

FASTER

# FASTERの話

Build 2021 (たぶん)最速Recap Day

Functionsのストレージプロバイダーとして突如現れた、Netheriteで使われているFASTERとはなにか？（触りだけ）

by Takekazu Omi(@Baleen.Studio)

2021/05/29 v1.0.0



BALEEN STUDIO  
Takekazu Omi @Baleen.Studio



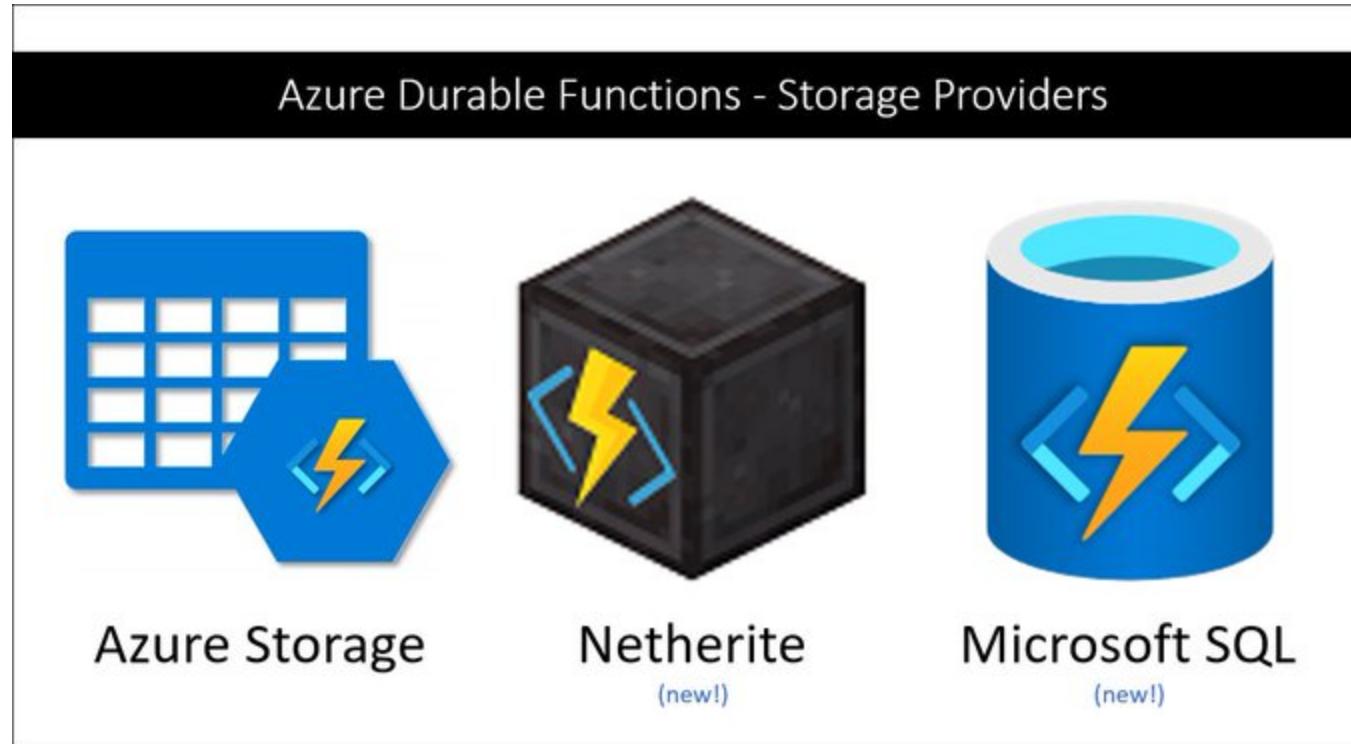
# 自己紹介

近江 武一 [@takekazuomi](#)

- 所属 [jazug](#)、[baleen.studio](#)（仲間を募集中）
- [GitHub](#)、Azure関連のPull request、気がついたら直す方向で
  - Private AKS Cluster の `.bicep` を [PR](#)
  - SQL Elastic Pool のTemplate が動かなくなってたので [PR](#)
- Blog [kyrt.in](#)、[zenn.dev](#)を使い始めた
  - [ARM template DSL、Bicep を使おう\(1\)](#)
  - [ARM template DSL、Bicep を使おう\(2\)](#)

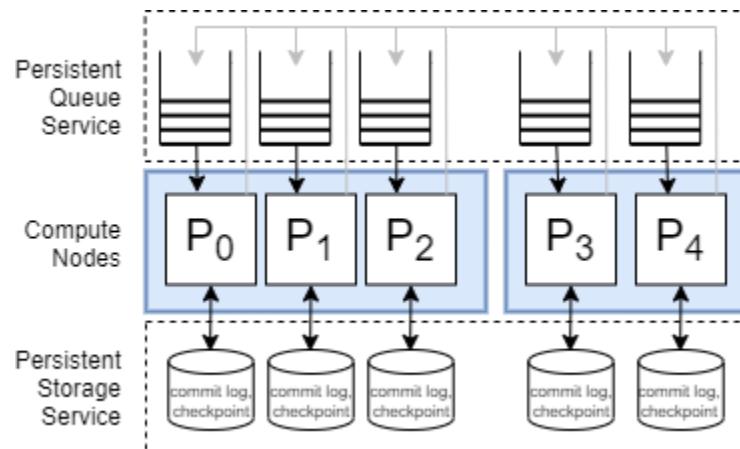
# 今日の話は、～Netherite～

New Storage Providers for Azure Durable Functions、参照



# Netherite

Netherite（ネザライト）は、マイクラのダイヤモンドより硬い超合金(Z?)。DFでは、メッセージを Azure Storage Queueに格納し、インスタンスの状態をAzure Storage Tableに格納している。Netheriteでは、EventHubs + Faster + Azure Page Blobs で実装



# FASTER

今日、話をしたかったのは、[FASTER](#)の部分

短く言うと、永続化レイヤー切り替えられて、ラッチフリーな key-value store. FASTER は、DFのAzure Tableの代わりをやってる。

[fasterlog-basics](#)

- FASTER Log
- FASTER KV

# FASTER Log

- C#製の**high-performance concurrent persistent recoverable log**、ランダムリーダーのライブラリ。
- 低遅延で非常に頻繁なコミット操作をサポートし、ディスク帯域幅をすぐに飽和させるぐらい高性能。
- 同期と非同期の両方のインターフェイスをサポート。ディスクエラーを処理の、チェックサムをサポート

# FASTER KV

- concurrent key-value store + cache (available in C# and C++)
- point lookups と heavy updates 向けにデザイン
- 高速の外部ストレージ (ローカルまたはクラウド) を活用して、メモリよりも大きなデータをサポート。
- パフォーマンスとコミット待機時間をチューニングできるチェックポイント技術を使用した一貫したリカバリモデルをサポート。

# 注目点

1. プロセス内にホストして使えるようなライブラリ
2. ちゃんとしたリカバブルなLog実装
3. Key Vaule 部分は、ラッチフリー
4. Page Blobを永続化先として選択化。IDevice

# まとめ

FASTER は、かなりちゃんと作られた、KV Storeのストレージエンジン、Page Blobを有効に使った実例も素晴らしい。ちょっと触った感じだとかなり素晴らしい。いろいろ使い道があるだろう。

- 今回のコンテンツ、
  - GitHub [20210529-faster](#)
  - slide share [20210529-faster](#)
- Powerd by [Marp](#)。ありがとうございました。

FASTER

終

