

~~めもおきば~~ TechReport 2019.07 新刊落としましたペーパー

2015年12月に同人活動初めてから、全ての参加イベントで1冊ずつ出してきたのですが、今回ついに新刊を落としました。夏コミか技術書典できちんと出します。

FaaSの次はなんなんだ(<i>a</i> 版).....	2
あとがき	4



Aki @ nekoruri

FaaS の次はなんなんだ（α版）

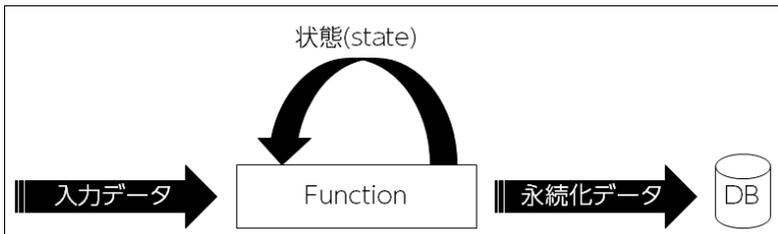
2014年に登場したAWS Lambda——あるいは2008年のGoogle App Engine——にはじまるFunction-as-a-Service(FaaS)は、「サーバーレス」の中心人物として、急速に普及しつつあります。

FaaSはスケーラビリティを確保するため、The Twelve-Factor Appでも「VI. プロセス」で謳われているように、そこで動かすコードに対してステートレスかつシェアードナッシングであることを制約として課しています。そのため、リクエストをまたぐ全てのデータは外部のデータストアにて永続化する必要があります。このステートレスという制約は、実装の容易さはさておき、とても分かりやすいものです。

その一方で、GraphQLなどによるCQRSアーキテクチャの採用や、IoTによる大量データ収集などを背景に、非同期なイベント駆動を効率よく捌くことも求められています。FaaSではイベント(あるいはそれを複数まとめたブロック)ごとにステートレスな実装が求められ、データストアへのアクセスが高頻度になるなどの課題が顕在化しています。

そのような背景の中で、シャーディングを利用したリアルタイムデータ処理がビッグデータ解析の分野以外にも注目を集めつつあります。

FaaSでは一つのリクエストをまたぐデータを全てステートレスとして排除しますが、性質の違いに気をつけると、長く永続化されるべきデータと、一時的な計算中データとしての状態(state)にわけることができます。

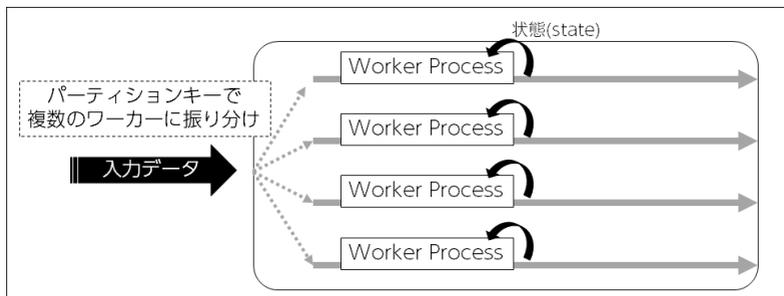


そもそも永続化すべきデータはもともとDBに保存すべきであり、FaaSが排除しているのは、まさにこの「状態」というわけです。その一方で、この状態を外部のデータスト

アに依存する必要があるため、今度はそのデータストア側がボトルネックとなってしまいます。

リアルタイムデータ処理基盤として利用されているストリーム処理エンジンでは、データベースでは一般的になりつつあるシャーディング(水平分割)の技術を用いることで、この「状態」をプロセス上のメモリに維持できます。

ストリーム処理エンジンのシャーディングでは、イベント(メッセージ)毎に Partition Key(分割キー)と呼ばれる値を付与します。この Partition Key に基づいて同じ値のイベントが、同じワーカープロセスで動作するように振り分けられます。



これにより、たとえば株価の変動データがリアルタイムで入力されるようなシステムで銘柄コードをパーティションキーとすることにより、ある銘柄コードの変動データは同じプロセスに振り分けられるため、メモリ上に一時保存した直近の過去データを利用して1時間の移動平均などを容易に求めることができます。

もちろんこれは非常に素直な考え方で、実際にはワーカープロセスが「落ちる」「遅れる」「間違える」場合など、分散システムならではの課題も発生します。ですがストリーム処理エンジンではそういった課題を解決するための仕組みとプログラミングモデルを提供してくれます。

というわけで、FaaSの次はこういったストリーム処理エンジンのプログラミングモデルの活用が進んでいくのではないかと考えています。

あとがき

なぜ最近ストリーム処理に注目しているかをごがつり書きたかったのですが、今回まったく筆が進まず、新刊としては実質落とすこととなりました。夏コミか技術書典であらためてきちんとまとめて出します。

めもおきば TechReport 2019.07

新刊落としましたペーパー

発行日 2019年7月27日 技術書同人誌博覧会

著者 Aki@nekoruri
aki@nekoruri.jp

発行 めもおきば
<https://d.nekoruri.jp/>