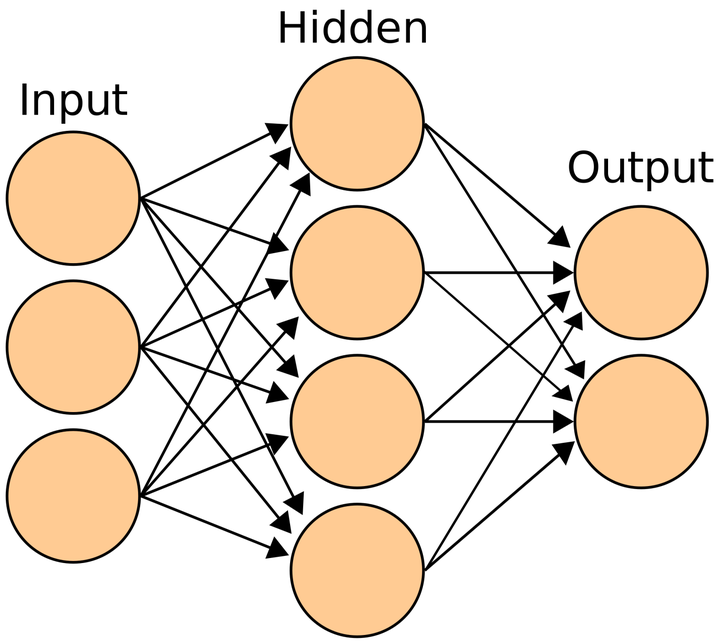
speech



1.1 텍스트 ? 이미지 ? 음성 ?

DEVIEW 2019

실제로 중요한건 모델의 규모와 서비스 형태



1.1 텍스트 ? 이미지 ? 음성 ?

서비스 요구 사항에 따라 서버, 모바일, 임베디드등 여러 환경 고려 필요

1.2 안되는것을 빠르게 포기하는 방법

DEVIEW
2019

데이터 없이 학습 없이 이루어지는 마술같은 ML

1. 수백 ~ 수천 토큰의 질의를 이해하는 모델
2. 도메인 전용 학습, 튜닝이 필요하지 않는 범용 모델
3. 10턴 이상의 문맥을 고려하여 이해하는 모델

1.3 되는걸 빠르게 파악하기

DEVIEW
2019

구축 -> 훈련 -> 평가 -> 개선으로 이어지는 흐름을 자동화

1. 자동으로 데이터셋을 개선하기, 정리하기 (강재욱님 발표)
2. 자동으로 모델을 평가하고 튜닝하기 (2018 DEVIEW 이재원님 발표)
3. 분산 환경과 파이프 라인으로 최대한 빨리 모델 학습하기
4. 재학습, 운영, 관리, 지표화를 통해서 통제 가능하고 운영 가능한 머신러닝 서비스 만들기
5. 유저의 암시적 피드백을(IRF) 통해서 데이터 개선하기

1.4 모델의 비용 계산하기 1

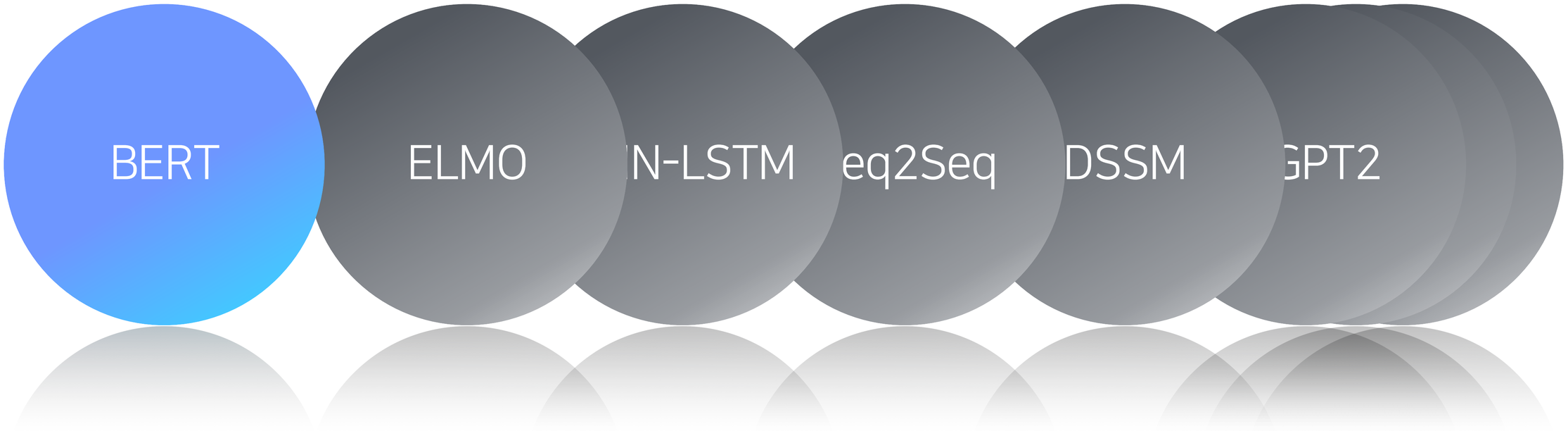
- 베이스 라인과 모델 앙상블의 중요성
- 언어별, 데이터 규모별 강점을 가지는 다양한 모델이 존재
- SOTA 하나로 모든 데이터를 최적으로 서비스 하는것은 어려움

https://wormwlrm.github.io/kwakcheolyong/

1.4 모델의 비용 계산하기 2

DEVIEW
2019

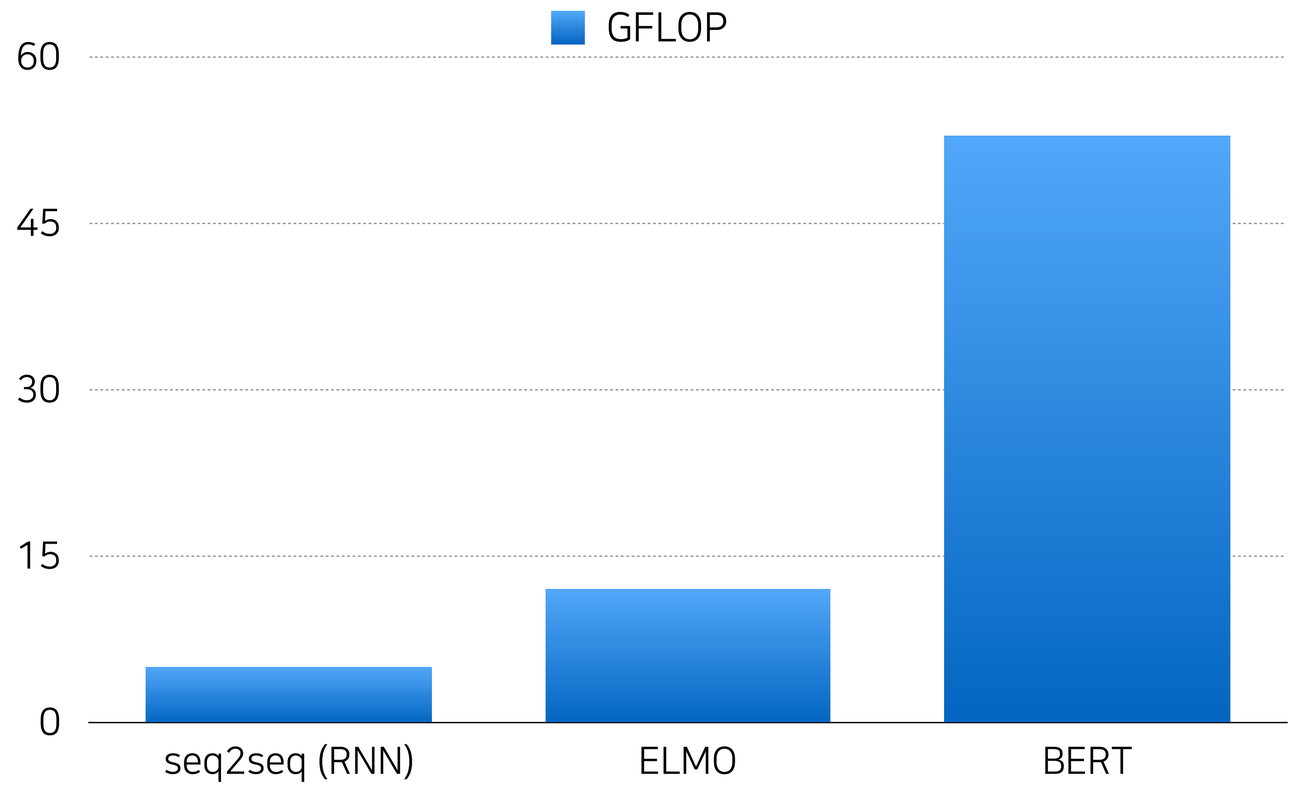
- 베이스 라인과 모델 앙상블의 중요성
- 언어별, 데이터 규모별 강점을 가지는 다양한 모델이 존재
- SOTA 하나로 모든 데이터를 최적으로 서비스 하는것은 어려움



1.4 모델의 비용 계산하기 3

DEVIEW 2019

- BERT(Transformer) 계열과 LSTM(RNN) 계열 모델의 계산 비용 비교



1.4 모델의 비용 계산하기 4

인공지능도 돈을 벌어야 합니다. 이제는 비용을 이야기 해야 합니다.

1. 모델 학습에 필요한 비용이 저렴하고
2. 사전 학습된 모델을 도메인 별 fine-tuning 하여 이용 가능하며
3. 모델의 연산 비용이 충분히 작고 최적화 되어 있으며
4. 모델의 물리적인 용량이 작아 한정된 인프라에서 다수를 서비스 할 수 있으며
5. 동시에 충분한 성능(정확도)를 보장해야 합니다.

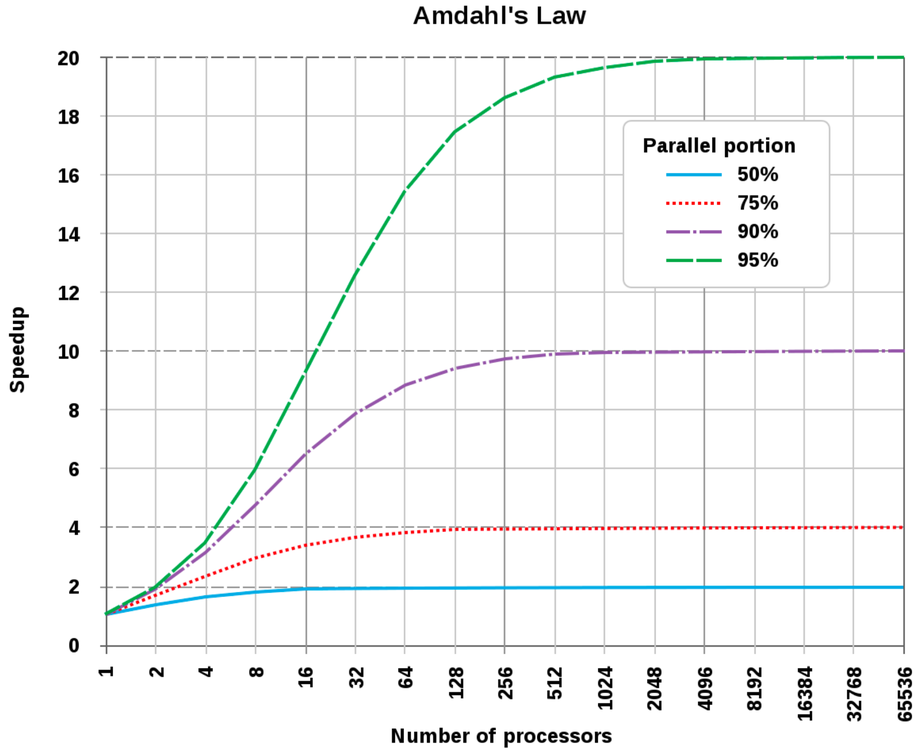
0. 즉 비용을 고려한 최적화가 필요합니다.

2. Amdahl's law 를 기억하며

2.0 Amdahl's law

DEVIEW 2019

무한의 컴퓨팅 자원을 투입 하더라도



Definition [edit]

Amdahl's law can be formulated in the following way:

$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}}$$

where

- S_{latency} is the theoretical speedup of the execution of the whole task;
- s is the speedup of the part of the task that benefits from improved system resources;
- p is the proportion of execution time that the part benefiting from improved resources originally occupied.

Furthermore,

$$\begin{cases} S_{\text{latency}}(s) \leq \frac{1}{1-p} \\ \lim_{s \to \infty} S_{\text{latency}}(s) = \frac{1}{1-p}. \end{cases}$$

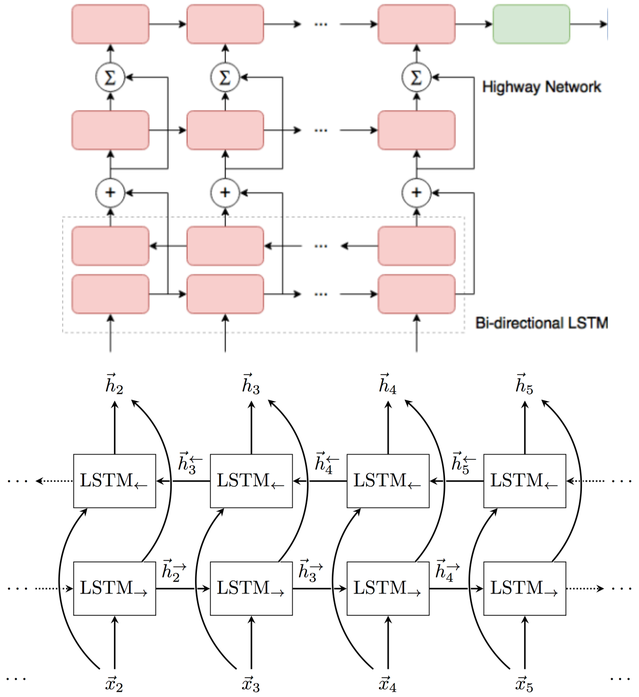
2.1 범용환경 ? 전용환경 ?

언제나 네이버 데이터 센터에 있다면 좋겠지만

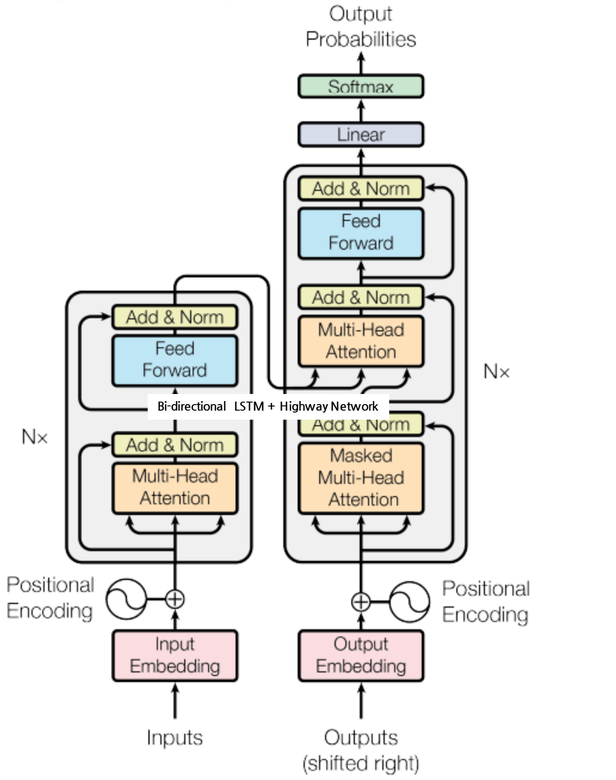
2.2 빠른 모델이 좋은 모델일까 ?

DEVIEW 2019

계산 수행 시간과 총 계산 비용은 별도의 문제



Bi-directional LSTM + Highway Network



Bidirectional Encoder Representations from Transformers (BERT)

2.3 아니 경제적인 모델

DEVIEW
2019

학습과 추론의 성능 최적화가 곧 비용인 시대

Artificial Intelligence / Machine Learning

Training a single AI model can emit as much carbon as five cars in their lifetimes

Deep learning has a terrible carbon footprint.

by **Karen Hao** Jun 6, 2019

The artificial-intelligence industry is often compared to the oil industry: once mined and refined, data, like oil, can be a highly lucrative commodity. Now it seems the metaphor may extend even further. Like its fossil-fuel counterpart, the process of deep learning has an outsize environmental impact.

https://www.technologyreview.com/s/613630/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/

DEVIEW
2019

3.조금만 더 줄어 주면 안될까?

3.1 모델과 대화하기 그리고 설득하기

DEVIEW
2019

모델 레이어, 파라메터 튜닝

L1 가지 치기

레이어간 파라메터 공유

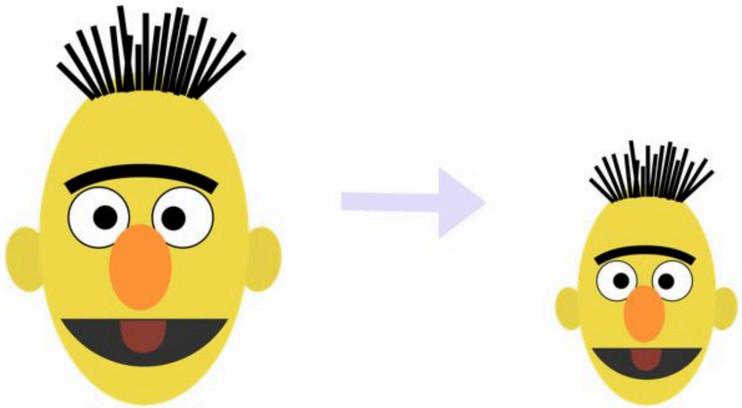
컴퓨팅 자원 친화적인 연산으로 교체

특정 장비별 오퍼레이션 최적화

3.1 모델과 대화하기 그리고 설득하기

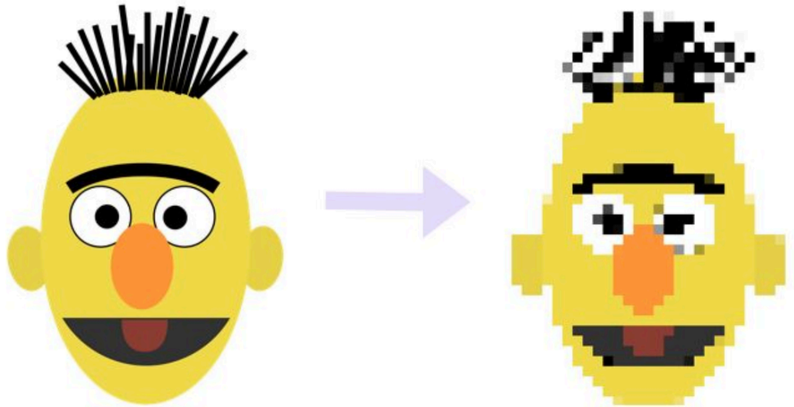
어떤 방법은 대가와 비용이 따릅니다.
모델의 정확도(성능)를 일부 포기해야 합니다.

Knowledge distillation



https://blog.rasa.com/compressing-bert-for-faster-prediction-2/

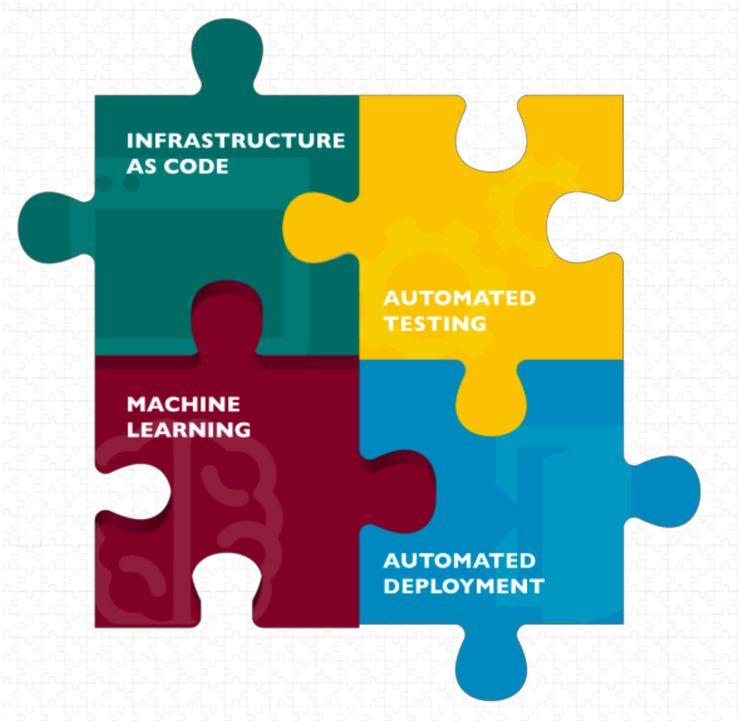
Quantization



3.2 그래도 안 될 거예요. 모델러와 소통하기

DEVIEW 2019

공동의 목표와 성과로 이어져야 하고 서비스와 제품으로 묶여야 합니다.



챗봇 빌더

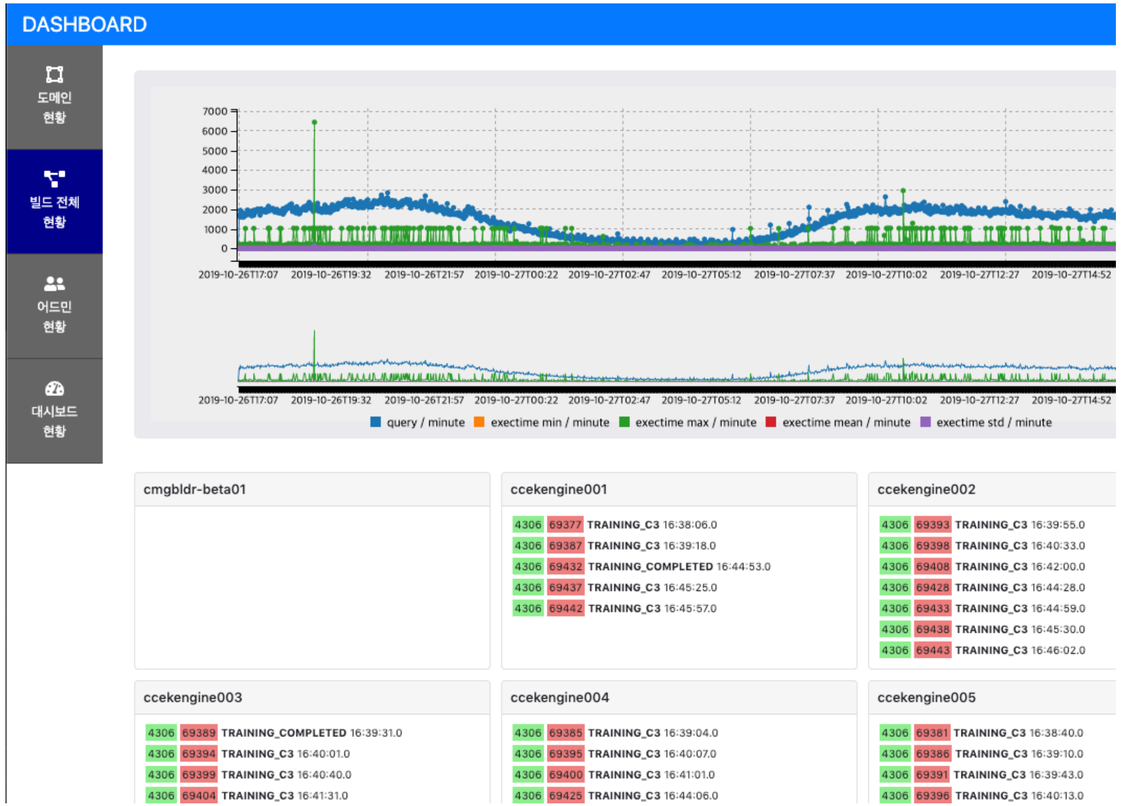
N

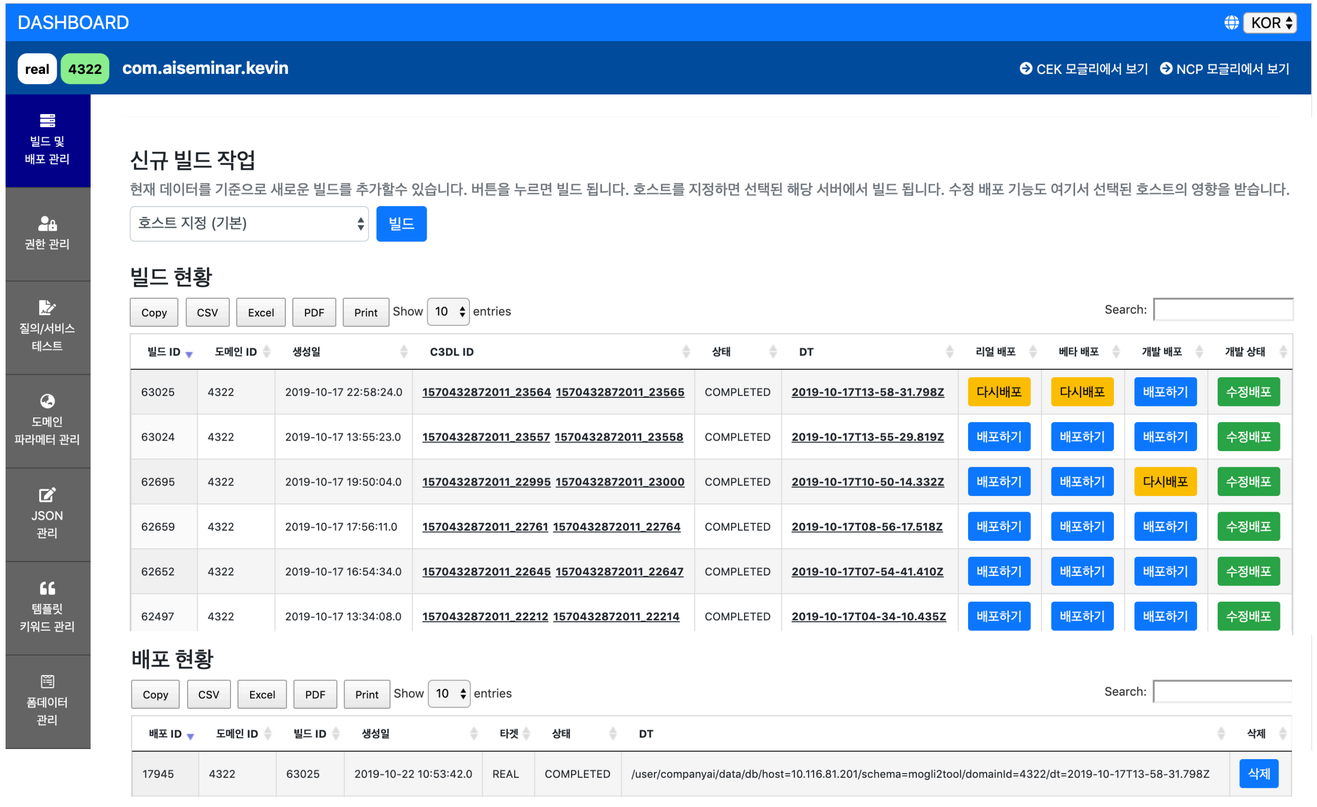
LINE

3.2 그래도 안 될 거예요. 모델러와 소통하기

DEVIEW 2019

공동의 목표와 성과로 이어져야 하고 서비스와 제품으로 묶여야 합니다.





3.2 그래도 안 될 거예요. 모델러와 소통하기

DEVIEW 2019

공동의 목표와 성과로 이어져야 하고 서비스와 제품으로 묶여야 합니다.

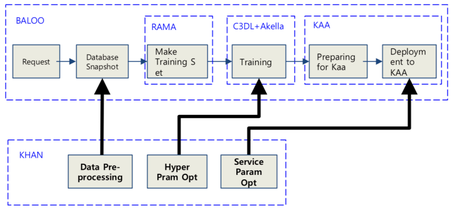
Khan

AutoML Project for Chatbot AI Builder

Overview

- End2End CB building pipeline
- Clean dataset
- Select appropriate features
- Select an appropriate models fam
- Optimize hyper-parameters
- Optimize service-parameters
- Automated model validation

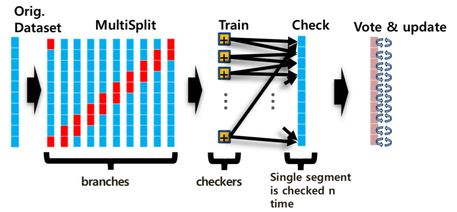
Chatbot Building Process



Automated Feature Eng.

- Intent and slot clustering
- Noisy query filtering
- Auto label error correction

PICO: Auto Lable Correction



Hyper-Param Opt.

- Coordinate search
- Genetic algorithm
- Bayesian optimization

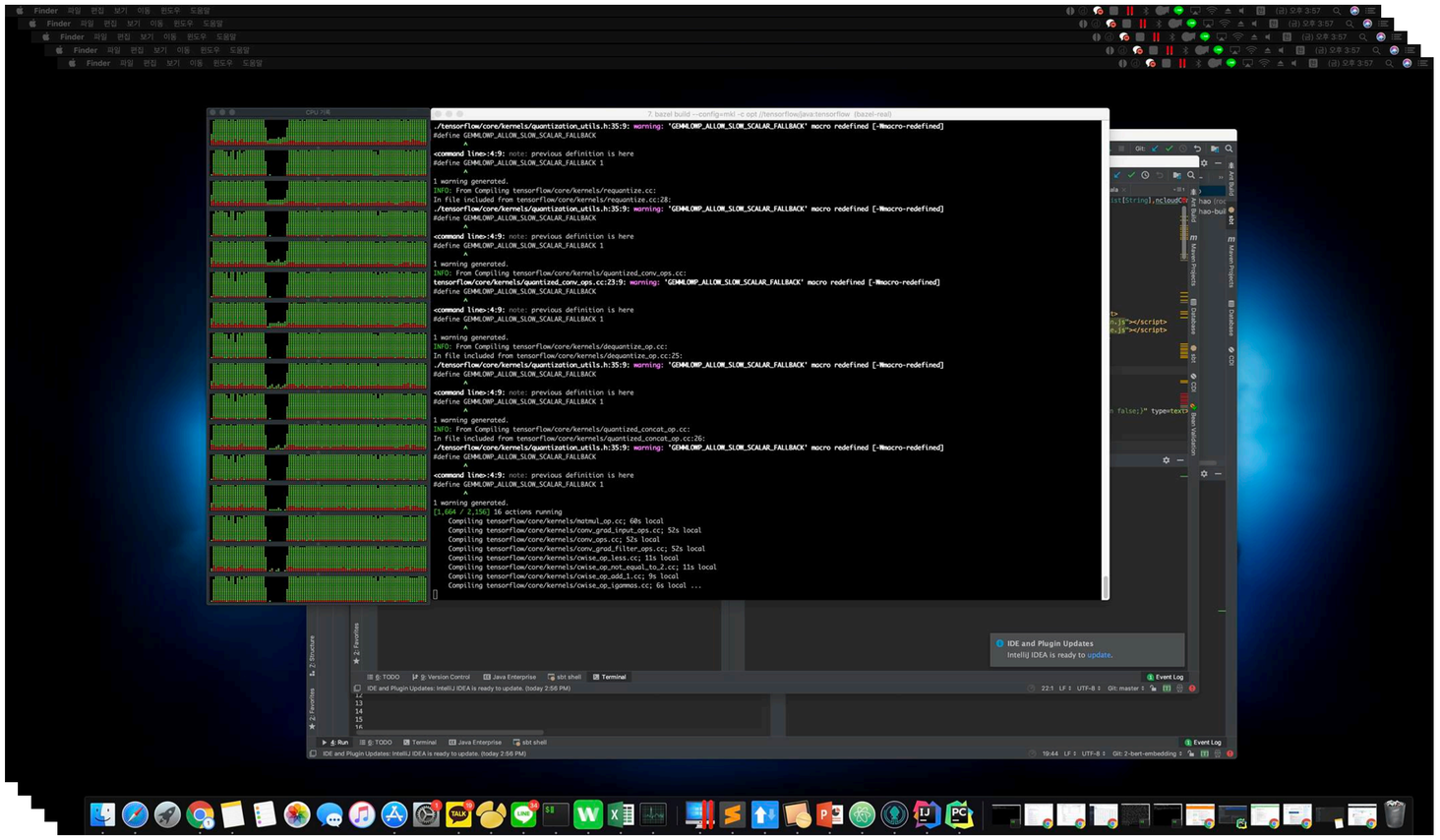
Coordinate Search for HiddenSizes

3.3 그래도 잘 안될거예요 이제 엔지니어에게 남은 모든걸 하기

DEVIEW 2019

프레임워크의 코드도 커스터마이징을 합니다.

수 없이 빌드와 테스트도 합니다. 연산 라이브러리도 교체합니다.

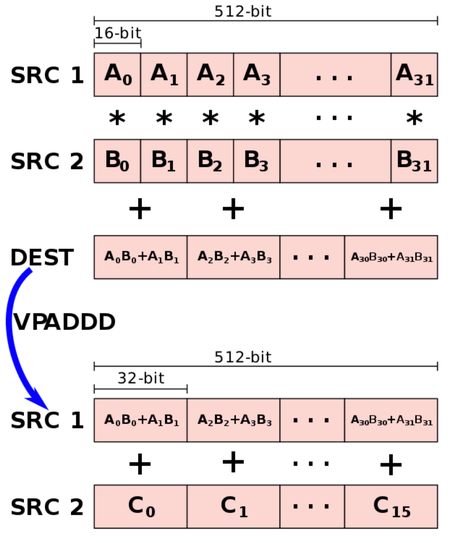


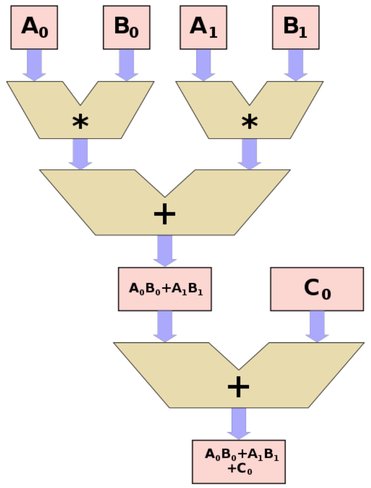
AVX-512 Vector Neural Network Instructions (VNNI) - x86

< x86

AVX-512 Vector Neural Network Instructions (AVX512 VNNI) is an x86 extension, algorithms.

VPMADDWD





VPADDD

3.3 그래도 잘 안될거예요 이제 엔지니어에게 남은 모든걸 하기

DEVIEW 2019

JNI(Java Native Interface) 수준으로 서비스 코드와 결합도 시도합니다.

네이티브 바이너리도 시스템과 환경별로 생성합니다.

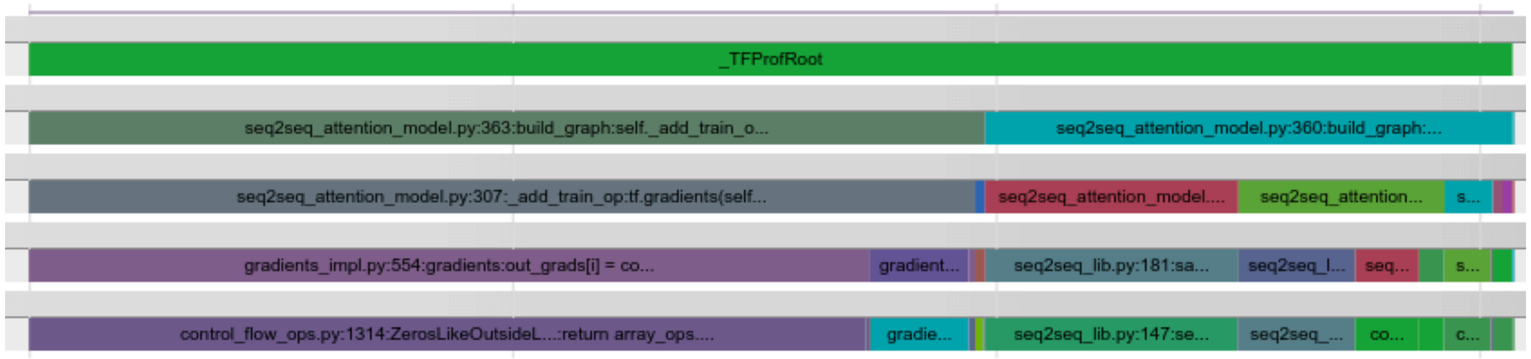
```scala
import java.nio.IntBuffer

import org.tensorflow.{Graph, Session, Tensor}

import scala.collection.JavaConverters._

class TensorFlowProvider(model: Model) extends AutoCloseable {

  private val graph: Graph = {
    val graph = new Graph()
    graph.importGraphDef(model.getBytes)
    graph
  }

  private val session: Session = new Session(graph)
```

```
[                  test tensorflow]$ ldd libtensorflow_framework.so.1
        linux-vdso.so.1 =>  (0x00007fff0994c000)
        libiomp5.so => /home                  /_bazel_irteam/e9fc3b58d0629b9
        libmklml_intel.so =>                  e/bazel/_bazel_irteam/e9fc3b58d
        librt.so.1 => /lib64/librt.so.1 (0x00007fa2700b7000)
        libpthread.so.0 => /lib64/libpthread.so.0 (0x00007fa26fe9b000)
        libdl.so.2 => /lib64/libdl.so.2 (0x00007fa26fc97000)
        libm.so.6 => /lib64/libm.so.6 (0x00007fa26f995000)
        libstdc++.so.6 => /lib64/libstdc++.so.6 (0x00007fa26f68e000)
        libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007fa26f478000)
        libc.so.6 => /lib64/libc.so.6 (0x00007fa26f0ab000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fa27a947000)
[                  test tensorflow]$ ldd java/libtensorflow_jni.so
        linux-vdso.so.1 =>  (0x00007ffc27e47000)
        libtensorflow_framework.so.1 (0x00007f0100c50000)
        libiomp5.so =>                  /bazel/_bazel_irteam/e9fc3b58d0629b9
        libmklml_intel.                 .cache/bazel/_bazel_irteam/e9fc3b58d
        libdl.so.2 => /lib64/libdl.so.2 (0x00007f00f8bc5000)
        libm.so.6 => /lib64/libm.so.6 (0x00007f00f88c3000)
        libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f00f86a7000)
        librt.so.1 => /lib64/librt.so.1 (0x00007f00f849f000)
        libstdc++.so.6 => /lib64/libstdc++.so.6 (0x00007f00f8198000)
        libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007f00f7f82000)
        libc.so.6 => /lib64/libc.so.6 (0x00007f00f7bb5000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f010d2d0000)
```

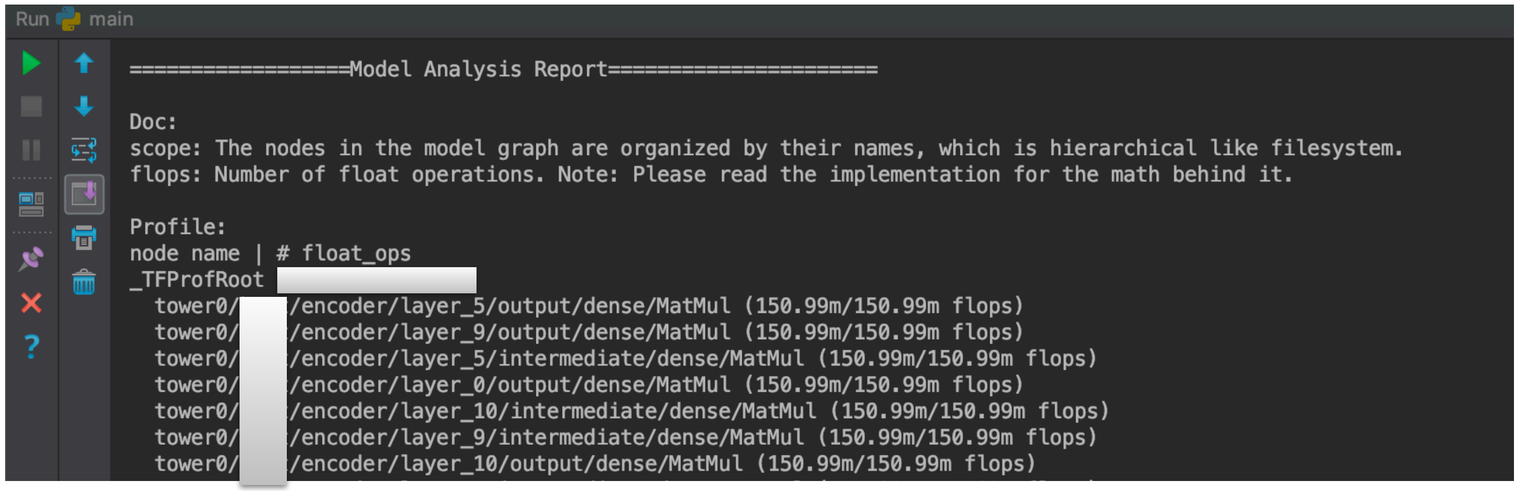
3.3 그래도 잘 안될거예요 이제 엔지니어에게 남은 모든걸 하기

DEVIEW 2019

프로파일링을 통해 모델의 계산 비용 분석도 진행합니다.



https://github.com/tensorflow/tensorflow/tree/master/tensorflow/core/profiler



3.3 그래도 잘 안될거예요 이제 엔지니어에게 남은 모든걸 하기

알려진 최적화들을 하나 하나씩 서비스에 적용합니다.

```
2019-10-15 22:00:16.824222: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 2500000000 Hz
2019-10-15 22:00:16.828214: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7f71cfbe9f20 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2019-10-15 22:00:16.828260: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
OMP: Info #212: KMP_AFFINITY: decoding x2APIC ids.
OMP: Info #210: KMP_AFFINITY: Affinity capable, using global cpuid leaf 11 info
OMP: Info #154: KMP_AFFINITY: Initial OS proc set respected: 0-79
OMP: Info #156: KMP_AFFINITY: 80 available OS procs
OMP: Info #157: KMP_AFFINITY: Uniform topology
OMP: Info #179: KMP_AFFINITY: 2 packages x 20 cores/pkg x 2 threads/core (40 total cores)
OMP: Info #214: KMP_AFFINITY: OS proc to physical thread map:
OMP: Info #171: KMP_AFFINITY: OS proc 0 maps to package 0 core 0 thread 0
OMP: Info #171: KMP_AFFINITY: OS proc 40 maps to package 0 core 0 thread 1
OMP: Info #171: KMP_AFFINITY: OS proc 4 maps to package 0 core 1 thread 0
OMP: Info #171: KMP_AFFINITY: OS proc 44 maps to package 0 core 1 thread 1
OMP: Info #171: KMP_AFFINITY: OS proc 8 maps to package 0 core 2 thread 0
OMP: Info #171: KMP_AFFINITY: OS proc 48 maps to package 0 core 2 thread 1
OMP: Info #171: KMP_AFFINITY: OS proc 65 maps to package 1 core 17 thread 1
OMP: Info #171: KMP_AFFINITY: OS proc 29 maps to package 1 core 18 thread 0
OMP: Info #171: KMP_AFFINITY: OS proc 69 maps to package 1 core 18 thread 1
OMP: Info #171: KMP_AFFINITY: OS proc 27 maps to package 1 core 19 thread 0
OMP: Info #171: KMP_AFFINITY: OS proc 67 maps to package 1 core 19 thread 1
OMP: Info #171: KMP_AFFINITY: OS proc 23 maps to package 1 core 20 thread 0
OMP: Info #171: KMP_AFFINITY: OS proc 63 maps to package 1 core 20 thread 1
OMP: Info #171: KMP_AFFINITY: OS proc 33 maps to package 1 core 24 thread 0
OMP: Info #171: KMP_AFFINITY: OS proc 73 maps to package 1 core 24 thread 1
OMP: Info #171: KMP_AFFINITY: OS proc 37 maps to package 1 core 25 thread 0
OMP: Info #171: KMP_AFFINITY: OS proc 77 maps to package 1 core 25 thread 1
OMP: Info #171: KMP_AFFINITY: OS proc 39 maps to package 1 core 26 thread 0
OMP: Info #171: KMP_AFFINITY: OS proc 79 maps to package 1 core 26 thread 1
OMP: Info #171: KMP_AFFINITY: OS proc 35 maps to package 1 core 27 thread 0
OMP: Info #171: KMP_AFFINITY: OS proc 75 maps to package 1 core 27 thread 1
OMP: Info #171: KMP_AFFINITY: OS proc 31 maps to package 1 core 28 thread 0
OMP: Info #171: KMP_AFFINITY: OS proc 71 maps to package 1 core 28 thread 1
OMP: Info #250: KMP_AFFINITY: pid 29837 tid 29947 thread 0 bound to OS proc set 0
2019-10-15 22:00:16.833505: I tensorflow/core/common_runtime/process_util.cc:115] Creating new thread pool with default inter op setting: 2. Tune using inter_op_parallelism_threads for best per
1
```

3.3 그래도 잘 안될거예요 이제 엔지니어에게 남은 모든걸 하기

DEVIEW 2019

시피유 데이터 시트를 뒤져봅니다. 물리적으로 연산장치 숫자가 적습니다. 인프라팀을 통해서 시제품 장비를 요청합니다.

다시 테스트를 하고 아직 구현 최적화에 문제가 있는 부분도 찾습니다. intel과 미팅도 합니다.

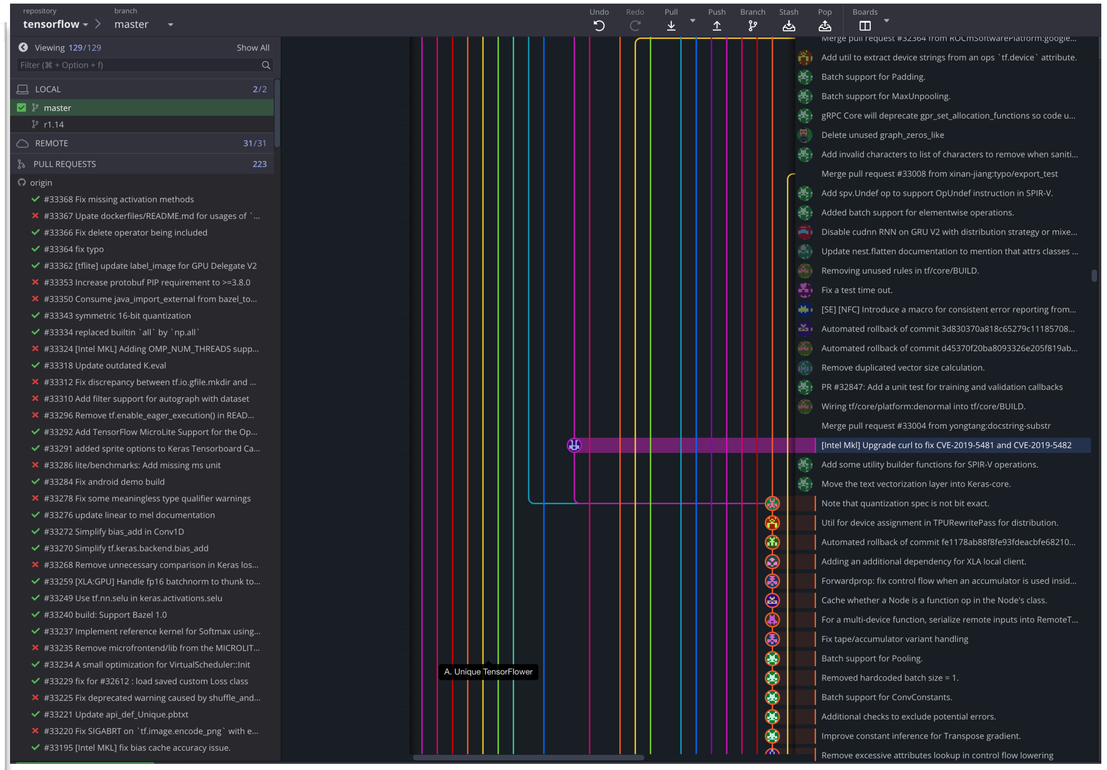
Products Solutions Support

intel

Support Home › Product Specifications › Processors

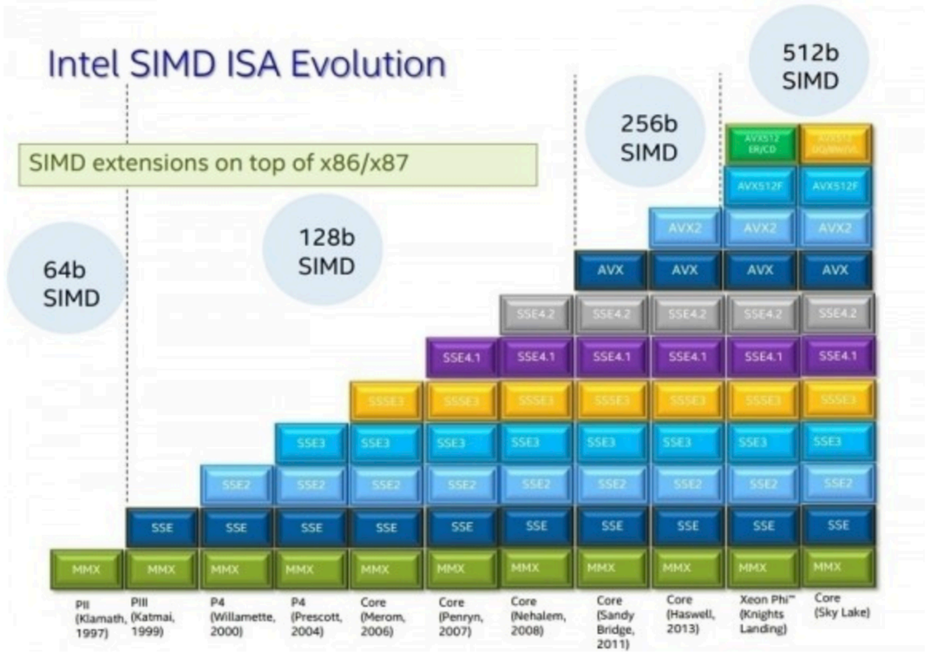
Advanced Technologies

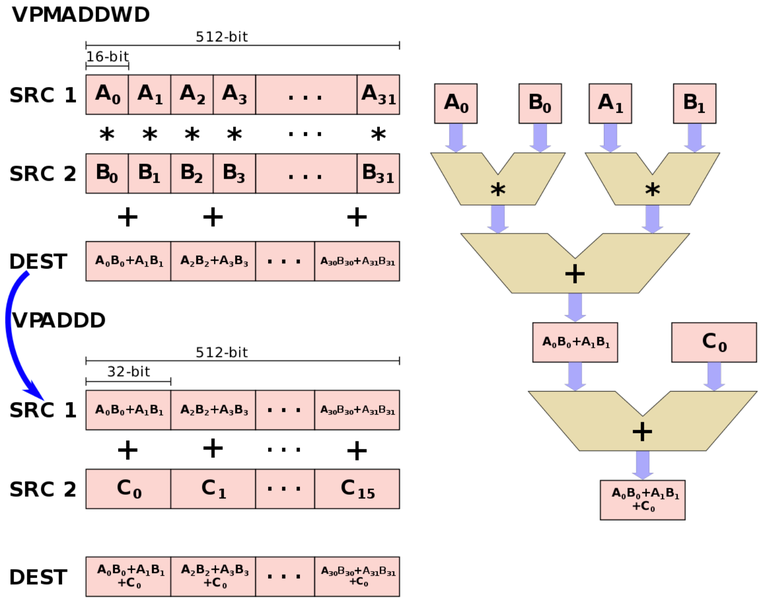
Intel® Optane™ Memory Supported ‡	No	No
Intel® Speed Shift Technology	Yes	Yes
Intel® Turbo Boost Max Technology 3.0 ‡	No	No
Intel® Turbo Boost Technology ‡	2.0	2.0
Intel® vPro™ Platform Eligibility ‡	Yes	Yes
Intel® Hyper-Threading Technology ‡	Yes	Yes
Intel® Virtualization Technology (VT-x) ‡	Yes	Yes
Intel® Virtualization Technology for Directed I/O (VT-d) ‡	Yes	Yes
Intel® VT-x with Extended Page Tables (EPT) ‡	Yes	Yes
Intel® TSX-NI	Yes	Yes
Intel® 64 ‡	Yes	Yes
Instruction Set Extensions	Intel® SSE4.2, Intel® AVX, Intel® AVX2, Intel® AVX-	Intel® SSE4.2, Intel® AVX, Intel® AVX2, Intel® AVX-51
# of AVX-512 FMA Units	1	2
Enhanced Intel SpeedStep® Technology	Yes	Yes
Intel® Volume Management Device (VMD)	Yes	Yes



3.4 AVX 512

확장 레지스터와 512bit SIMD 명령어를 지원하는 AVX-512

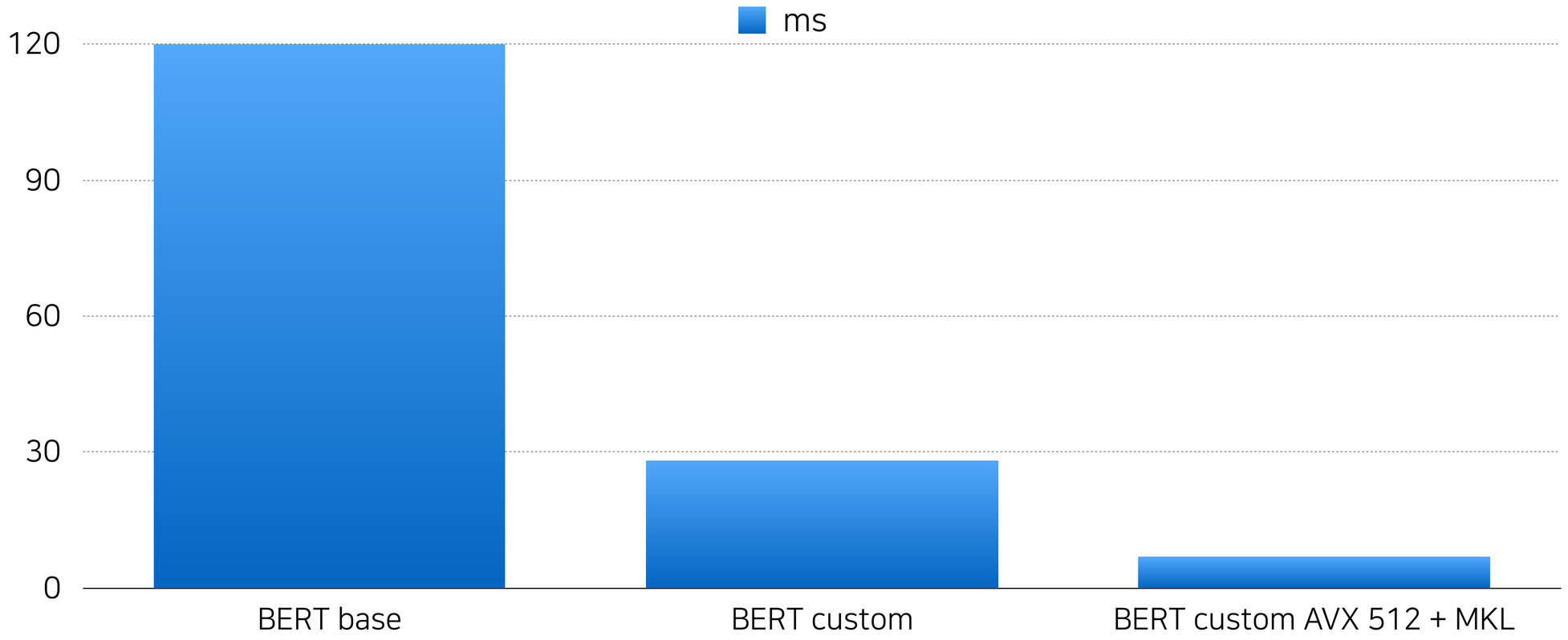




3.4 AVX 512

현재 챗봇 서비스에 사용하는 모델 기준 단일 추론 7ms 수준으로 개선

- batch 1, intel xeon sp2 2.5Ghz 20 core



3.5 AVX 512 VNNI

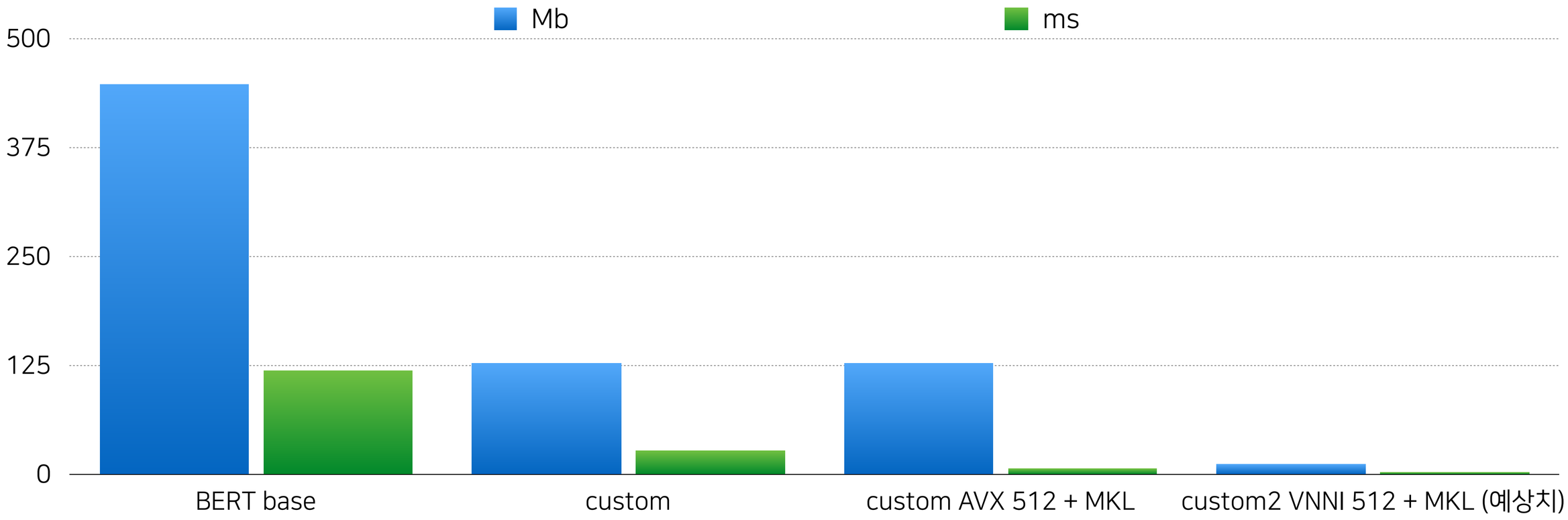
AVX-512 Vector Neural Network Instructions (VNNI)

- 전용 하드웨어의 개입 없는 없는 모델 양자화의 경우 실제 모델의 사이즈만 개선
- 모델 사이즈에 의한 메모리 제약 조건이 큰 모바일 탑재 환경에서만 유의미
- 하지만 AVX-512 VNNI등의 명령어는 하드웨어 수준에서 양자화된 모델 추론 가속

3.5 AVX 512 VNNI

DEVIEW 2019

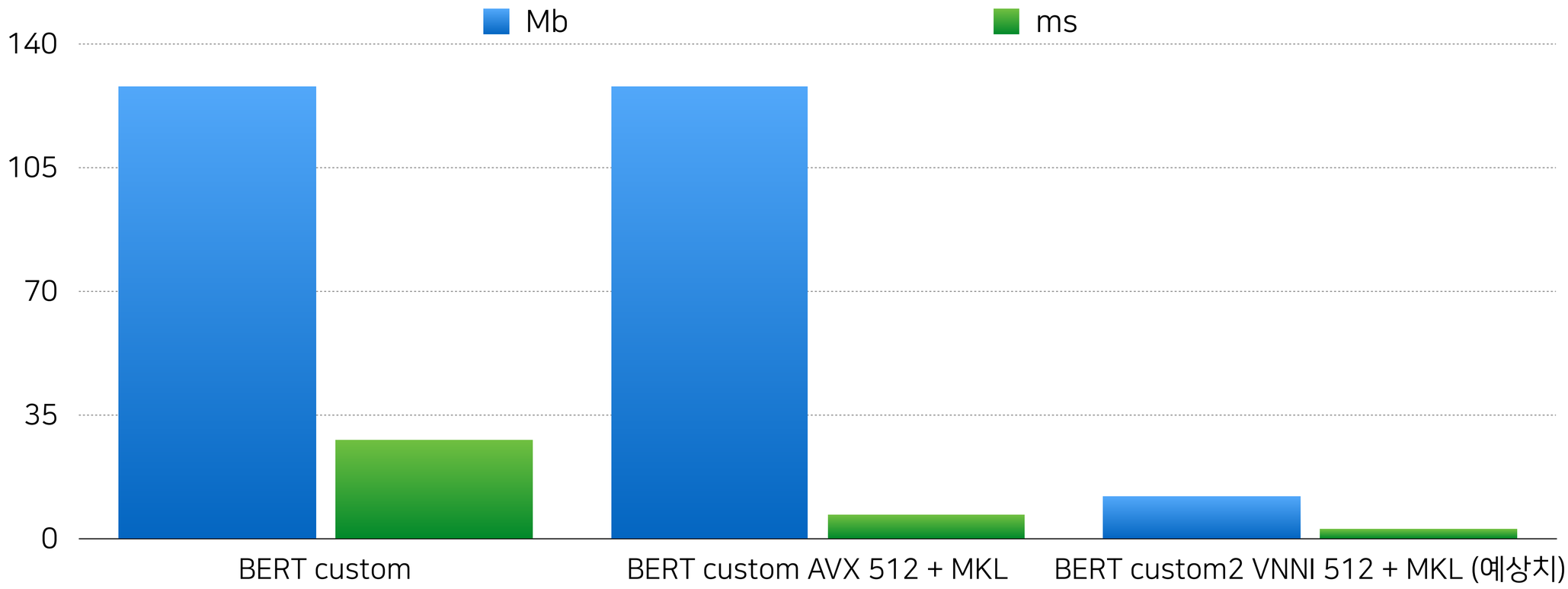
최신의 최적화 기법과 하드웨어 가속을 모두 적용한다면



3.5 AVX 512 VNNI

DEVIEW 2019

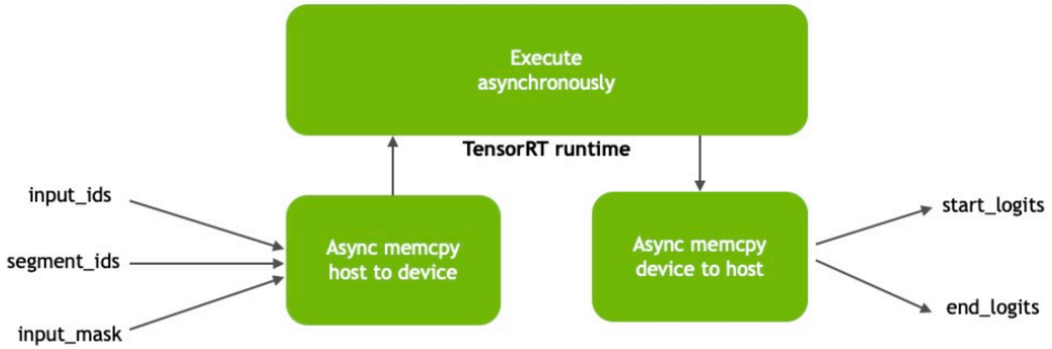
최신의 최적화 기법과 하드웨어 가속을 모두 적용한다면



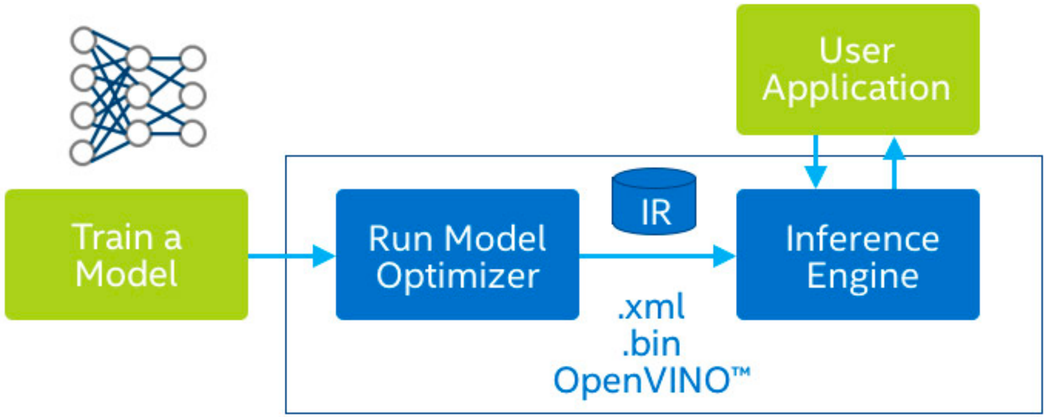
3.6 tensorRT

DEVIEW 2019

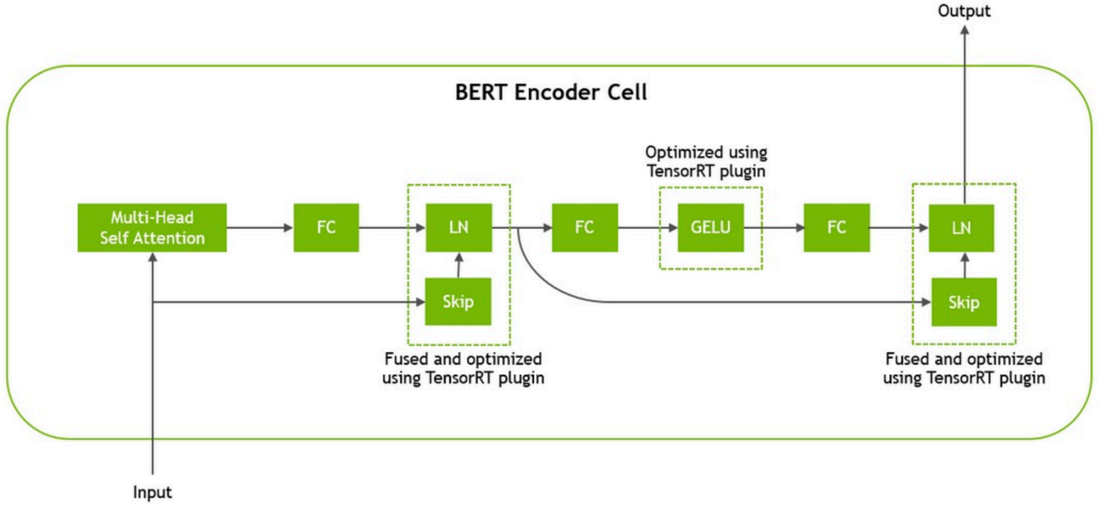
특정 환경 런타임을 기반으로한 추론 라이브러리등 존재



https://developer.nvidia.com/tensorrt



https://docs.openvinotoolkit.org/latest/_docs_IE_DG_Introduction.html



DEVIEW
2019

4.모델 정말 1만개를 서비스 했을까

4.1 auto ml + auto quantization

DEVIEW 2019

Auto ML에서 사용 가능한 재료들을 최대한 자동화

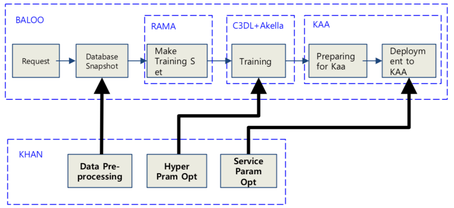
Khan

AutoML Project for Chatbot AI Builder

Overview

- End2End CB building pipeline
- Clean dataset
- Select appropriate features
- Select an appropriate models fam
- Optimize hyper-parameters
- Optimize service-parameters
- Automated model validation

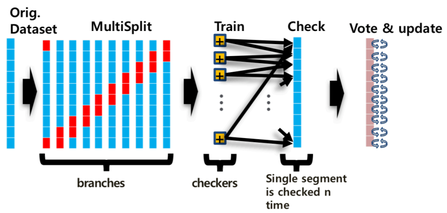
Chatbot Building Process



Automated Feature Eng.

- Intent and slot clustering
- Noisy query filtering
- Auto label error correction

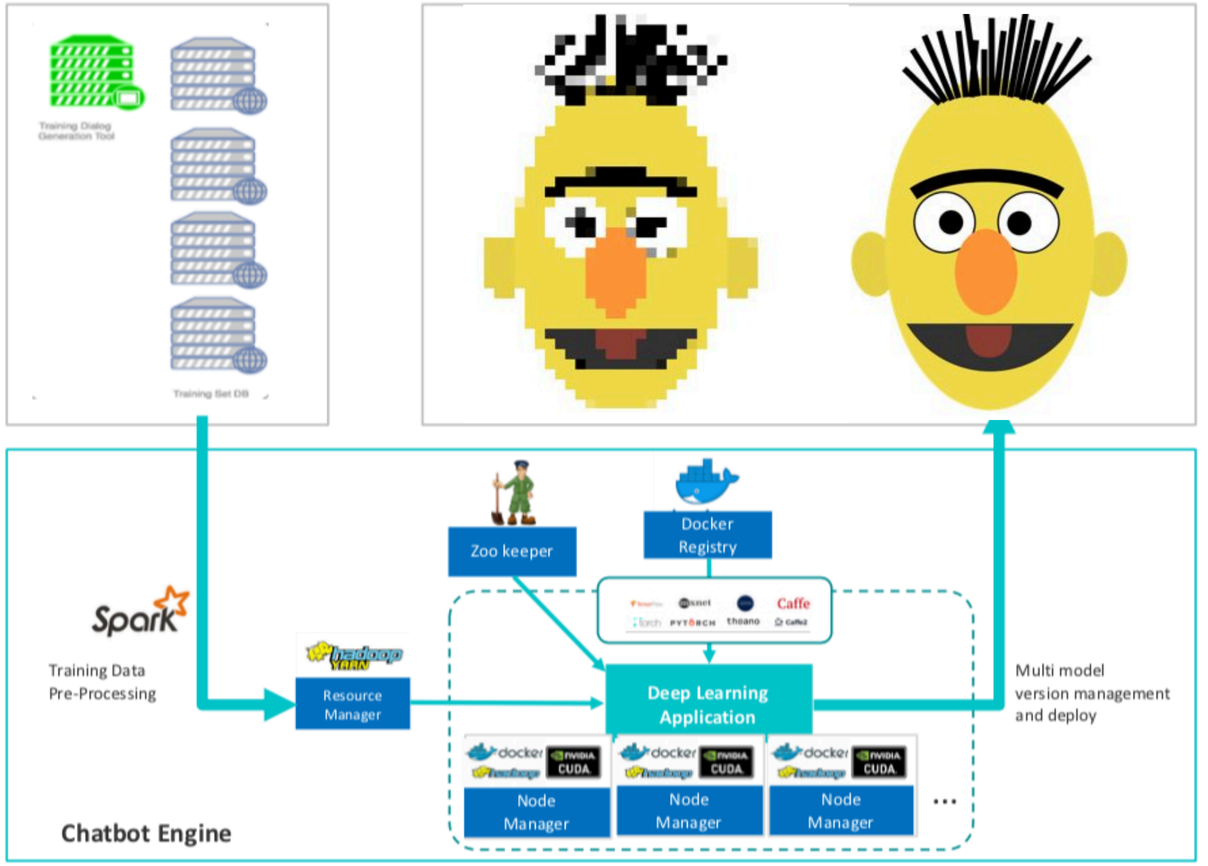
PICO: Auto Lable Correction



Hyper-Param Opt.

- Coordinate search
- Genetic algorithm
- Bayesian optimization

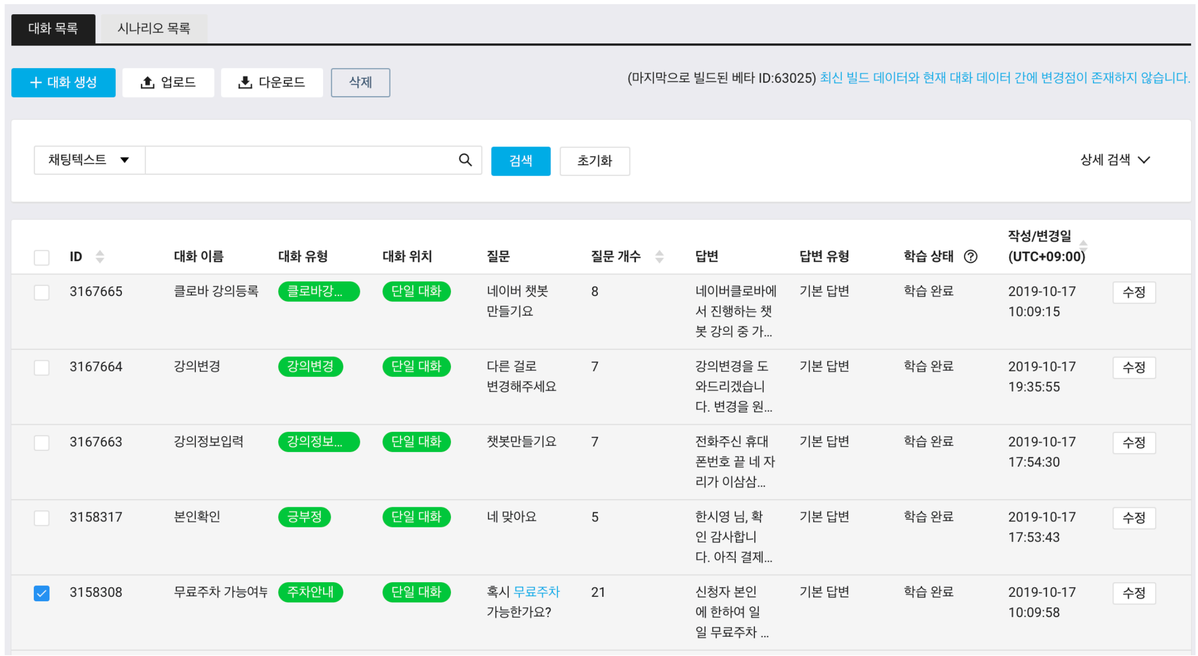
Coordinate Search for HiddenSizes

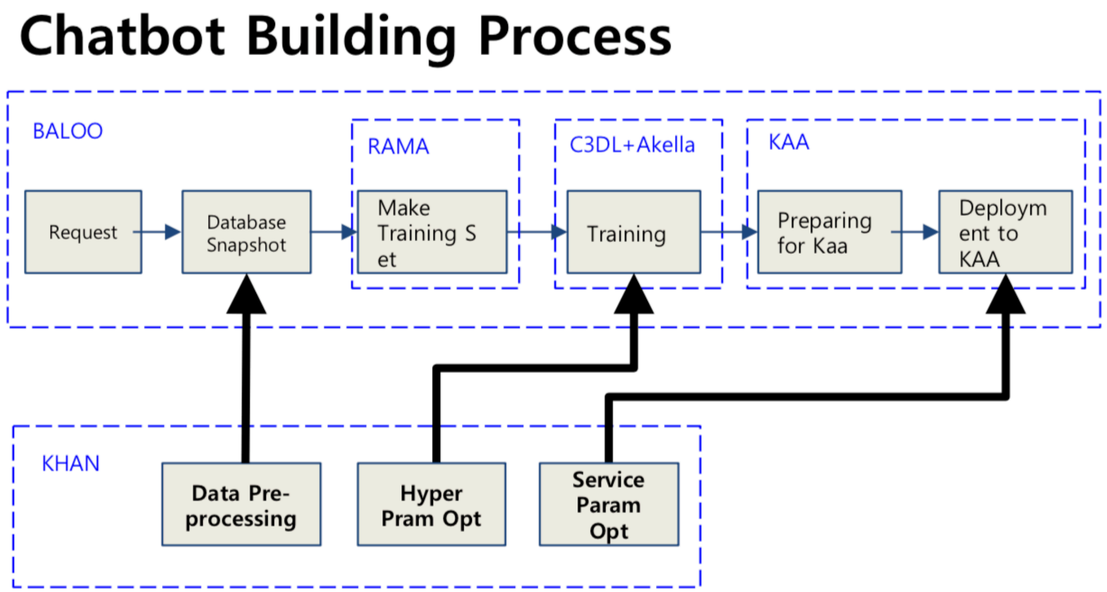


4.2 one source multi environment models

DEVIEW 2019

다양한 환경에서 (일반서버, GPU서버, 웨어러블, 가상, 오케스트레이션) 서빙 가능한 모델을 하나의 소스와 빌드 시스템으로 관리

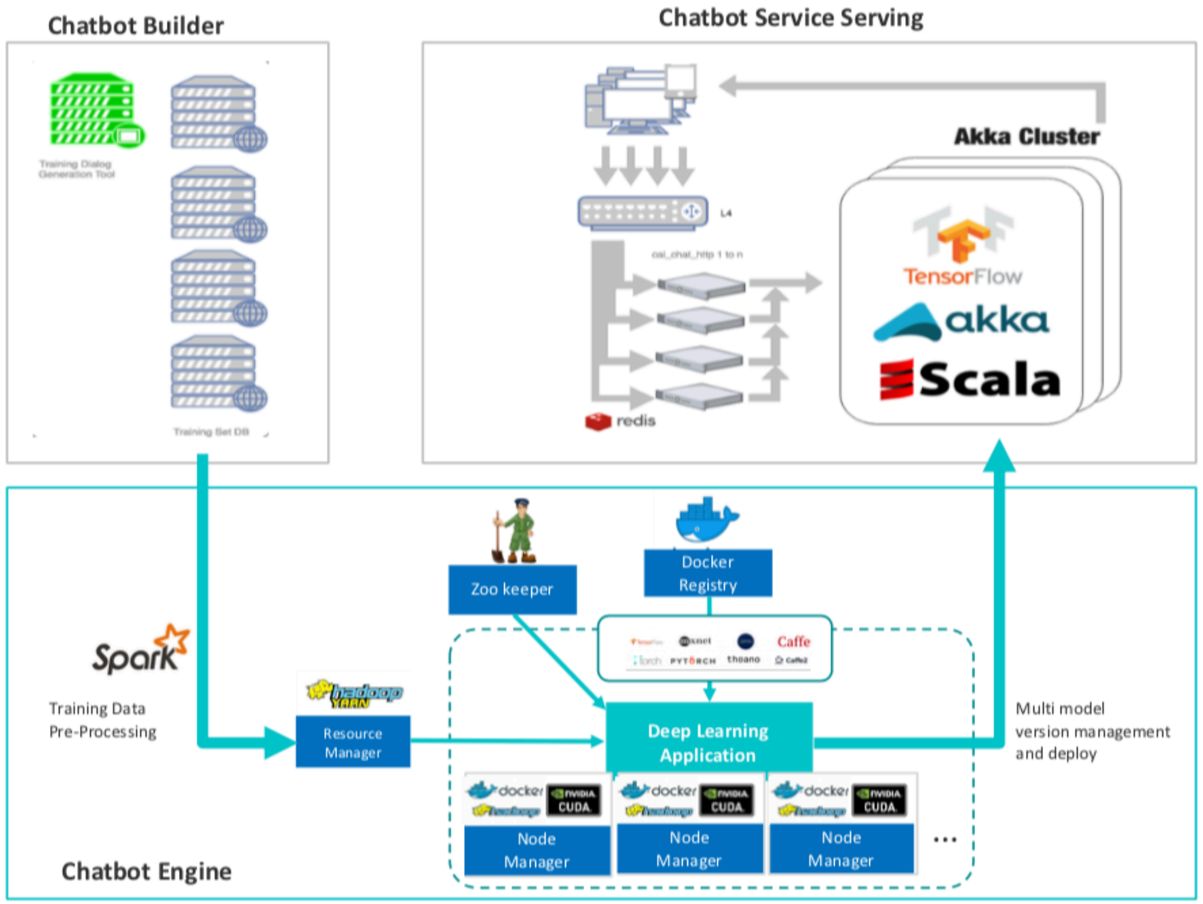




4.2 one source multi environment models

DEVIEW 2019

학습된 모델과 데이터를 프리즈된 버전별 스냅샷 통해 저장, 관리, 배포



신규 빌드 작업

현재 데이터를 기준으로 새로운 빌드를 추가할수 있습니다. 버튼을 누르면 빌드 됩니다.

호스트 지정 (기본) 빌드

빌드 현황

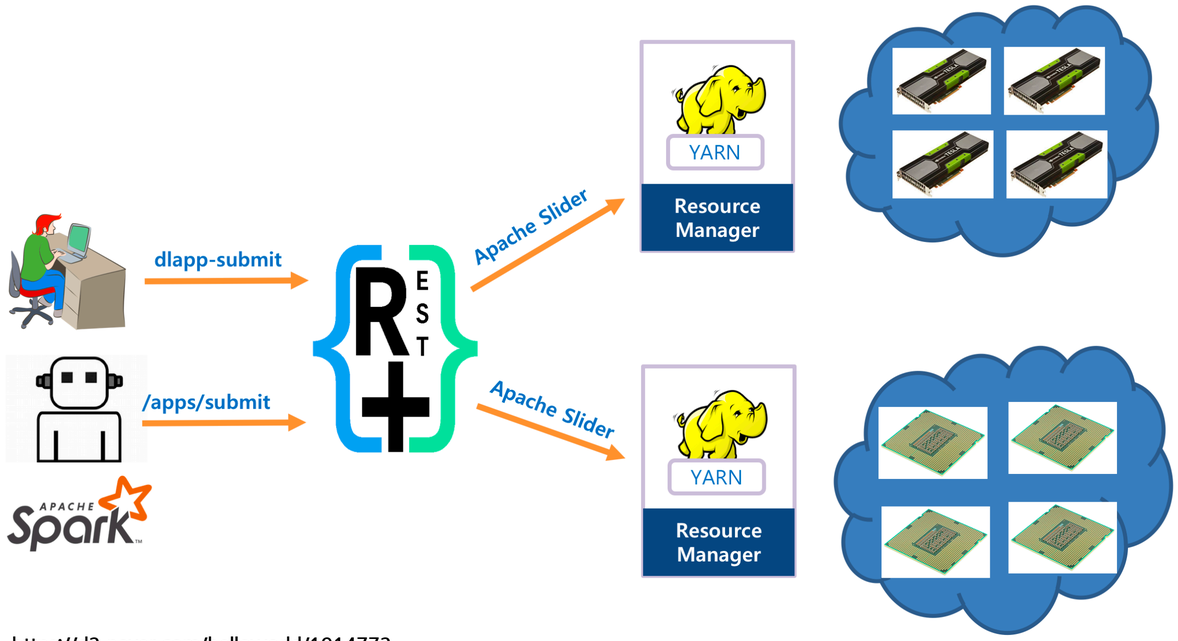
Copy CSV Excel PDF Print Show 10 entries

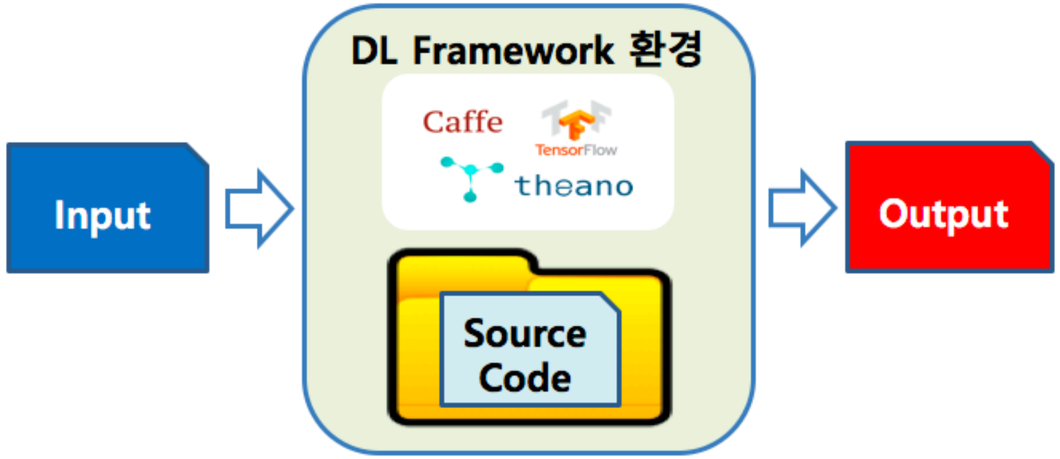
DT	리얼 배포	베타 배포	개발 배포	개발 상태
2019-10-17T13-58-31.798Z	다시배포	다시배포	배포하기	수정배포
2019-10-17T13-55-29.819Z	배포하기	배포하기	배포하기	수정배포
2019-10-17T10-50-14.332Z	배포하기	배포하기	다시배포	수정배포
2019-10-17T08-56-17.518Z	배포하기	배포하기	배포하기	수정배포
2019-10-17T07-54-41.410Z	배포하기	배포하기	배포하기	수정배포
2019-10-17T04-34-10.435Z	배포하기	배포하기	배포하기	수정배포

4.2 one source multi environment models

DEVIEW 2019

네이버 딥러닝 분산 플랫폼, C3DL 기반으로 동작



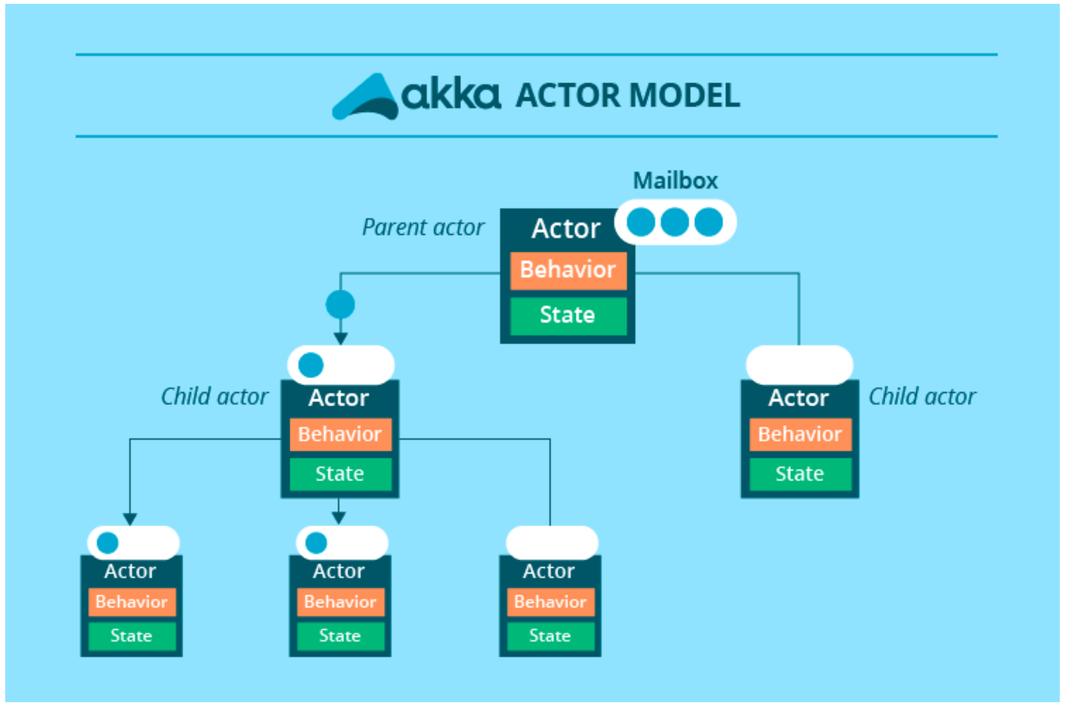


https://d2.naver.com/helloworld/1914772

4.3 decentralized clusters - (kaa)

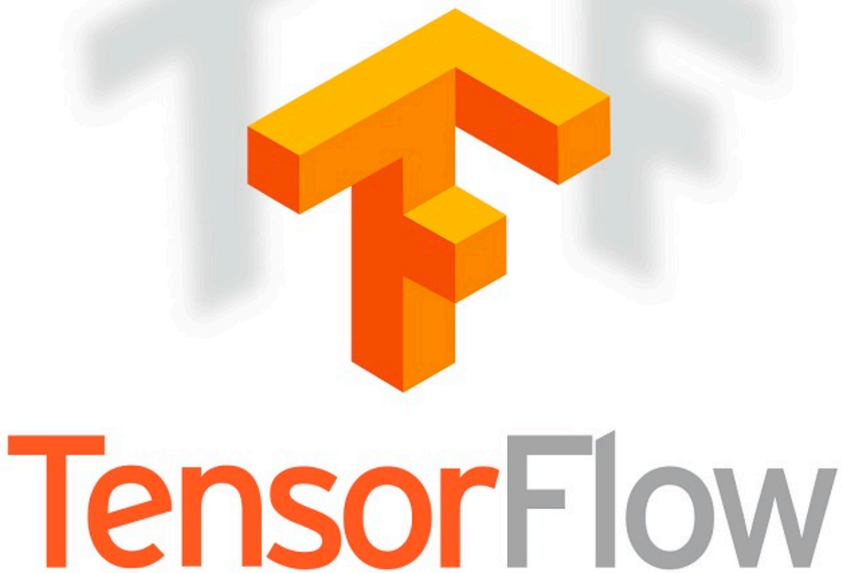
DEVIEW 2019

비 중앙화된 분산 클러스터를 이용한 머신러닝 모델 서빙 엔진을 설계



4.4 decentralized clusters - (kaa)

하나의 코드 베이스로 동작하는 데이터 파이프라이닝과 서빙 시스템

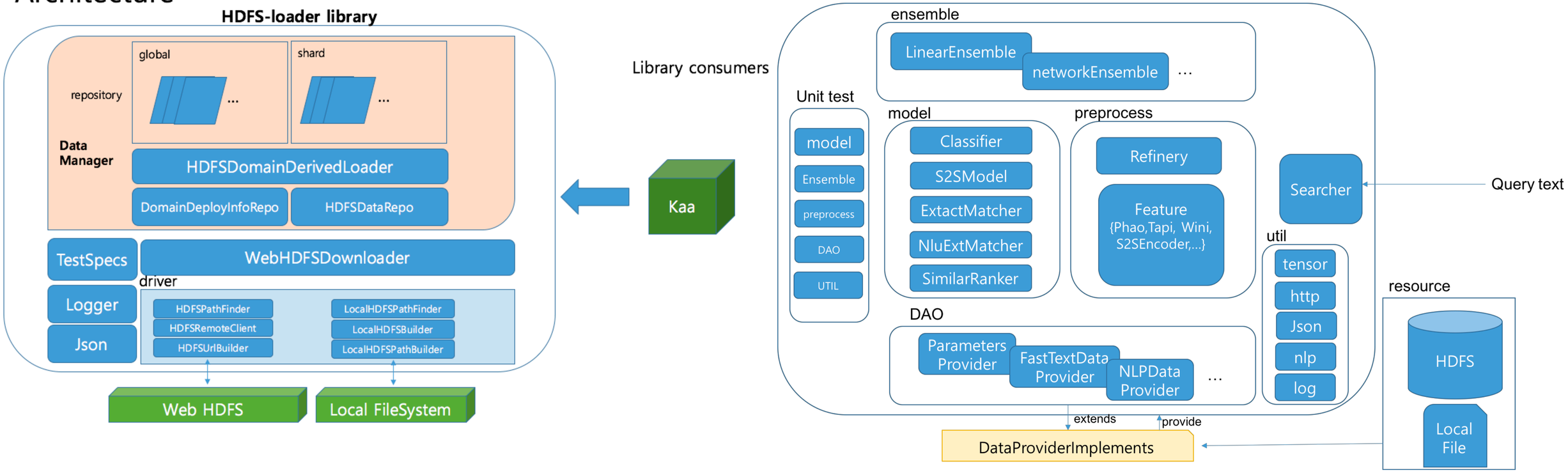


4.4 decentralized clusters - (kaa)

DEVIEW 2019

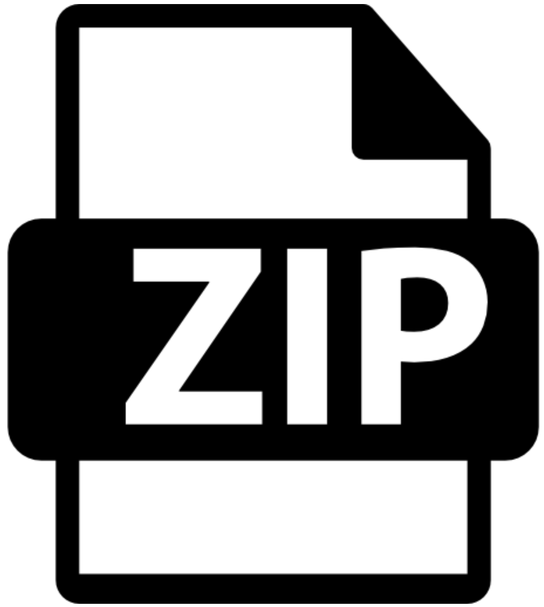
모델의 실험과 평가의 대한 공통 코드를 정적 모듈화 -> 신속한 모델 설계, 실험

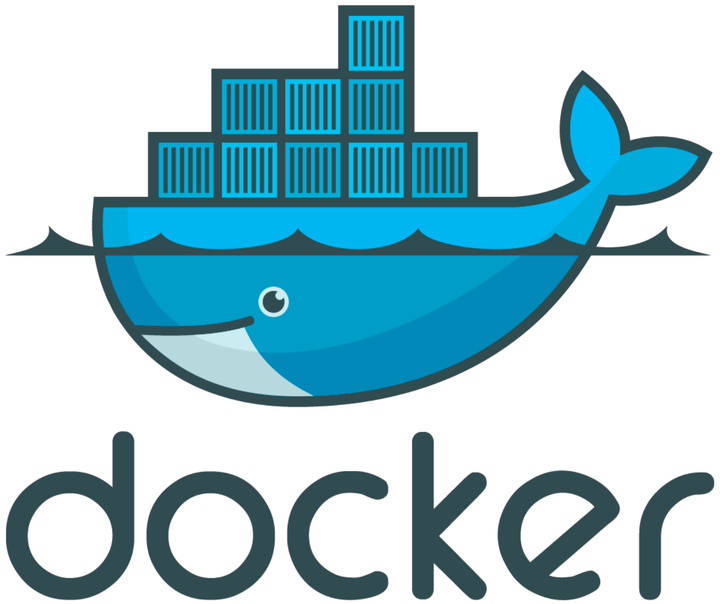
Architecture



4.5 decentralized clusters - (kaa)

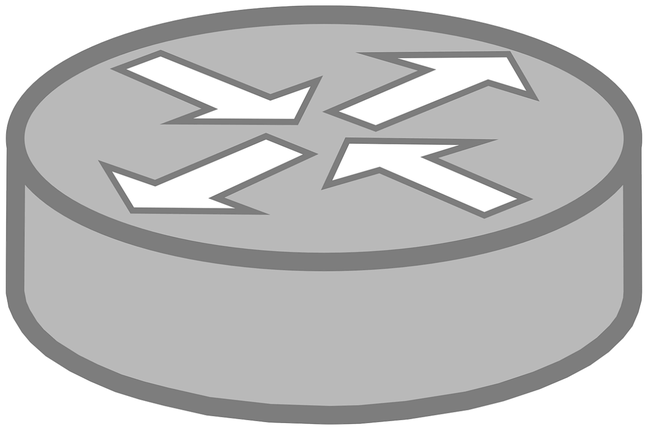
On-premise등 B2B를 위한 패키징을 시스템 초기 설계 수준에서 고려

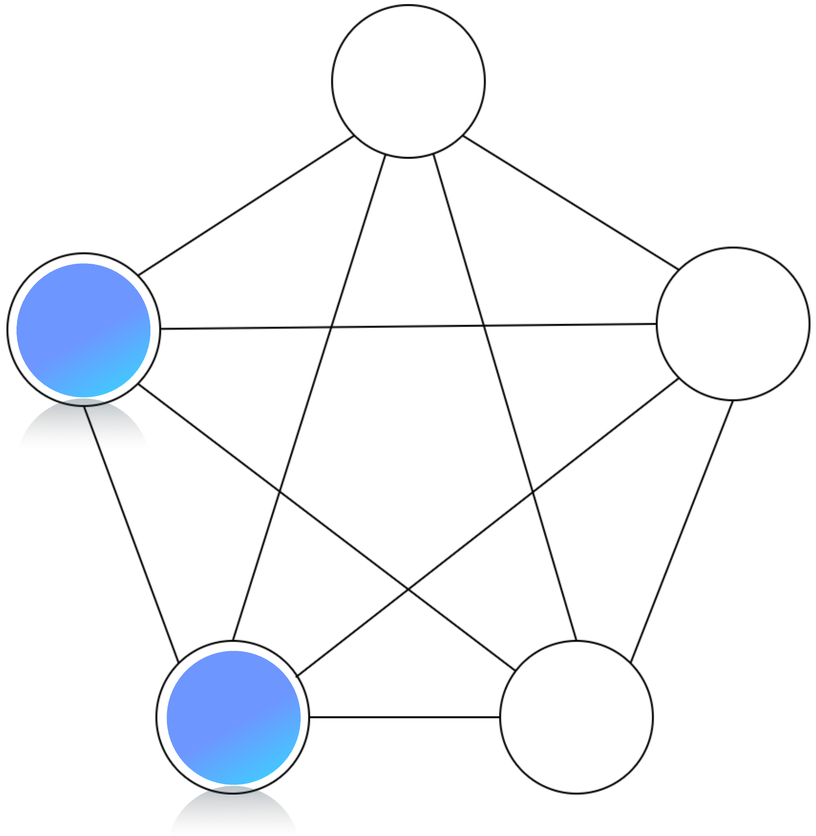




4.6 decentralized clusters - (kaa)

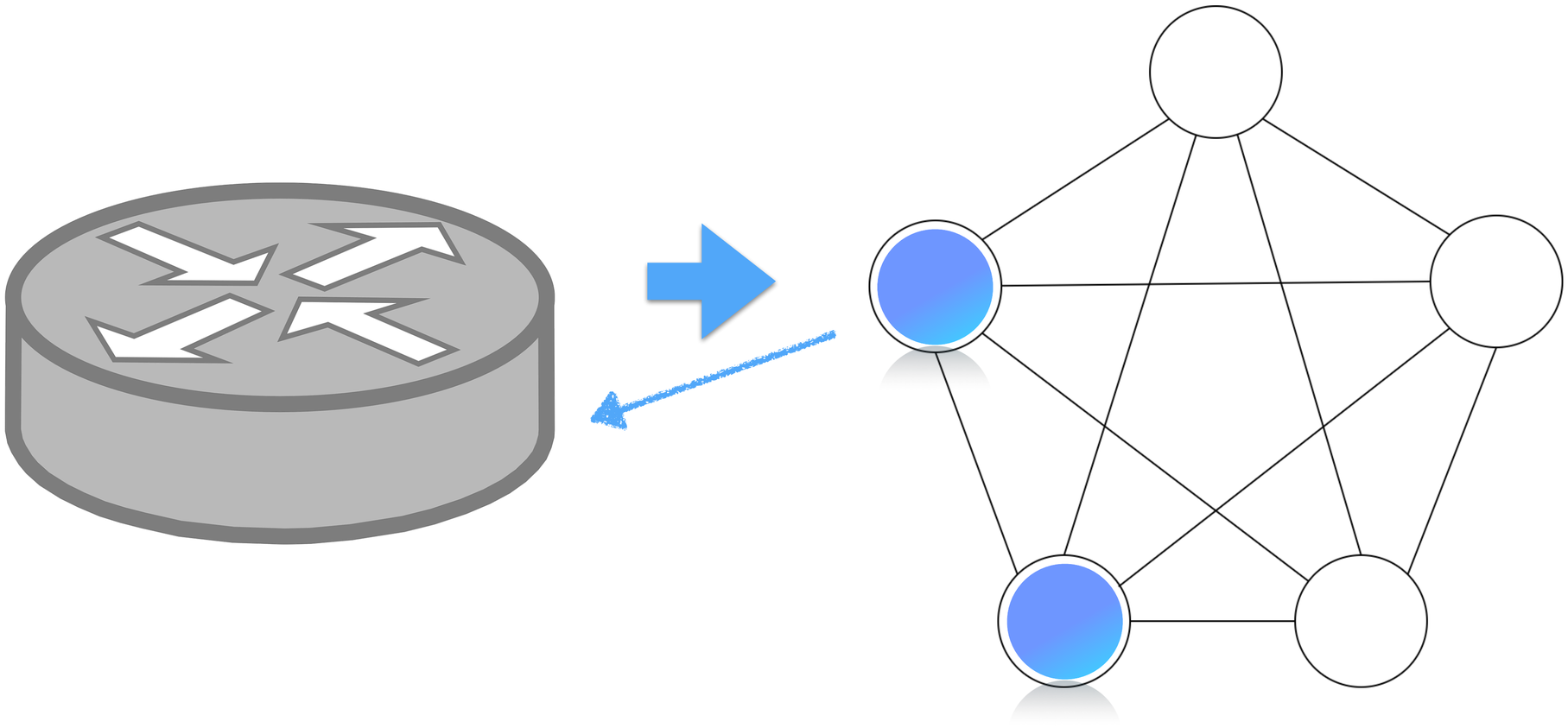
클러스터는 적어도 두벌 이상의 데이터가 존재





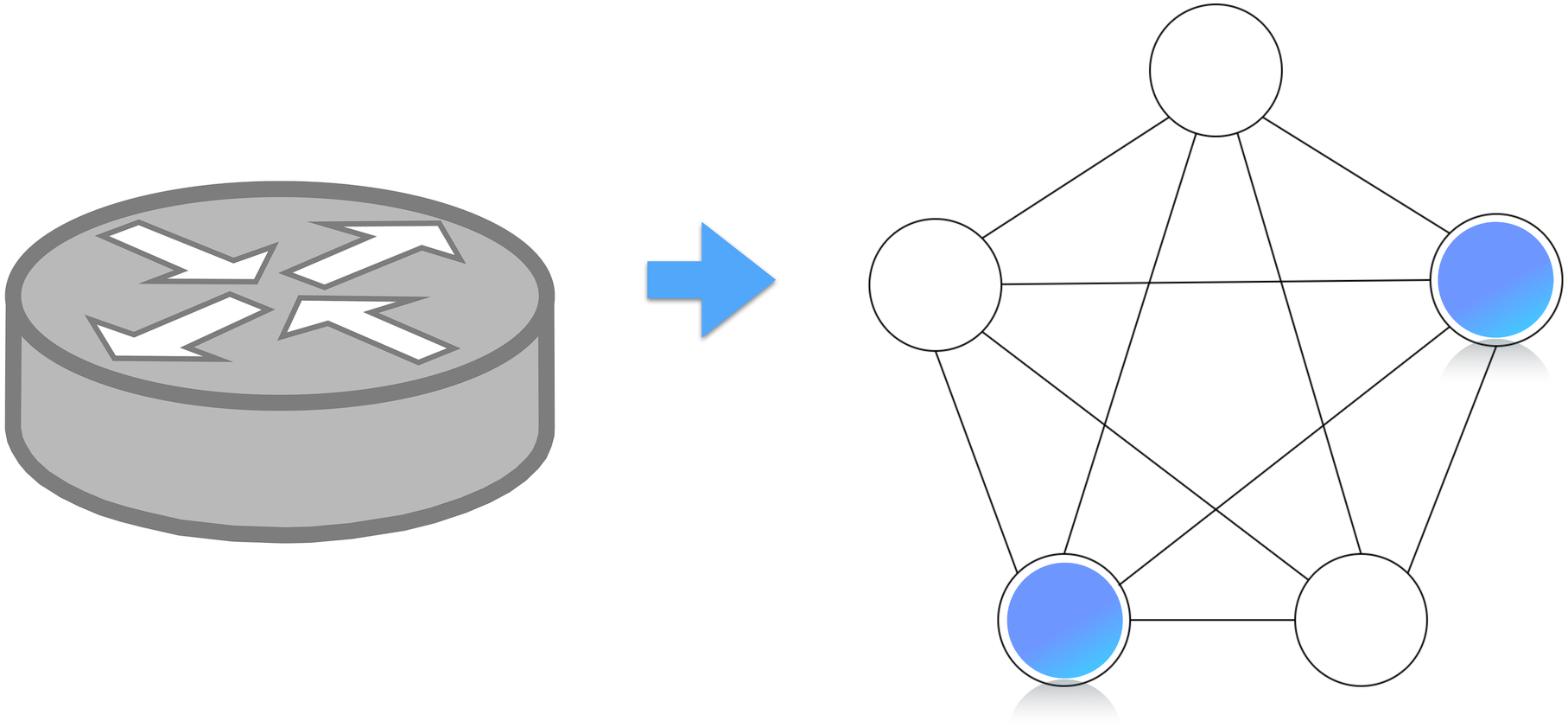
4.6 decentralized clusters - (kaa)

클러스터는 적어도 두벌 이상의 데이터가 존재



4.6 decentralized clusters - (kaa)

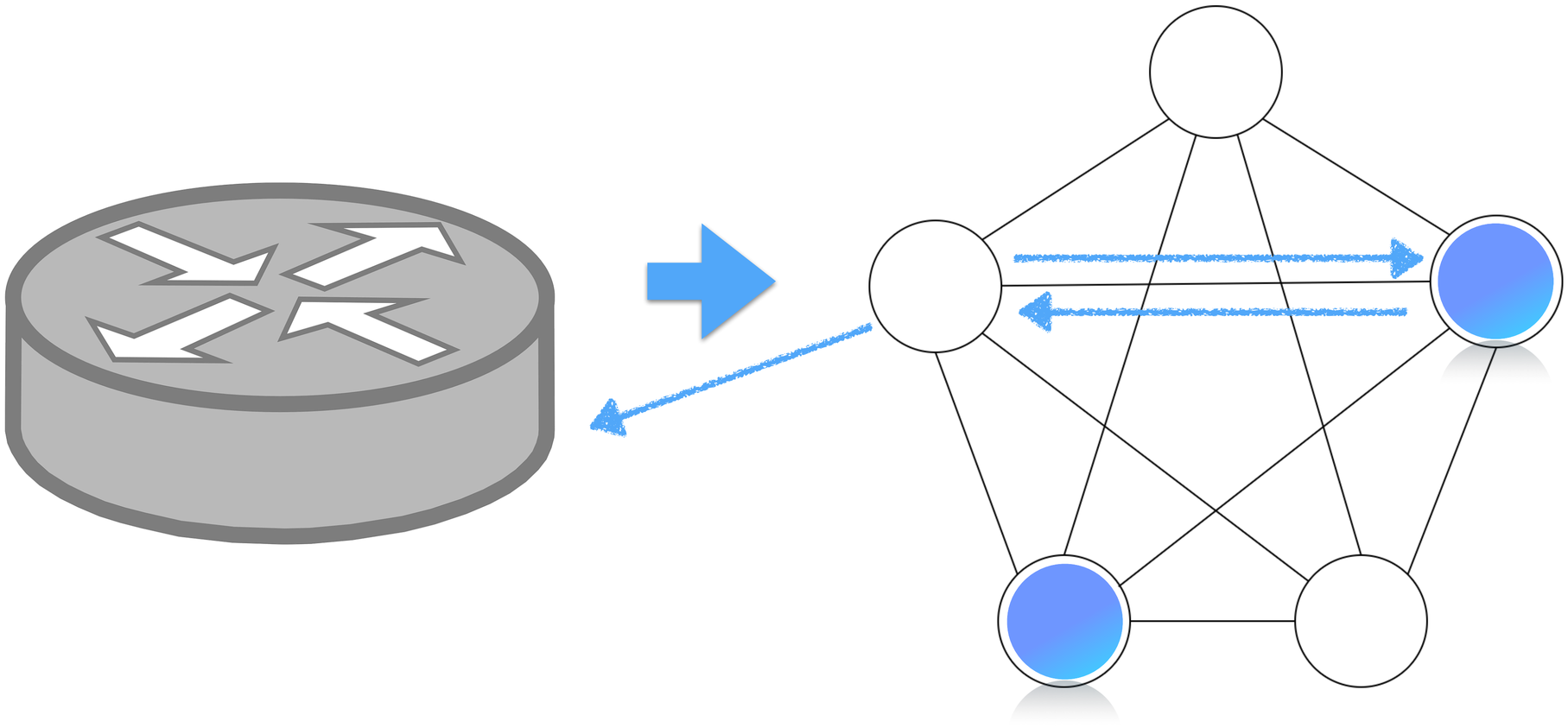
클러스터는 적어도 두벌 이상의 데이터가 존재



4.6 decentralized clusters - (kaa)

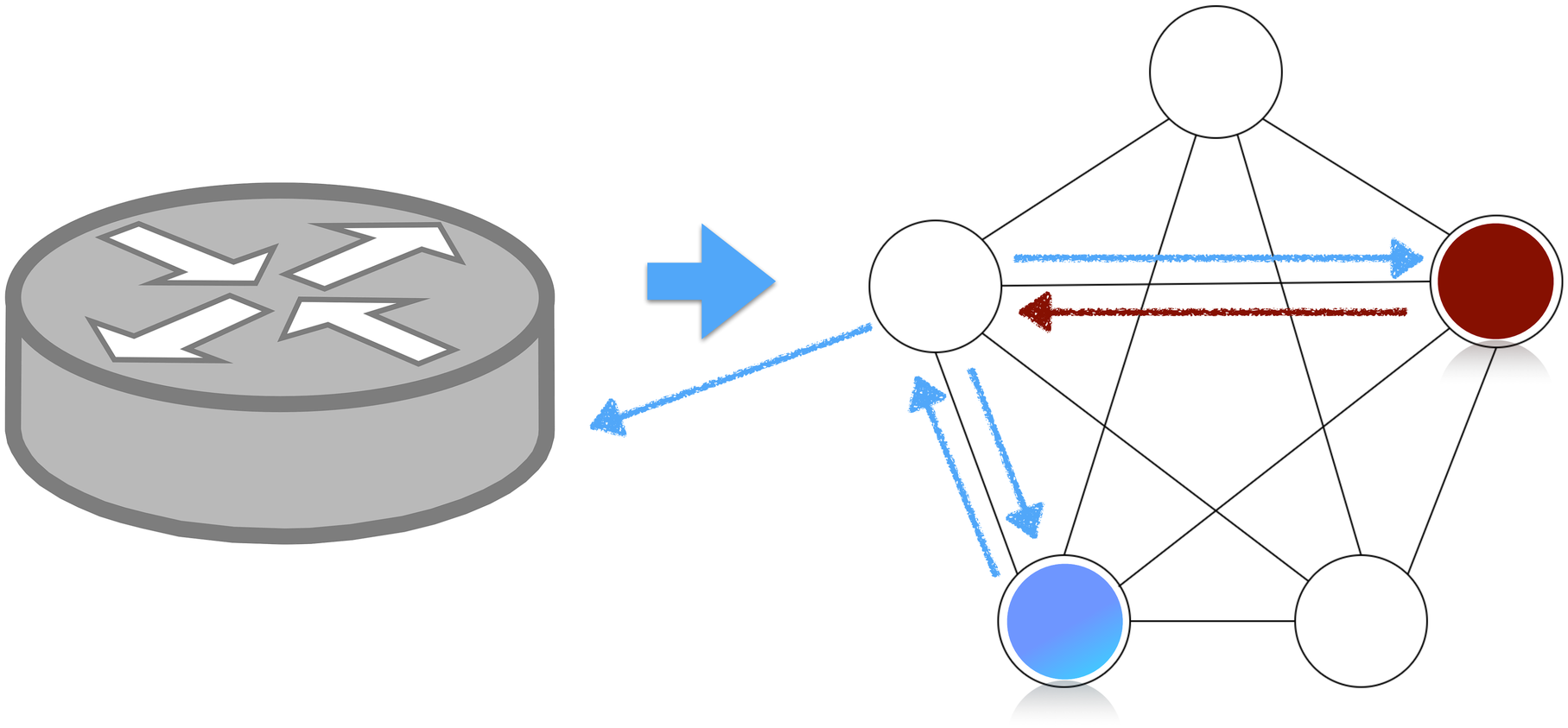
DEVIEW
2019

클러스터는 적어도 두벌 이상의 데이터가 존재



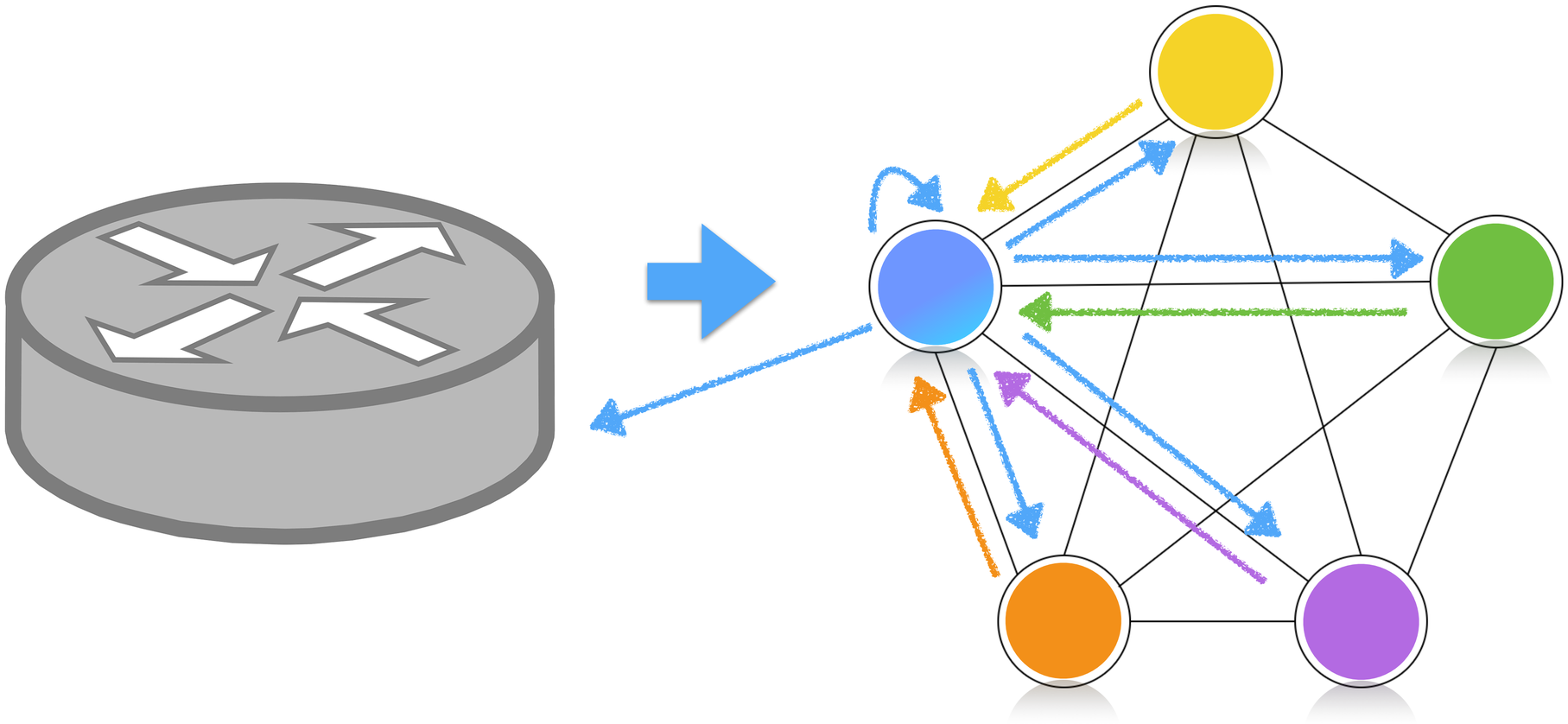
4.6 decentralized clusters - (kaa)

대상 노드가 죽거나 타임 아웃 경우 Dead Letter 처리



4.6 decentralized clusters - (kaa)

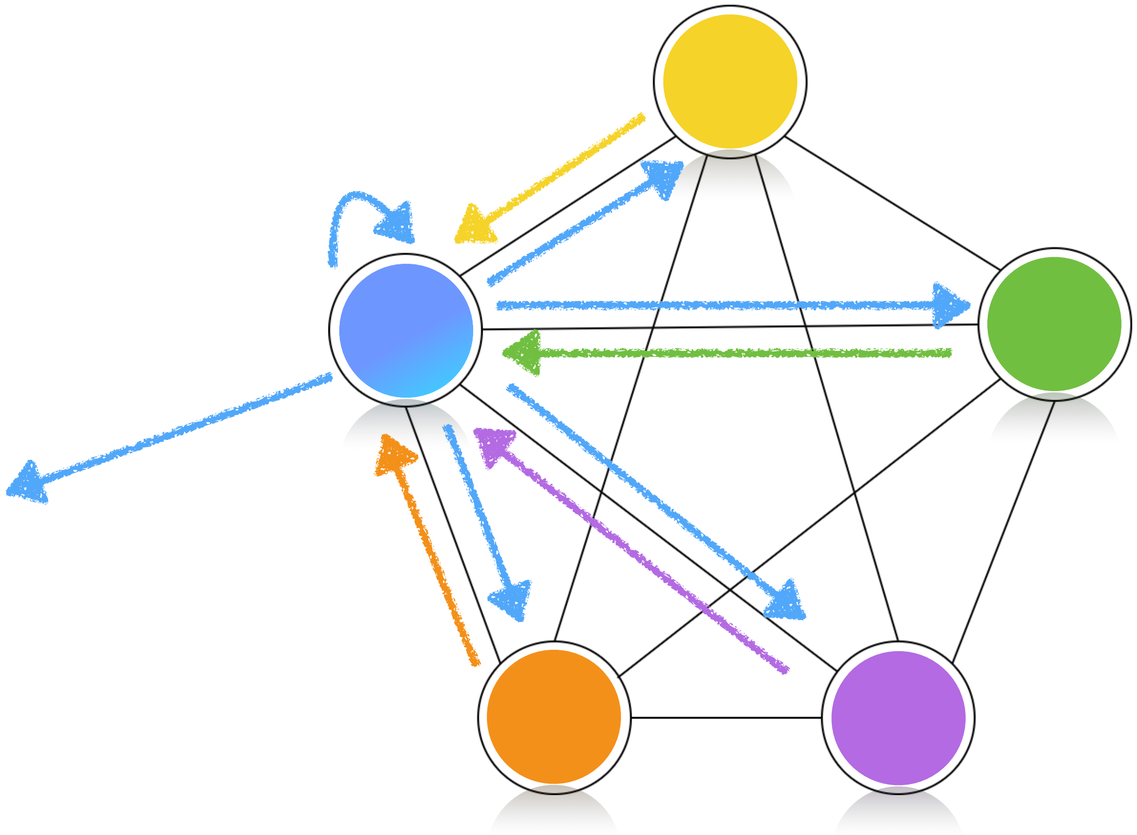
클러스터의 구성과 배포 옵션에 따라 성능, 안정성, 비용 고려



4.6 decentralized clusters - (kaa)

DEVIEW 2019

네이버의 검색을 지탱하는 대규모 인프라 시스템을 통한 모델 오프로딩

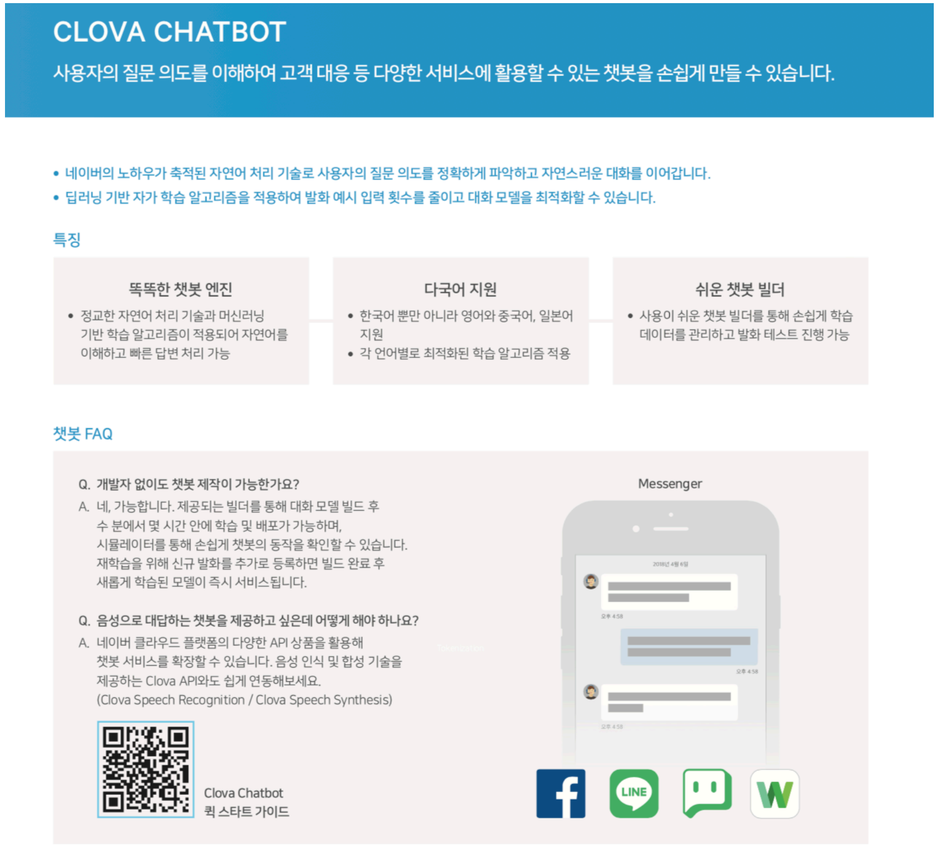


deview 2016 https://www.slideshare.net/deview/233-67606519

4.7 decentralized clusters - (kaa)

DEVIEW 2019

소수의 엔지니어로 15만개 이상의 모델을 학습하고 글로벌 환경에서 1만 여개의 모델을 서빙



powered by NAVER CLOUD PLATFORM

DEVIEW
2019

Q & A

DEVIEW
2019

Thank You