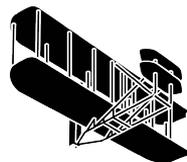


# リリースラッシュを支えた Cloud Spanner 事例

吉田 貴勝

株式会社WFS シニアマネージャー



WRIGHT FLYER  
STUDIOS

# スピーカー自己紹介



吉田 貴勝  
株式会社WFS  
シニアマネージャー

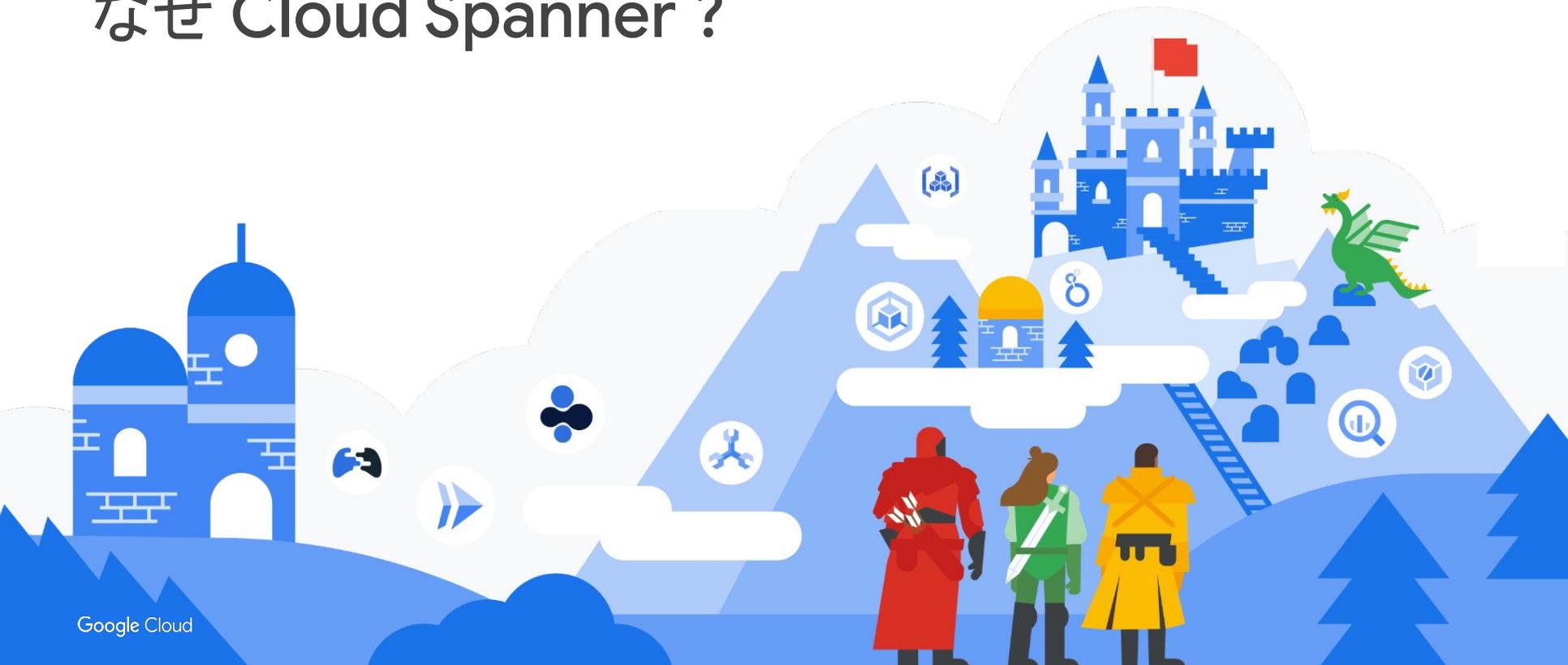
株式会社WFS でサーバエンジニアをやっています。  
好きなものは猫とビール。

WFS はスマートフォン向けゲームの開発運用をしています。  
直近1年で「ヘブンバーンズレッド」をはじめ、4つのゲームタイトル  
をリリースさせていただきました。

このリリースラッシュを支えた、特に Cloud Spanner について弊  
社で得られた知見をお話していきたいと思えます。



# なぜ Cloud Spanner ?



# モバイルゲームの特性

モバイルゲームというサービスは、トラフィック変化が非常に大きい

ゲームリリース時やゲーム内のイベント開始時などアクセスが集中する

1日の中だけ見ても、変化が大きい



# これまでは MySQL

WFS では主に MySQL を利用してきた

トラフィックに対して柔軟なスペックのコントロールが難しい

シャーディングなどの負荷分散を頑張ってきた

カラム追加やスペック調整はゲームをメンテナンスに入れて対応



# Cloud Spanner の特性 (引用)

無制限のスケーリング、強整合性、最大 99.999% の可用性を備えたフルマネージド リレーショナル データベースです。

- 無制限のスケーリングによって、リレーショナル セマンティクスと SQL のすべてのメリットを享受
- 任意のサイズで開始し、ニーズの拡大に応じて制限なしでスケーリング
- 計画的ダウンタイムのない、オンラインでのスキーマ変更で高可用性を実現
- リージョンや大陸全体にわたる強整合性で高性能のトランザクションを提供
- 自動シャーディングなどの機能により手動のタスクを排除し、イノベーションに注力

Cloud Spanner

<https://cloud.google.com/spanner?hl=ja#all-features>



# Cloud Spanner による解決

トラフィックに対して柔軟なスペックのコントロールが難しい

=> ノード数(性能)をダウンタイムなしで変更可能

シャーディングなどの負荷分散を頑張ってきた

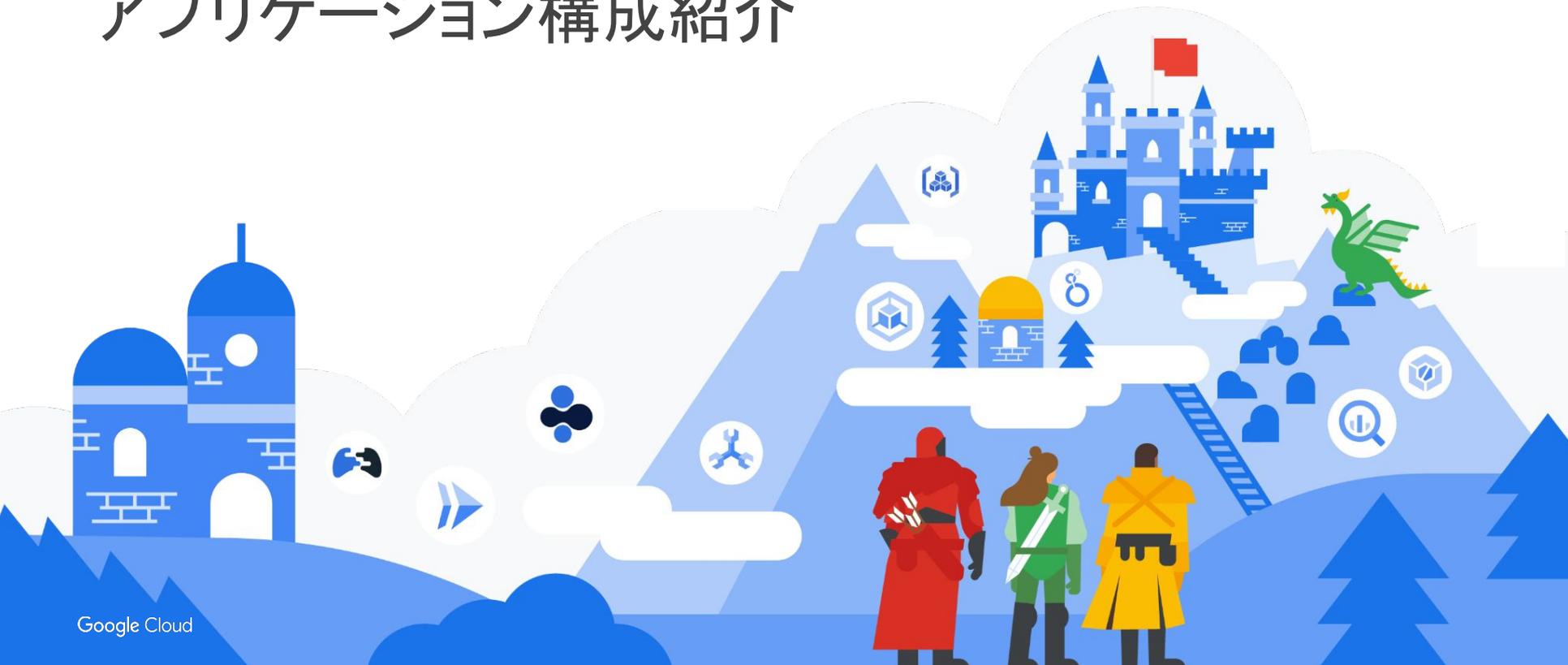
=> ワークロードに合わせて自動でシャーディング

カラム追加やスペック調整はゲームをメンテナンスに入れて対応

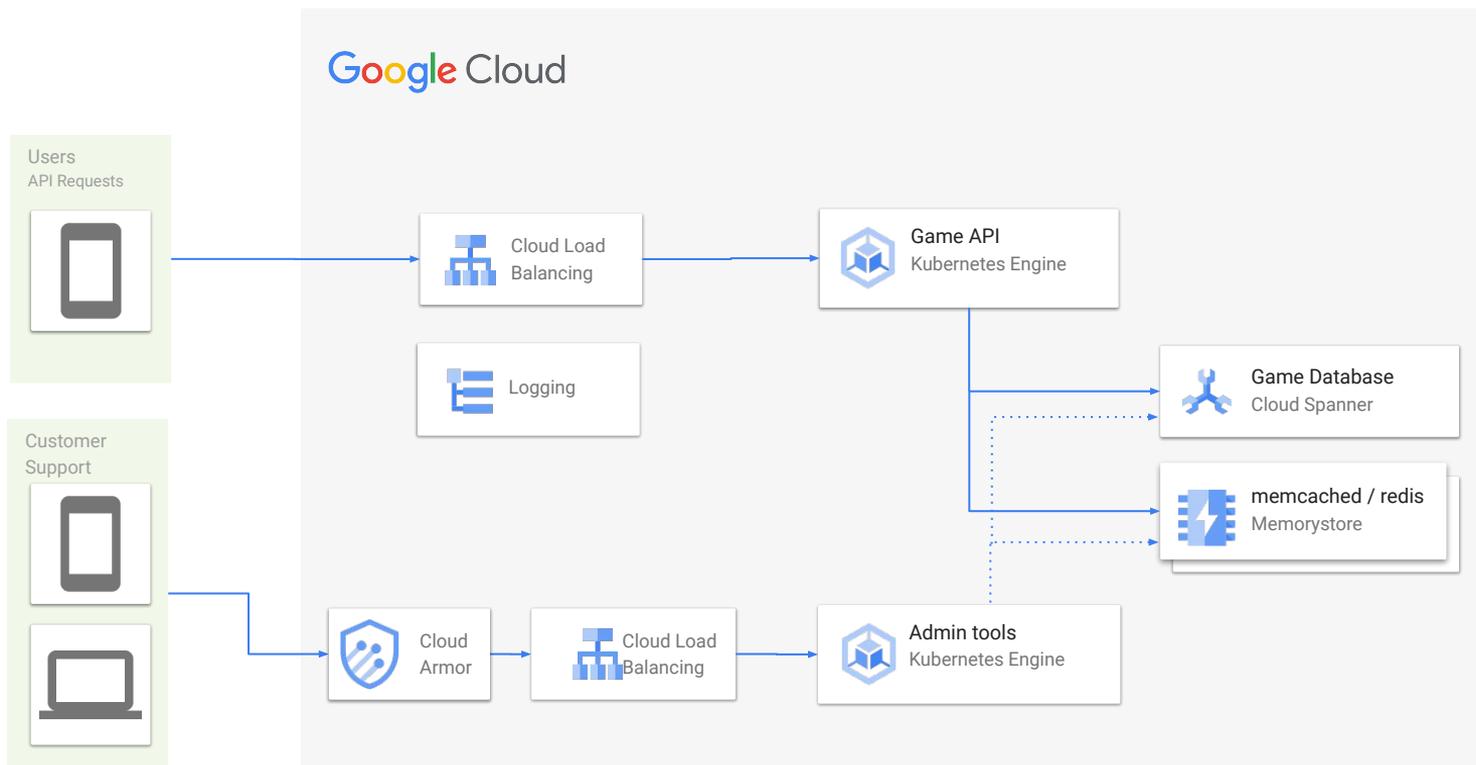
=> オンラインでスキーマ変更が可能



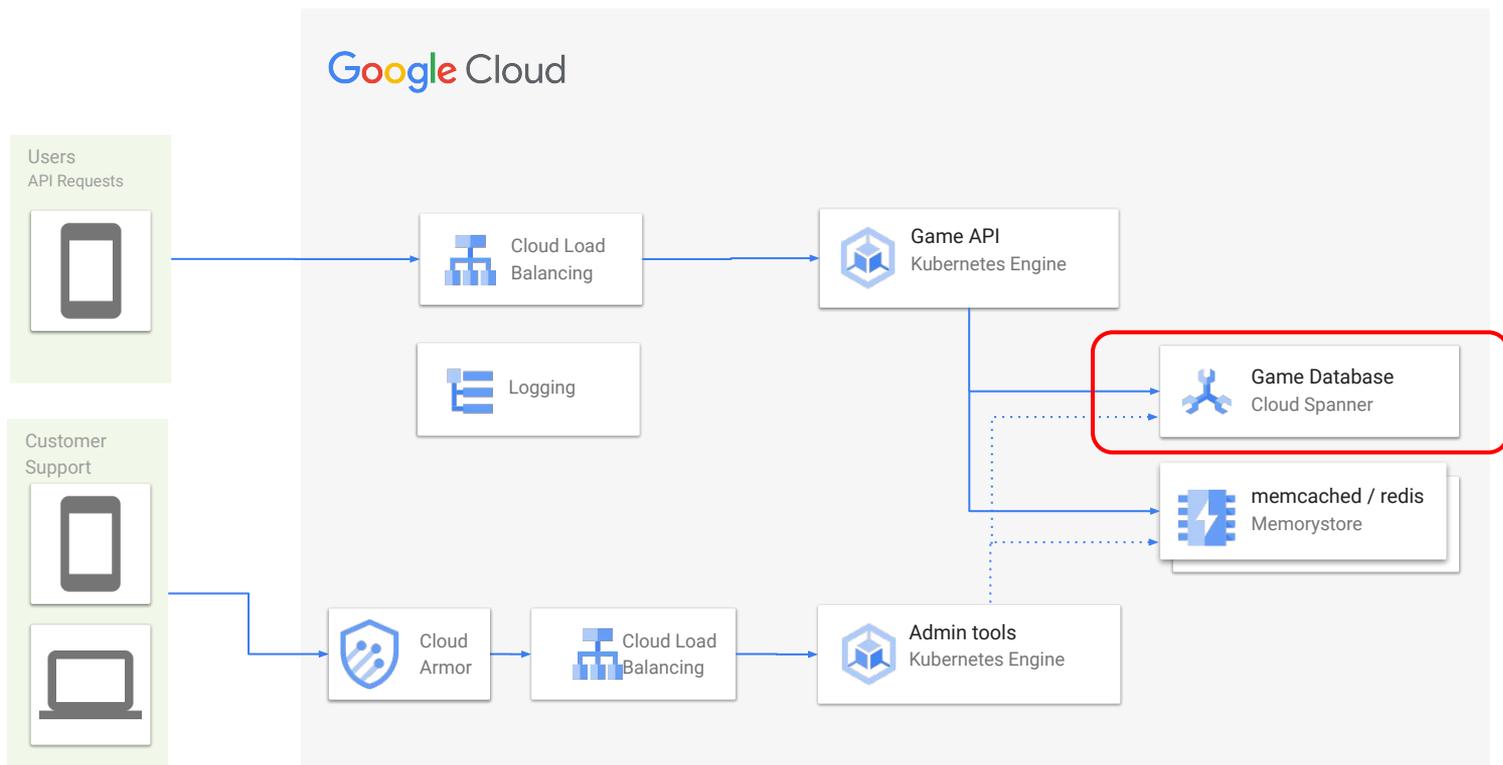
# アプリケーション構成紹介



# アプリケーション構成紹介



# アプリケーション構成紹介



# Cloud Spanner

## 負荷試験で知る Split 分割

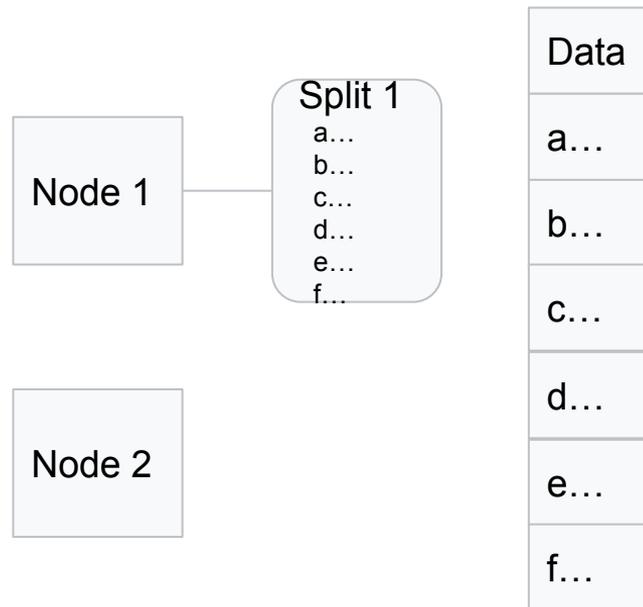


# Split とは？

スプリットは連続した行の範囲を保持する

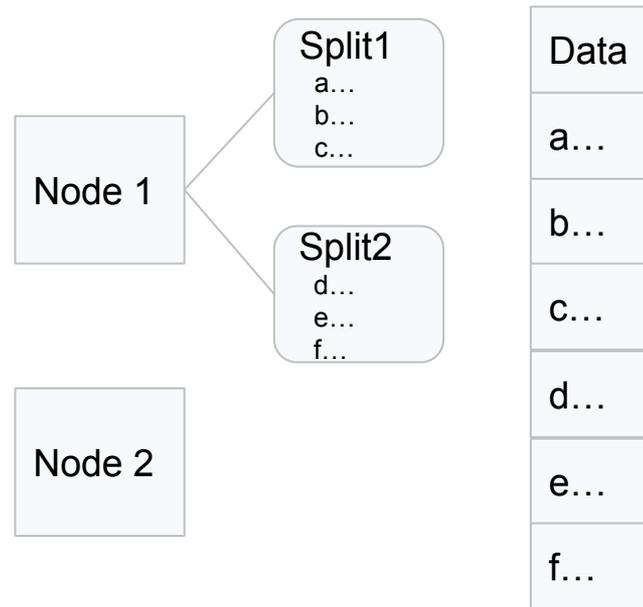
各テーブル行の Primary Key / Secondary Index によって  
担当される Split が決まる

※右図の a... などは Primary Key の値



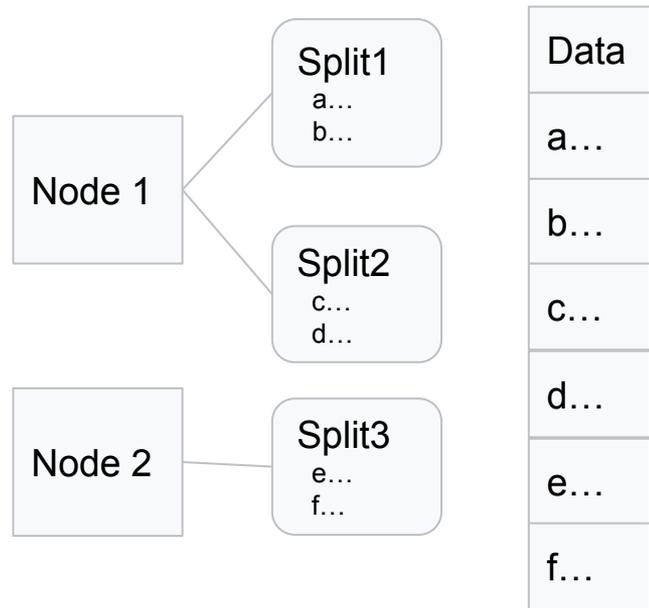
# Split とは？

負荷または扱うデータサイズに応じて Split は分割される



# Split とは？

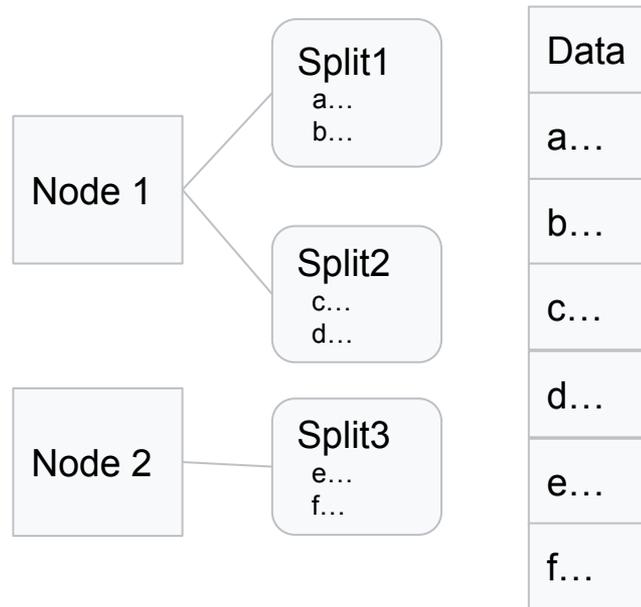
ノードの台数や負荷状況に応じて Split は再配置される



# Split とは？

利用者はいつどこのノードに Split の配置・分割するなど  
コントロールすることはできません

また、各ノード上で Split がどのように配置されているかを知る  
ことはできません



# Split 分割・再配置を体験する

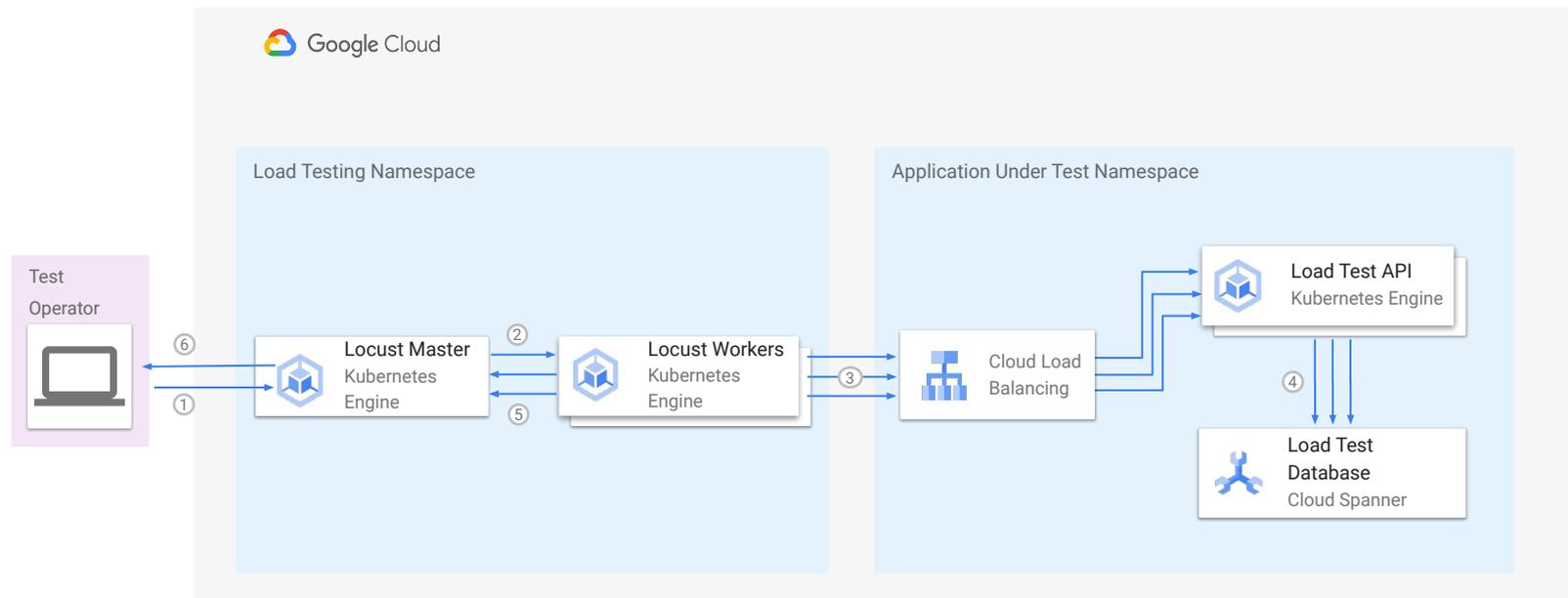
簡易的な負荷試験で Split 分割・再配置のふるまいを確認します

Cloud Spanner へアクセスする簡単なアプリケーションを用意し、Locust を用いて負荷をかけます

- Cloud Spanner へのアクセスは同一テーブル、同一レコードに対して以下
  - read: 5
  - write: 4
- Node: 20
- 目標は 10000 RPS



# 負荷試験の構成



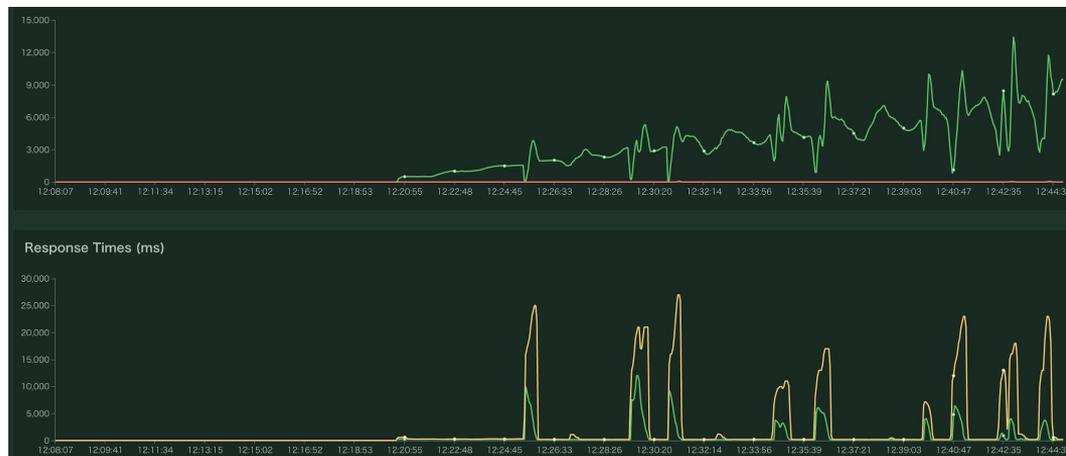
- |  |                         |
|--|-------------------------|
| 1 Tester starts test                       | 4 Talk to Cloud Spanner |
| 2 Master starts load                       | 5 Worker report results |
| 3 Workers send request to LoadTest Service | 6 Tester views results  |



# 負荷試験結果 (Split 分割・移動を観測)

Split 分割が起きていない状態の  
Cloud Spanner に負荷をかけた際に  
観測したグラフです

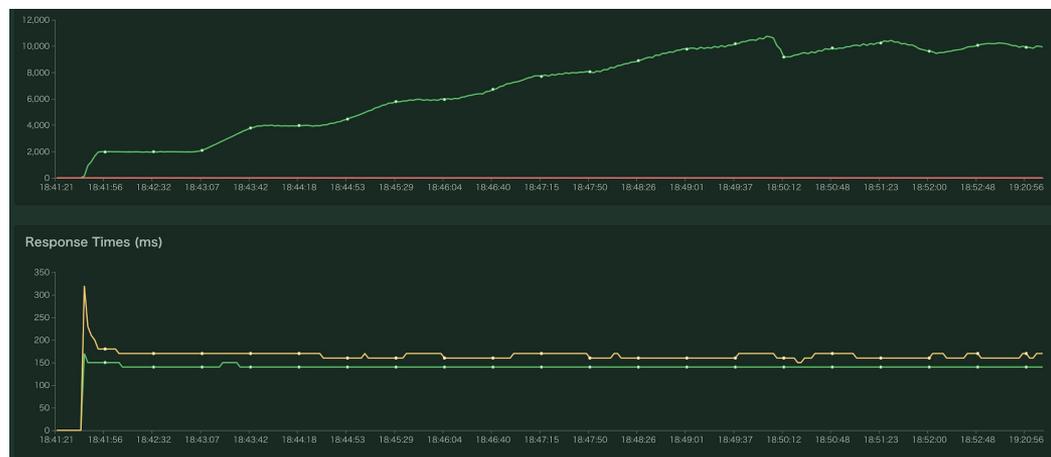
- 上のグラフが RPS
- 下のグラフがレスポンスタイム



# 負荷試験結果 (Split 分割済)

再度同じ負荷をかけて観測したグラフです

- 上のグラフが RPS
- 下のグラフがレスポンスタイム



# 負荷試験結果 (Split 分割済)

立ち上がりでスパイクが起きているのは  
セッション確立時の影響によるものです

以後はセッションプールの利用によって  
安定しています



# Cloud Spanner

## ノード数変更による影響



# ノードを1度に大きく減らす

こちらは Cloud Spanner のノード数を  
15 台から 10 台へ減らしたときのグラフです

- 上のグラフ緑色のラインがノード数
- 下のグラフがレイテンシ

Google Cloud の BLOG でも紹介されています

- ノードを減らす際には全体の 10% 程度にする
- 減らしてから 10 分以上時間をあける

[Scaling Cloud Spanner Instances](#)



# ノードを少しずつ減らす

Cloud Spanner のノード数を 20 台から  
少しずつ減らしていきます。

- 上のグラフは CPU 合計
- 下のグラフはノード数



# ノードを少しずつ減らす (CPU 影響)

13:50 頃から 1 台ずつ減らしています

減らした後は 10 分以上時間をあけています



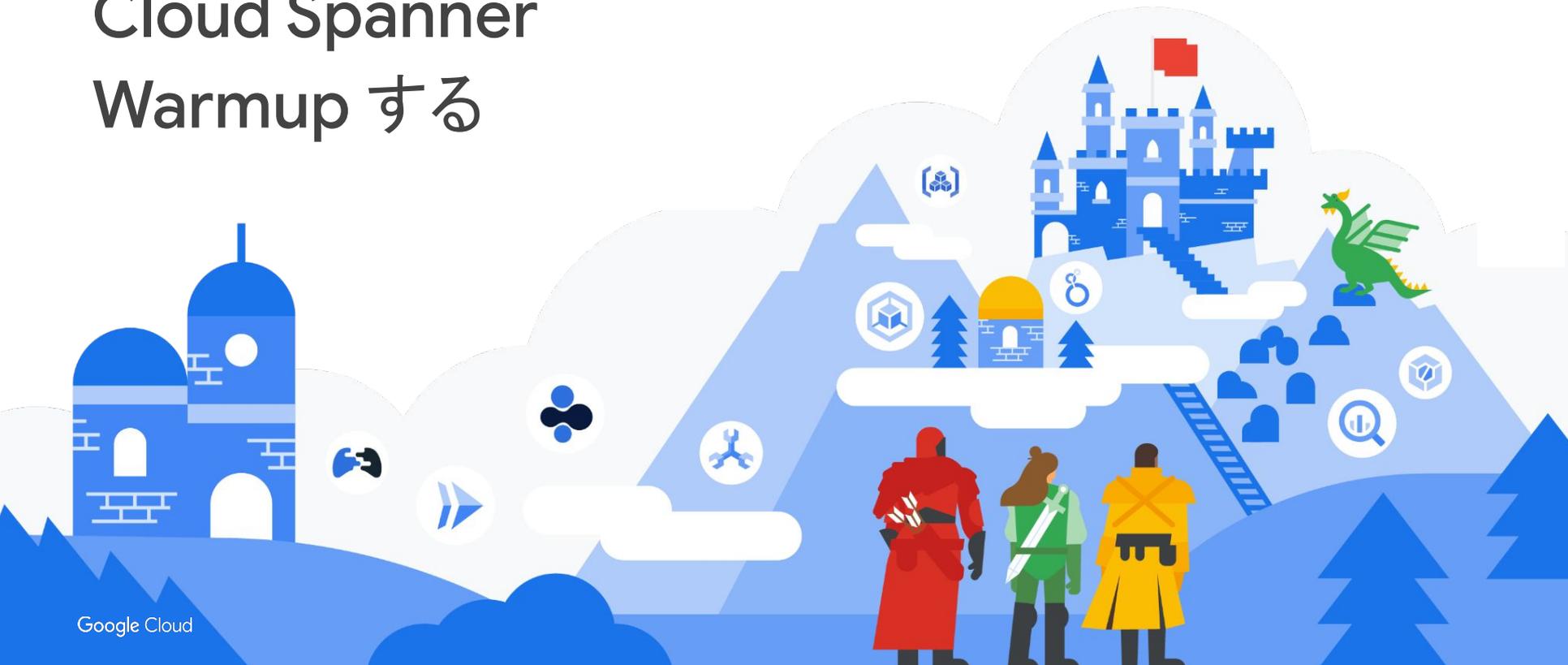
# ノードを少しずつ減らす(レイテンシ影響)

立ち上がりで Split 分割・移動によるレイテンシ悪化が見られますが、困んだあたりを注目してもらえればと思います。

- 上のグラフが RPS
- 下のグラフがレスポンスタイム



# Cloud Spanner Warmup する



# Warmup とは？

サービス開始などの大きなアクセスが想定される際に Split を事前に分割・移動させておくこと

想定されるワークロードに合わせて、負荷をかけるのが推奨されています



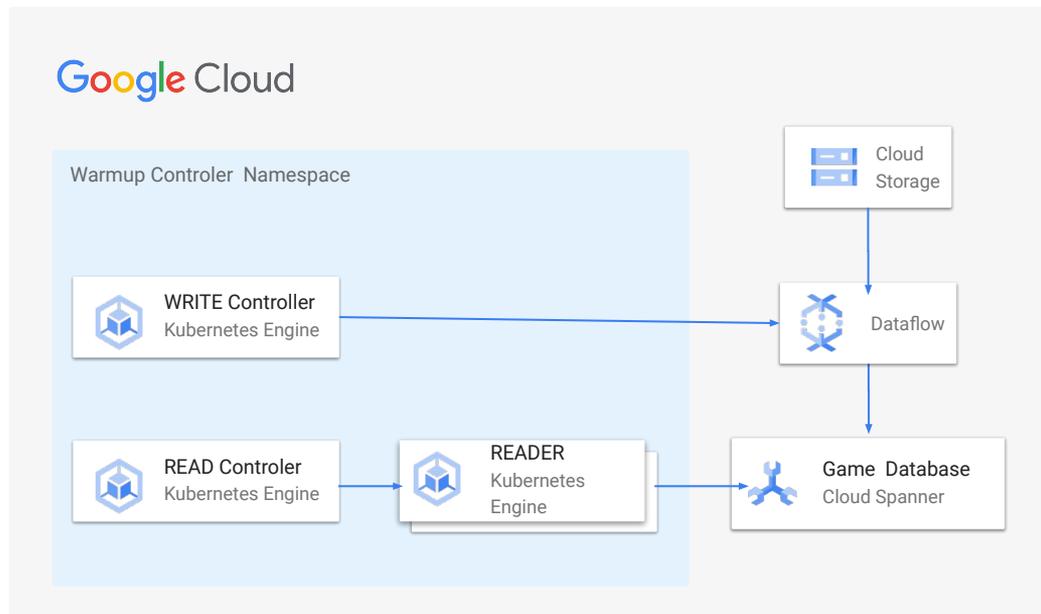
# ウォームアップ例

WFS での Warmup 例です

こちらの図は簡略化していますが、  
DataFlow で WRITE を行い、  
GKE pod が READ を行う形です

詳細は GREE 石松の発表資料にて  
ご確認いただければと思います

<https://techcon.gree.jp/2021/session/ShortSession-6>



# ウォームアップ例

ゲームのワークロードに合わせて負荷をかけていきます

ノード数は段階的に増やしています  
ノード数が少ない方が Split が少ない時点では  
負荷をうまく集中させられるためです

コンピューティング容量

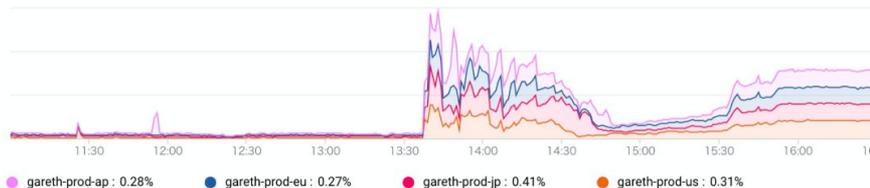
コンピューティングユニ...  
処理ユニット



● 処理ユニット: 35000

CPU 使用率 - 合計 ⓘ

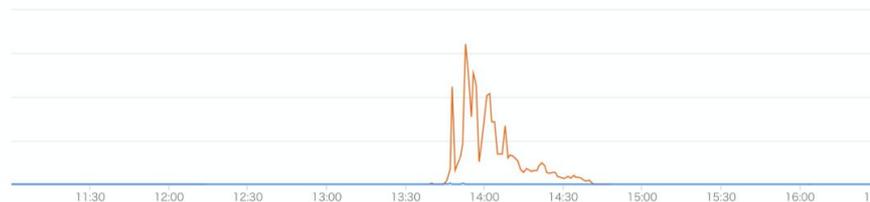
Group By  
データベース



レイテンシ ⓘ

閾値  
読み取り / 書き...

パーセンタイル  
99



● 書き込み - 99 パーセンタイル: 17.3ms

● 読み取り - 99 パーセンタイル: 81.9ms

# ウォームアップ例

CPU 使用率を見てもらうと分かる通り  
2回に分けて負荷をかけています

1度目は WRITE アクセス  
2度目は READ アクセス

コンピューティング容量

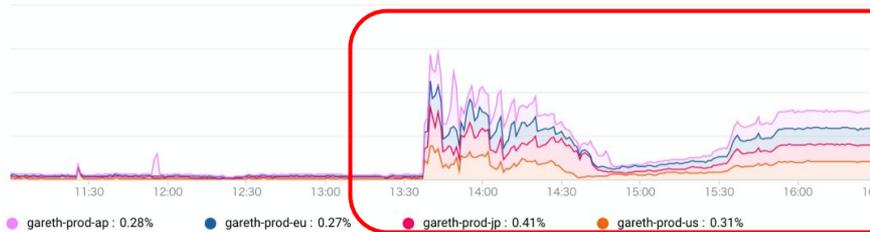
コンピューティングユニ...  
処理ユニット



● 処理ユニット: 35000

CPU 使用率 - 合計

Group By  
データベース



レイテンシ

閾値  
読み取り / 書き...

パーセンタイル  
99



● 書き込み - 99 パーセンタイル: 17.3ms

● 読み取り - 99 パーセンタイル: 81.9ms

# ウォームアップ例

1 度目の山ではレイテンシ悪化が見られました  
ここで Split 分割・移動が起きていそうです

2 度目の山ではレイテンシ悪化が見られません  
Split が十分と判断しています

Warmup 完了まで 1 時間ほどでした

コンピューティング容量

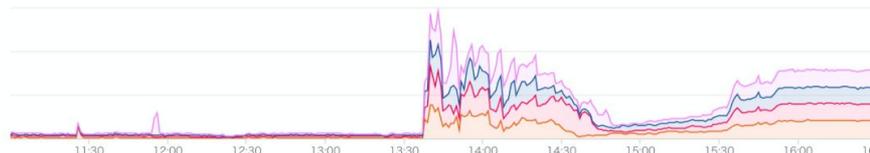
コンピューティングユニ...  
処理ユニット



● 処理ユニット: 35000

CPU 使用率 - 合計 ⓘ

Group By  
データベース



● gareth-prod-ap : 0.28% ● gareth-prod-eu : 0.27% ● gareth-prod-jp : 0.41% ● gareth-prod-us : 0.31%

レイテンシ ⓘ

関数  
読み取り / 書き...  
パーセンタイル  
99



● 書き込み - 99 パーセンタイル: 17.3ms ● 読み取り - 99 パーセンタイル: 81.9ms

# Cloud Spanner

## HotSpot による影響



# HotSpot とは？

Cloud Spanner でデータを扱う単位が Split ですが、

Primary Key の作り方によっては同一 Split に対してアクセスが集中してしまう問題があります

これが HotSpot です



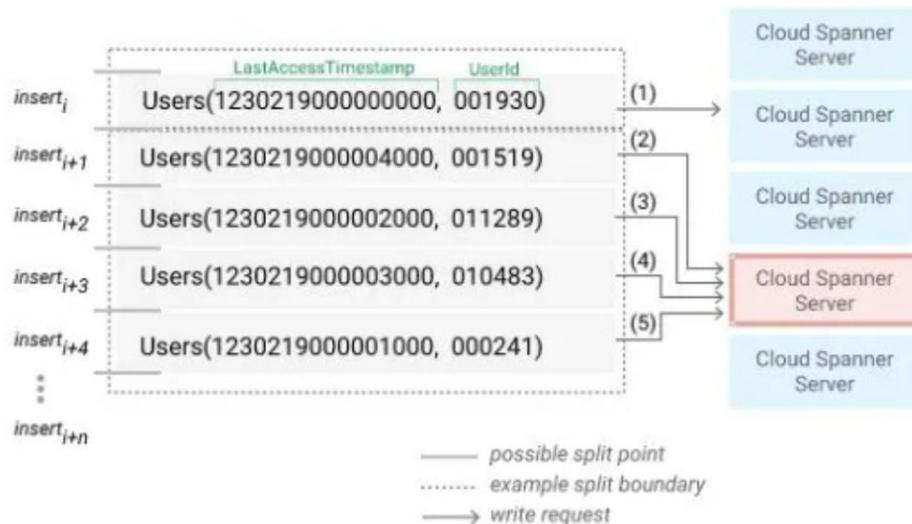
# HotSpot とは？

## キーのアンチパターン

- シーケンシャルな値
- パターンの少ない接頭辞をつける
- timestamp のような時間

これらは Split が偏る問題があります

一意キーのハッシュ値や UUIDv4 等が推奨されています



<https://www.slideshare.net/GoogleCloudPlatformJP/gke-spanner-cloud-spanner>

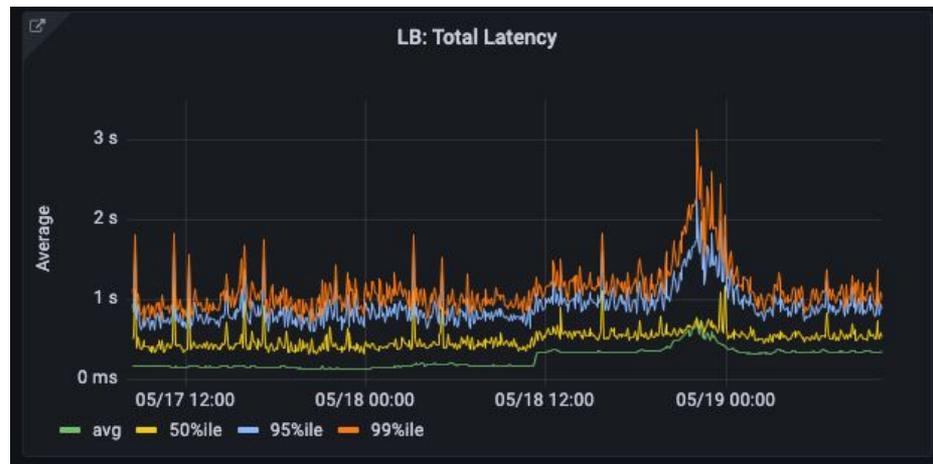


# HotSpot が起きるとどうなる

こちらは、実際にゲーム内で HotSpot が起きたときの LB Latency のグラフです。

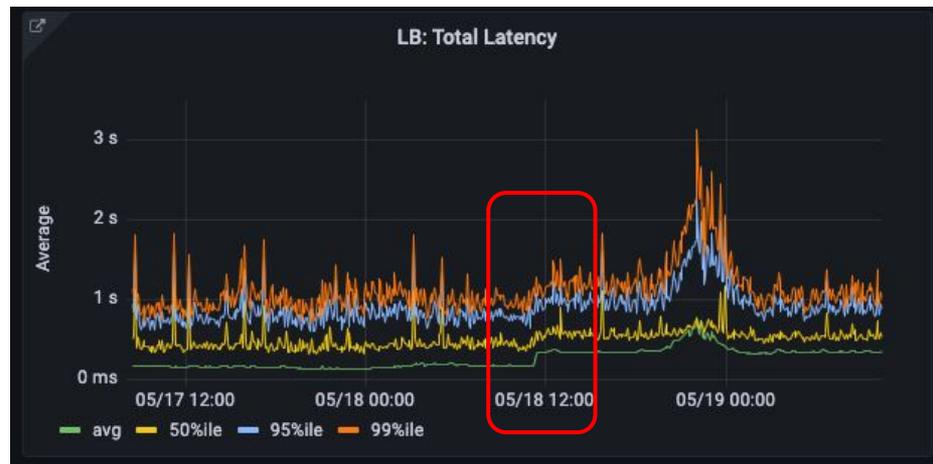
単一のレコードに対して、多数の READ アクセスが集中してしまった状態です。

WRITE はしていないレコードになります。



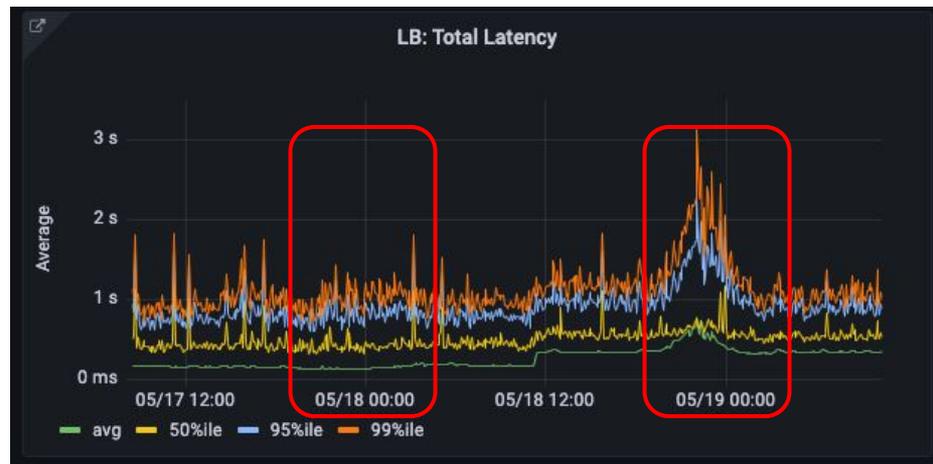
# HotSpot が起きるとどうなる

5/18 12:00 付近でレイテンシ悪化が見られます。  
こちらが HotSpot が発生しはじめた箇所です



# HotSpot が起きるとどうなる

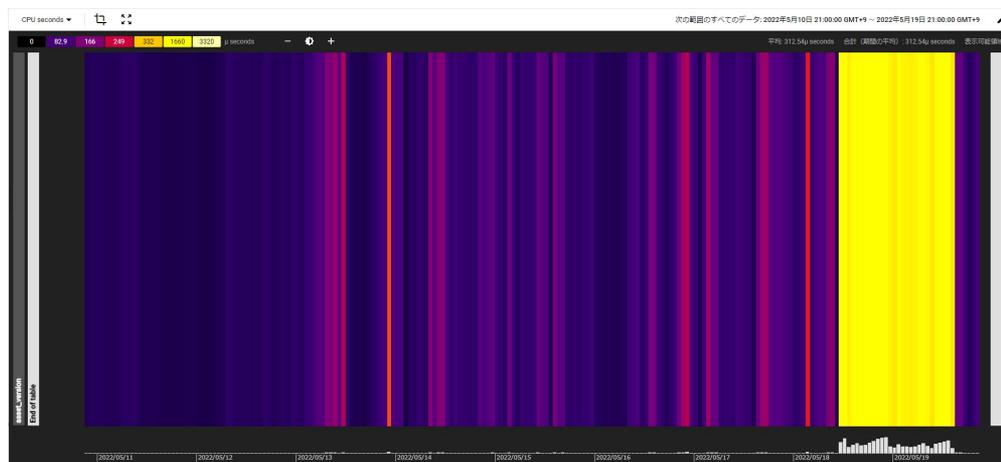
5/18～5/19 にかけての夜間の山については、  
前日の同時間帯と比べて大きくなっており、  
全体的に負荷が高まっている状況でした



# HotSpot が起きるとどうなる

Key Visualizer で見た CPU seconds の  
様子です

問題の起きたタイミングからハッキリと  
黄色く出ているのが確認できます



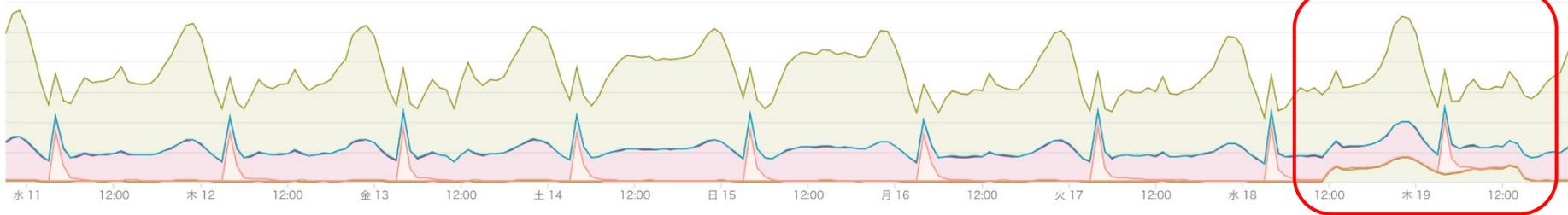
# HotSpot が起きるとどうなる

CPU 使用率についても変化が見られました

CPU 使用率 - オペレーション タイプ ⓘ

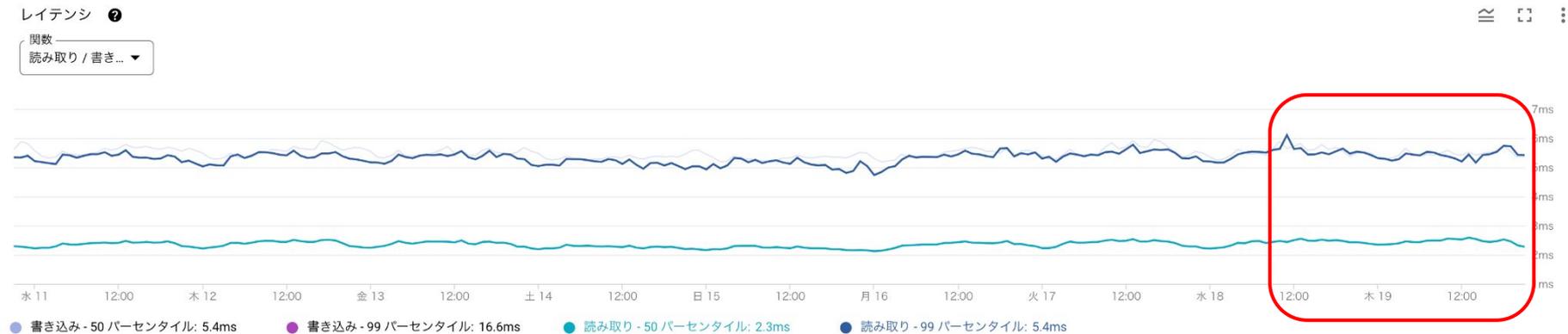
優先度

すべての優先度 ▼



# HotSpot が起きるとどうなる

レイテンシには影響がないように見えます

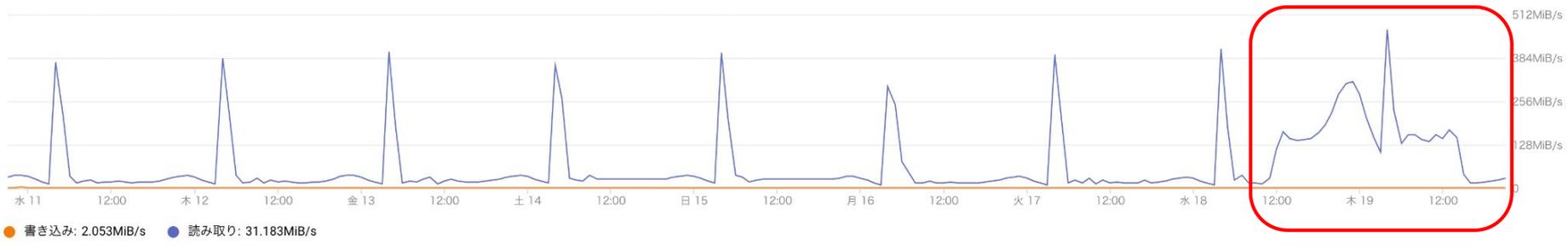


# HotSpot が起きるとどうなる

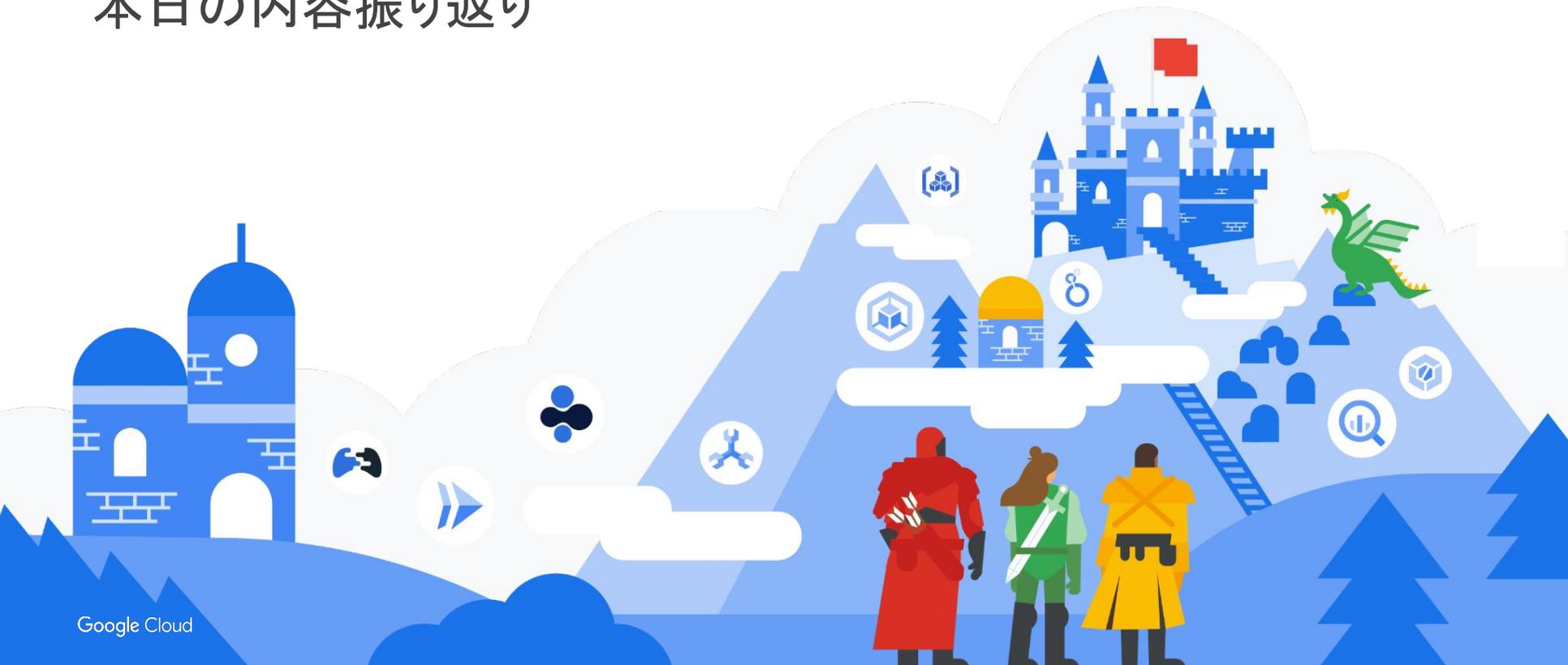
スループットが上がっています

こちらが影響して、LB Latency が高まったと考えられます

スループット



# 本日の内容振り返り



# Split を理解することが大事

## ウォームアップによる事前の Split 分割

- 急なトラフィックにも耐えられる
- レイテンシの様子を見て判断

## PK / Secondary Index の作り方

- uuidv4 などが推奨
- ホットスポットが起きないように気を付ける

## ホットスポットが起きると

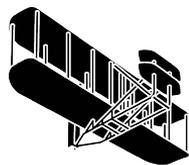
- パフォーマンス悪化
- モニタリングや KeyVisualizer を確認

## ノードの減らし方

- 全体台数の 10%
- 10 分以上間隔をあけて減らす



PR WFS では新しい仲間を募集しています！



WRIGHT FLYER  
STUDIOS

<https://www.wfs.games/recruit/special/>



# Thank you

