スケーラブルな分散ストリーム処理基盤の検討と構築

理学専攻・情報科学コース　加藤 香澄

1　はじめに

各種センサの普及とクラウドコンピューティング技術の発達により，一般家庭におけるライフログの収集と利用が可能となった．これらの技術は，お年寄りや子供を見守る安全サービスや防犯対策･セキュリティといった用途に応用されている．加えて，ディープラーニングが画像や音声の識別に広く用いられるようになった．ディープラーニングは，中間層が多層化されたニューラルネットワークを用いた機械学習手法であり，TensorFlow[1] や Chainer 等のディープラーニングライブラリが開発されている．しかし，ディープラーニングの処理の重さは課題となっている．センサデータを活用するサービスでは，一般家庭に解析用のサーバやストレージを設置することは困難である．一般に、これらのデータはセンサからクラウドに送られ，クラウドで解析処理される．しかし，動画像解析を利用するサービスでは，多くのセンサから大量のデータが継続してクラウドに送られることが想定され，その解析処理も非常に重く，リアルタイムに処理するのは困難である．

本研究では，複数センサから送られる大量の動画像データの解析を高効率に行うことを目的とし，スケーラブルな分散ストリーム処理基盤の構築手法を提案する．分散メッセージングシステムの Apache Kafka[2]（以降，Kafka と呼ぶ）と分散実行フレームワーク Ray[3] を用いた分散ストリーム処理基盤を構築し，その性能を示す．実験結果から，Kafka と Ray を用いた提案分散ストリーム処理基盤が高いスケーラビリティを有することを示す．

2　分散ストリーム処理基盤

本研究では，図 1 のような大規模ストリームデータ処理基盤を想定している．各家庭に設置されたセンサからクラウドに送信された動画像データは，ストリーム処理基盤によって収集され，分散処理基盤へ渡される．分散処理基盤がデータを受け取ると，データの解析処理がディープラーニングフレームワークを用いて行われ，結果がサービスマネジメントシステムを通してユーザに返される．動画像データの流量に応じて，スケーラブルに処理可能な分散処理基盤の構築を目指している．

ストリーム処理基盤として Kafka，分散処理基盤として Ray の採用を検討している．既発表研究 [4] から，Kafka クラスタにより大量のセンサデータに対するスケーラブルなメッセージングが可能であることを確認している．また，予備実験により一般的な分散処理基盤である Apache Spark[5] より Ray の方が高速かつスケーラブルに処理できることを確認している [6]．想定する処理基盤のプロトタイプ実装では，Kafka クラスタが各家庭のセンサから収集した画像データを Ray クラスタに送信し，クラスタ内の計算ノードで画像データの識別処理を行う．

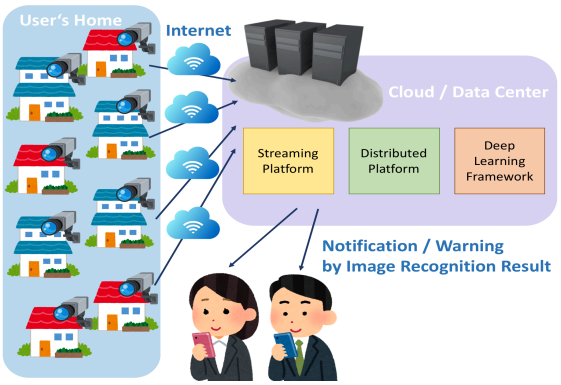

図 1: 想定する大規模分散ストリーム処理基盤

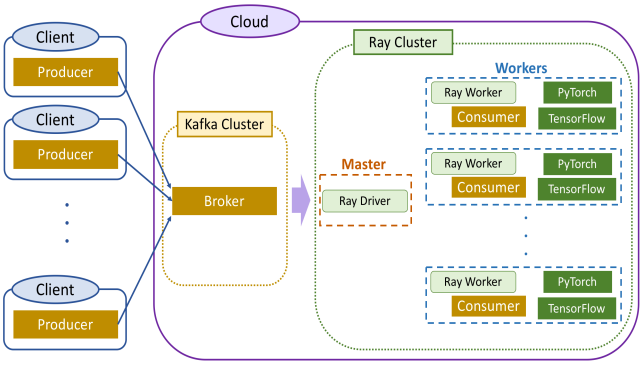

図 2: Ray と Kafka を用いた分散ストリーム処理基盤

3　実験

図 2 に Kafka と Ray を用いた提案分散ストリーム処理基盤を示す．Kafka Broker がクライアントである各家庭のセンサの Kafka Producer から画像データを受け取り，Ray クラスタのワーカに起動した Kafka Consumer に渡す．各 Consumer において画像データの識別処理が行われる．

図 3 に Kafka と Ray を用いた分散ストリーム処理の実験構成を示す．実験では，Producer を Broker と同じ計算ノードに配備し，以下の手順で処理する．(1) Kafka を起動し，Python プログラムを実行する．(2) 画像データが Producer から Broker に渡される．(3) データを受け取ると Broker は Ray クラスタのワーカノードに起動した Consumer にデータを送信する．(4) Consumer においてディープラーニングライブラリ PyTorch[7] と TensorFlow を用いた画像の識別処理が行われる．実験には，Kafka v. 1.1.0，Ray v. 0.5.3，PyTorch v. 0.4.1，TensorFlow v. 1.8.0 を用いた．

実験では，データセットとして ImageNet[8] を用いる．ImageNet の各画像サイズは約 26KB ほどである．表 1 に実験に用いる計算機の性能を示す．クラスタ内のマスタと全ワーカに用いるノードは同一の性能であり，各ノードは図 4 に示すように 1Gbps のネットワークで接続されている．また，実験では Kafka v. 1.1.0，Ray v. 0.5.3，PyTorch v. 0.4.1，TensorFlow v. 1.8.0

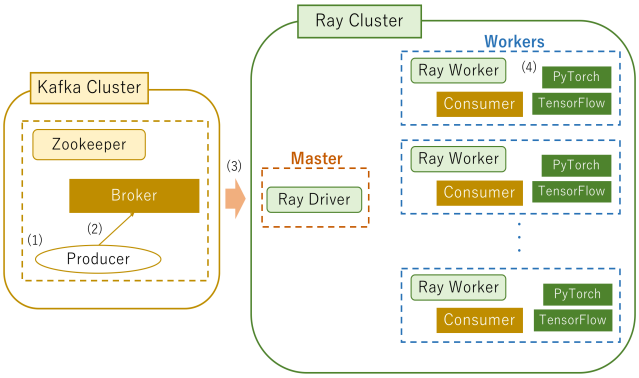

図 3: Kafka と Ray による分散ストリーム処理の実験構成

表 1: 実験に用いた計算機の性能

OS	Ubuntu 16.04LTS
CPU	Intel(R) Xeon(R) CPU W5590 @3.33 GHz 4 core×2 sockets(8 core)
GPU	NVIDIA GeForce GTX 980
Memory	48 Gbyte

を用いた．

Producer 数を 1 に固定し，Ray ワーカおよび Consumer 数を 1，2，4，8，12 と変化させると同時にバッチサイズを 1，5，10，20 と変化させスループットの計測を行った．バッチ化により一定量のデータの処理をまとめて行い，より高速に大量データを処理できることがある．計測結果を図 5 に示す．縦軸には 1 秒あたりの処理画像数をスループットで示している．図 5 によると，バッチサイズ 1-5 間で大幅なスループット向上が見られるが，10-20 間ではさほどスループットが向上していないことがわかる．また，全バッチサイズにおいてワーカ数増加によりスループットが向上している．以上の結果から，Kafka と Ray を用いたスケーラブルな分散ストリーム処理基盤が構築できることが示せた．

4　まとめと今後の課題

Kafka と Ray を用いた分散ストリーム処理基盤を構築しその性能を調査した．実験から，提案した分散ストリーム処理基盤が高いスケーラビリティを有することを示した．今後の課題として，実際のユースケースを想定した動画像データ解析への適応や，別環境での実験を行い提案分散ストリーム処理基盤の評価を行うことを検討している．

謝辞

この成果の一部は，JSPS 科研費 JP16K00177，平成 30 年度国立情報学研究所公募型共同研究，国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務及び JST CREST JPMJCR1503 の支援により得られたものです．

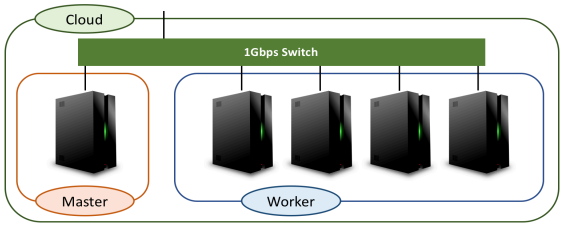

図 4: 実験環境

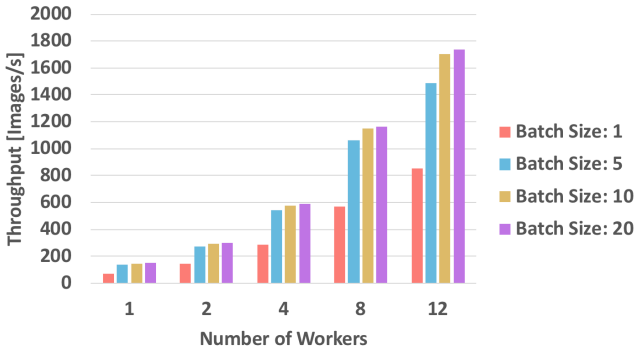

図 5: Kafka と Ray を用いた分散ストリーム処理基盤におけるバッチサイズに基づくスループット変化

参考文献

[1] M. Abadi, et al.,“TensorFlow: Large-scale machine learning on heterogeneous systems,”2015, http://download.tensorflow.org/paper/whitepaper2015.pdf. pp.1-19. [Online]. http://tensorflow.org/

[2] “ Apache kafka, ” https://kafka.apache.org/

[3] P. Moritz, et al., “ Ray: A Distributed Framework for Emerging AI Applications, ” 2017. http://ray.readthedocs.io/en/latest/index.html

[4] 一瀬絢衣ほか, “ Kafka を利用したリアルタイム動画像解析フレームワークのレプリケーションによる性能変化の調査, ” 第 10 回データ工学と情報マネジメントに関するフォーラム (DEIM2018), I3-3, 2018.

[5] “ Apache Spark, ” https://spark.apache.org/.

[6] K. Kato, et al., “ Construction Scheme of a Scalable Distributed Stream Processing Infrastructure Using Ray and Apache Kafka, ” 34th ISCA International Conference on Computers and Their Applications (CATA 2019) (To appear).

[7] A. Paszke, et al.,“Pytorh: Tensors and dynamic neural networks in python with strong gpu acceleration, ” 2017. https://pytorch.org/

[8] J. Deng, et al., “ ImageNet: A large-scale hierarchical image database. ” IEEE Conference on Computer Vision and Pattern Recognition, 2009. http://www.image-net.org/