Hewlett Packard Enterprise

MySQL Cluster ユーザー事例紹介 ～JR東日本情報システム様における導入事例～

日本ヒューレット・パッカード株式会社
テクノロジーコンサルティング事業統括
高橋 智雄

2015年12月15日

自己紹介

- 氏名：高橋 智雄
- 所属：日本ヒューレット・パッカード株式会社
- 仕事：データベース関連のコンサルタント
  Oracle Databaseのトラブル対応やチューニングなど
  最近はオープンソースのDBMSも
- 最近：MySQL Cluster検証、技術支援
  MS SQL Server PDWソリューション開発
  Vertica導入
  Oracle RAC導入
  Oracle Database性能検証
  PostgreSQL 技術コンサルティング
  Oracle DatabaseからPostgres Plusへの移行PoC
  Oracle Databaseチューニング支援
  MySQL 移行支援

Hewlett Packard Enterprise

はじめに

ＪＲ東日本情報システム様が構築・運用している「障害情報システム」
には、 MySQL Cluster とHPE Moonshot Systemが採用されています。

本セッションではMySQL Cluster とHPE Moonshot Systemが採用された背景や、システムの実現性を追求した数多くの検証内容などをご紹介します。

システム全体像の事例紹介サイト

http://h50146.www5.hp.com/products/servers/proliant/casestudy/jeis/

Hewlett Packard Enterprise

アジェンダ

1. MySQL Cluster導入の背景
2. MySQL Cluster概要
3. MySQL Cluster On HPE Moonshot System検証結果
4. まとめ

1.MySQL Cluster導入の背景

ＪＲ東日本情報システム様と障害情報システムのご紹介

ＪＲ東日本情報システム様概要

- ＪＲ東日本様の情報システム部門が独立して1989年に誕生。略称はJEIS
- ＪＲ東日本グループ約70社を中心に広くシステム提案・開発・運用を手がけている。本日紹介する「障害情報システム」もその中の一つ

障害情報システムの目的と位置づけ

- あるシステムで発生した問題に対して、その原因と解決方法を「障害情報システム」で全社共有し、同種の問題を未然に防ぐ
- ナレッジベースであり社内向けシステムという位置付けではあるが、業務上重要なシステム
- 障害情報システムの構築をとおして、**将来的にミッションクリティカルなシステムの構築**に役立つ新しい技術、ノウハウを習得

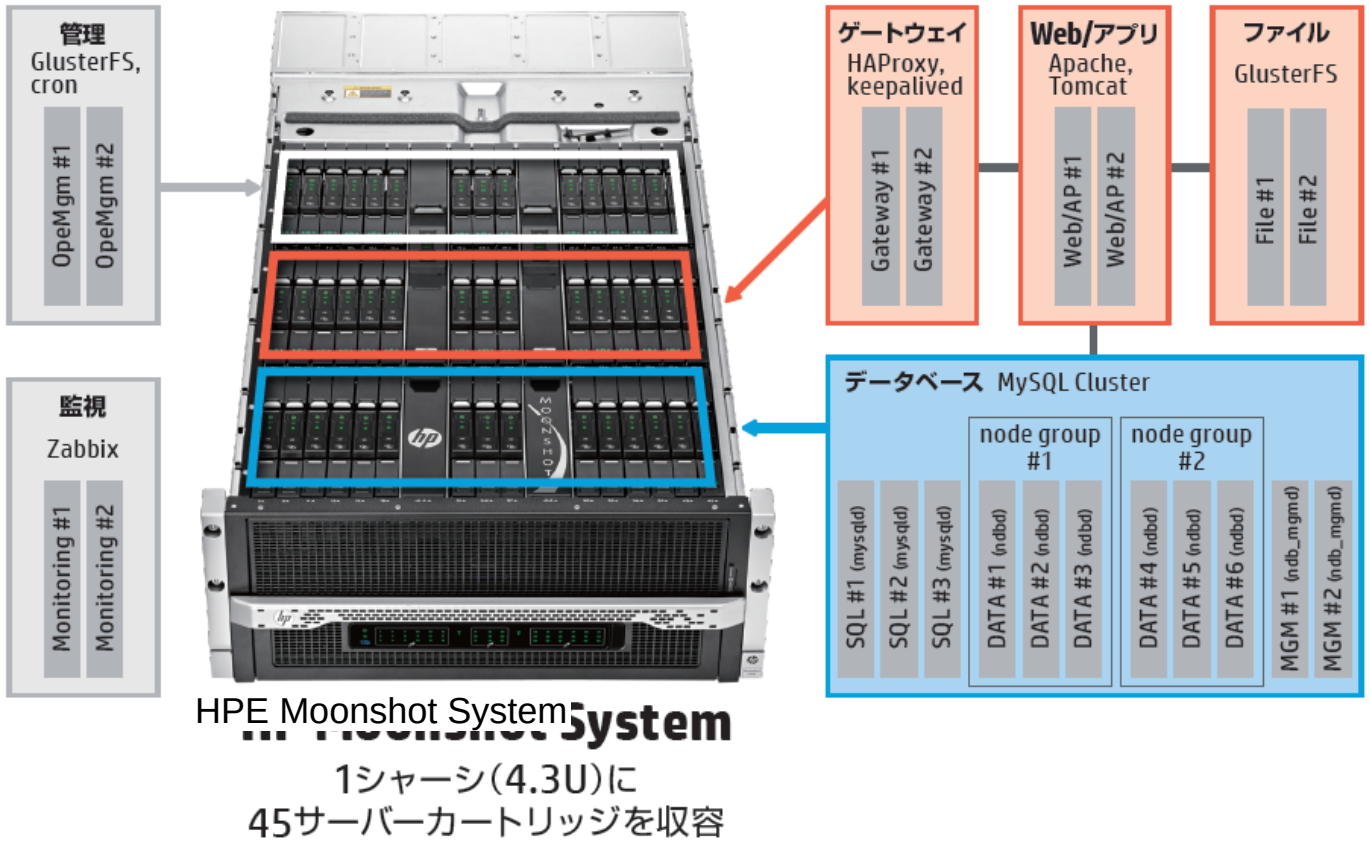
Hewlett Packard Enterprise

障害情報システムの特徴
オープンソースソフトウェアの採用

- システム構築における方針と**チャレンジ**
  - オープンソースソフトウェア（OSS）を全面的に採用
  - 新しい超高密度サーバー「HPE Moonshot System」の採用
    1シャーシ／45サーバーノード内に障害情報システムの全機能を集約し、
    「**1シャーシ＝1システム**」を実現

HPE Moonshot Systemご紹介

障害情報システムの全機能をHPE Moonshot System 1シャーシに集約



- 第2世代カートリッジ m300
  - CPU：Atom C2750（8コア）
  - メモリ：32GB
  - ネットワーク：1G bps× 2
  - ディスク：1TB HDDもしくは 240GB SSD

Hewlett Packard Enterprise

データベースに対する課題

データ保持方式と可用性の実現方式が課題

- データ保持方式
  想定データ量が1台のサーバーカートリッジが搭載できるディスク（最大1TB）を超える可能性があったため、外部の共有ストレージを使わずMoonshot Systemのシャーシ内にデータを保持する方式の検討が課題
- 可用性
  サーバー障害によるサービス停止時間を最小限にする冗長化方式の検討が課題

高可用性を備えた分散DBMS MySQL Clusterを採用

Hewlett Packard Enterprise

導入までの体制と流れ

- 体制
  - JEIS様：障害情報システムの開発主幹
  - エスケイケイ様：JEIS様の開発子会社で、実際の開発を担当
  - 日本ヒューレット・パッカード：MySQL Clusterの検証と開発時の技術支援
- 導入までの流れ
  - 2014年1月～3月：第1回目検証
    第一世代のMoonshotを使用して、性能検証や耐障害性を検証
  - 2014年4月～5月：第2回目検証
    実システムで使用する第二世代Moonshotでの性能検証
  - 2014年6月～2015年1月：システム開発

大きな問題なし

本日の後半でご紹介

Hewlett Packard Enterprise

2.MySQL Cluster概要

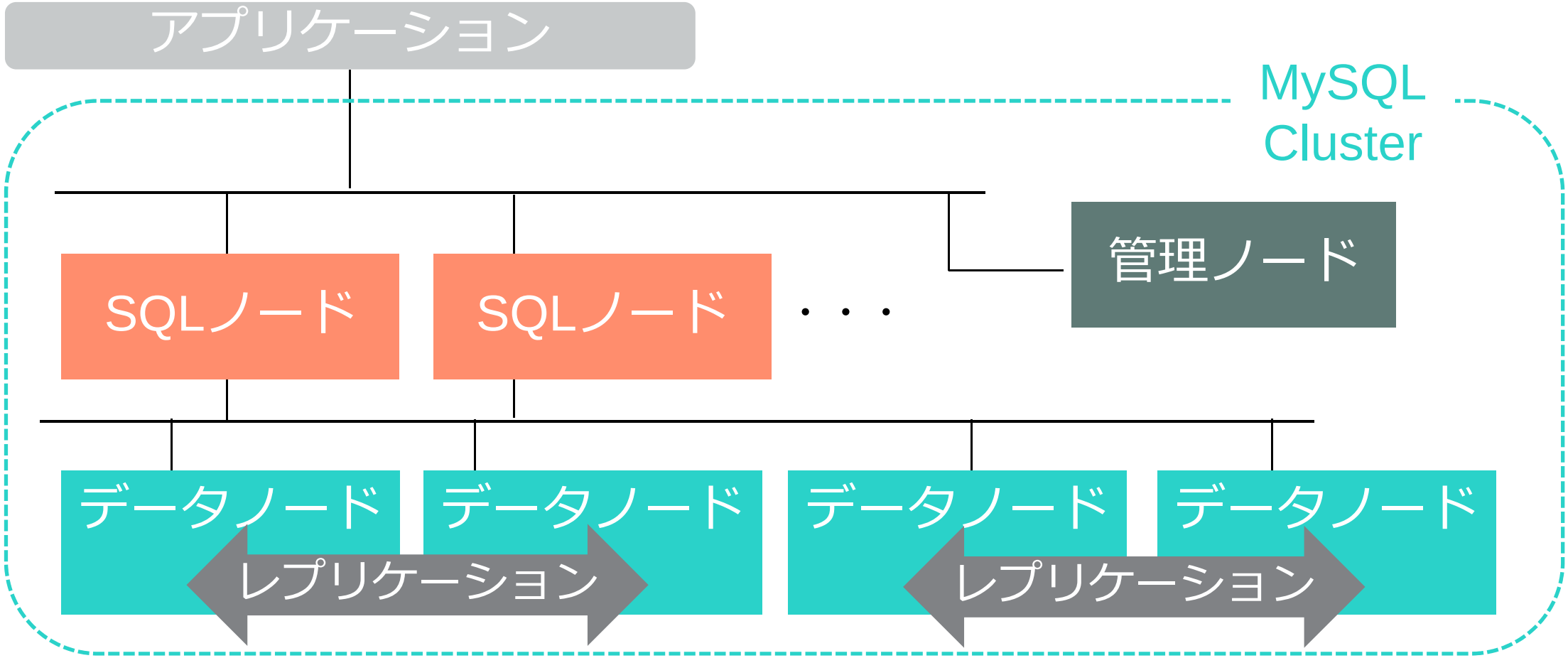
MySQL Clusterとは

シェアードナッシングアーキテクチャのインメモリ分散データベースシステム

- リアルタイム・パフォーマンス
  - インメモリデータベースであるため非常に高速
- 高可用性
  - SPOFのないシェアードナッシング・アーキテクチャ
  - 各データ・ノードのデータは、同期的に他のデータ・ノードにレプリケート
  - 障害時は通常1秒以内にクラスタ内の他のノードに自動フェイルオーバー
- スケーラビリティ
  - データをノード間で自動的にシャーディングするためスケーラブル
  - オンラインでのノード追加が可能
  - Active／Activeアーキテクチャであり、どのノードでも更新処理が可能

MySQL Clusterの構成

3種類のノードにより構成

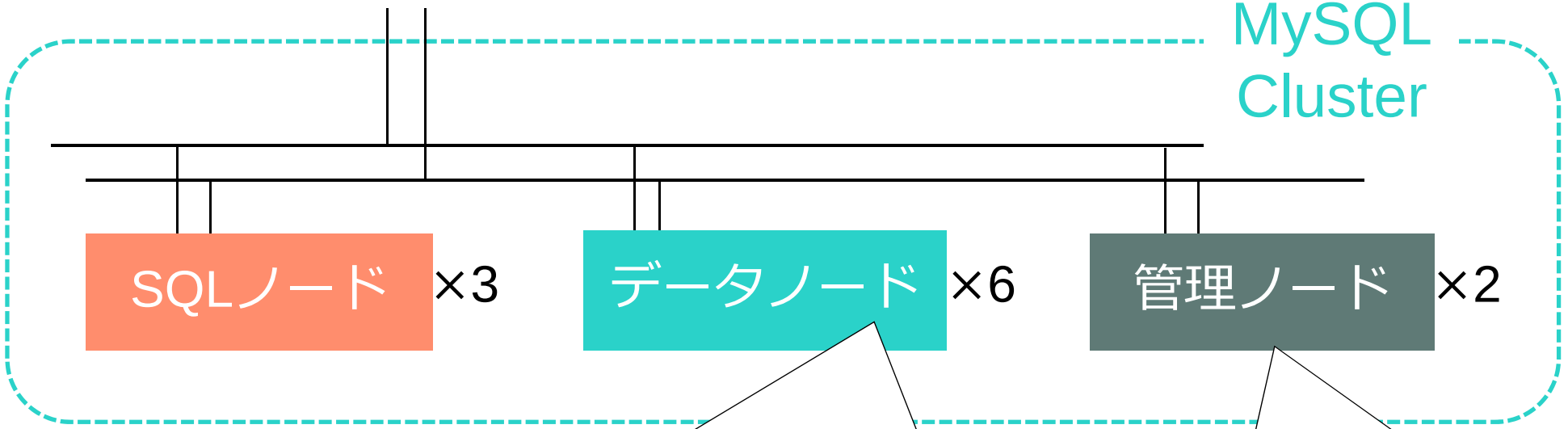


各ノードの役割

- 管理ノード
  - MySQL Cluster内のノードを管理
  - 複数台により冗長化も可能
- データノード
  - 実データを保持するノード
  - 複数台で同期データ・レプリケーション
  - 複数台により冗長性を高め、かつスケールアウトにより性能向上
- SQLノード
  - 通常のMySQL Server（mysqld）に相当
  - アプリからのSQL文を解析し、データノードにアクセスしてデータを返す
  - 複数台によりスケールアウト可能（冗長性はロードバランサやアプリで実装）

障害情報システムで採用したデータベース構成

サーバーの2重障害に備えた冗長構成を採用



- CentOS 6.5
- MySQL Cluster 7.3.5

- 3台でレプリケーションを行うグループ（ノードグループ）を2グループ構成
- 同一ノードグループの2台のデータノードがダウンしてもサービス継続が可能

管理ノードがダウンしてもサービス継続可能なため、2台構成とした

Hewlett Packard Enterprise

商用版とコミュニティ版の差異

サポート構成の制限と商用ツール

- 商用版で**サポートされるレプリカ数は1または2**のみ
- 商用版のみに付属するツール
  - MySQL Enterprise Monitor
    - SQL ノード、データノードの状態、システムリソースなどを監視
    - 問題のあるクエリの特定に役立つQuery Analyzer機能
  - MySQL Cluster Manager
    - MySQL Cluster の管理を一元化するCLIの商用ツール
    - インストール、設定変更、ノード追加、ローリングリスタートの自動化

3.MySQL Cluster On HPE Moonshot System 検証結果

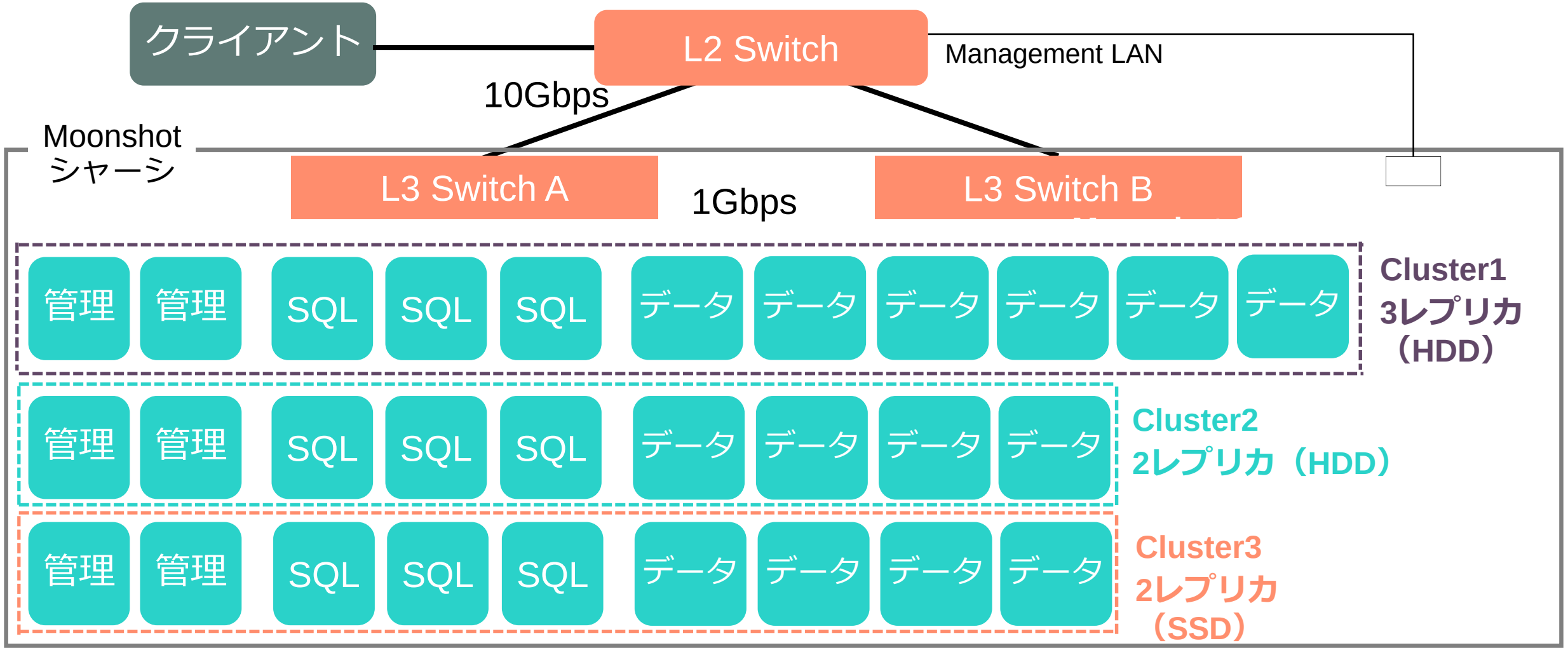
検証概要

第2世代Moonshot m300カートリッジを使用して性能検証を実施

本日ご紹介

1. MySQL Cluster構成の違いによる性能比較
2. ノード追加によるスケーラビリティ確認
3. 単一SQLの処理時間性能測定
   - ① プライマリーキーによる1レコード取得
   - ② インデックスによる複数レコード取得
   - ③ インデックスによる複数レコード取得 + ソート
   - ④ 非インデックス列への中間一致
4. メモリテーブル、ディスクテーブル（SSD）、ディスクテーブル（HDD）の性能比較

検証環境構成イメージ



Hewlett Packard Enterprise

検証1 MySQL Cluster構成による性能比較

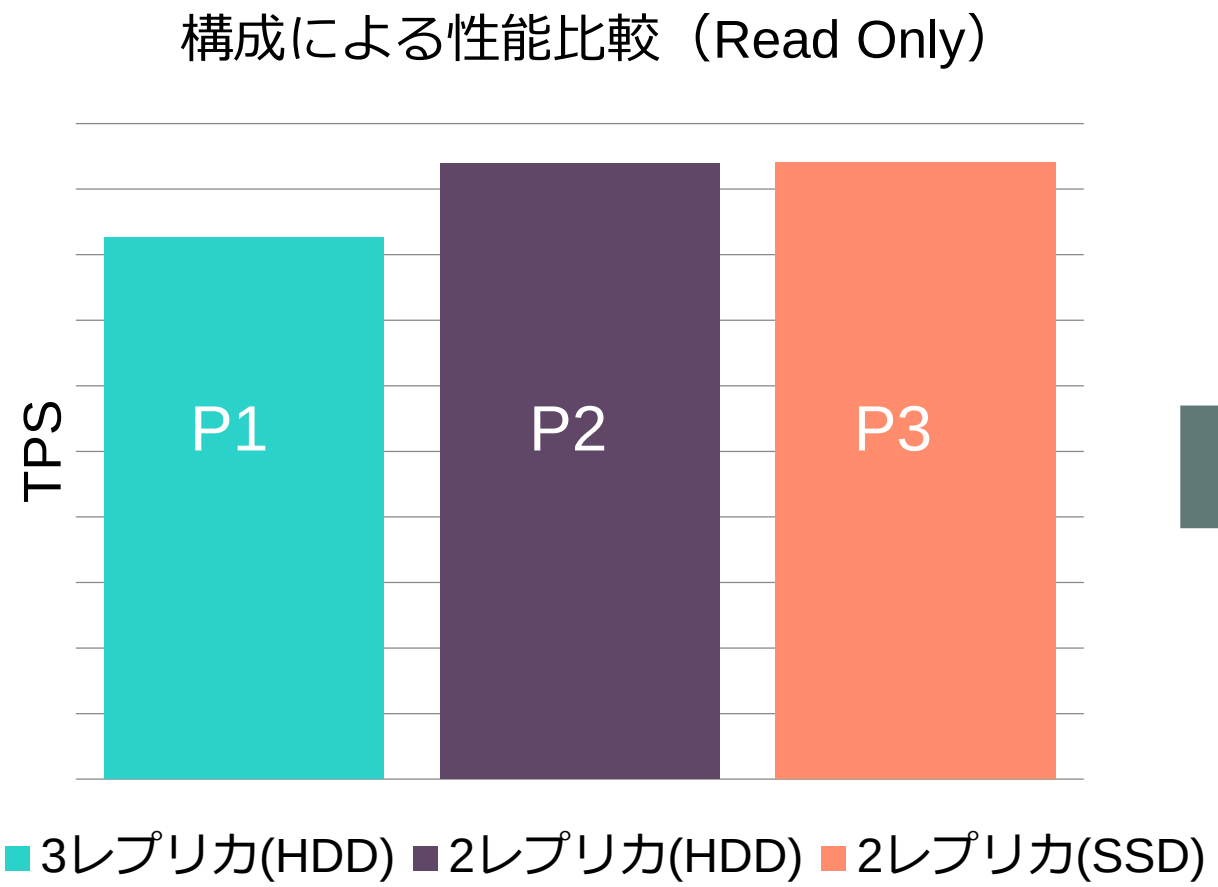
- 下記パターンの性能測定を実施し、レプリカ数の差が性能に与える影響と、メモリテーブルのみの環境で、ストレージ種別が性能に与える影響を確認

構成No	SQLノード数	データノード数	レプリカ数	ストレージ
P1	3	6	3	HDD
P2	3	4	2	HDD
P3	3	4	2	SSD

- sysbench OLTP complexモードを使用してスループット（TPS）を測定
- レコード数は100万
- それぞれの構成についてRead Only（1-1）とRead Write（1-2）の2ケースを測定

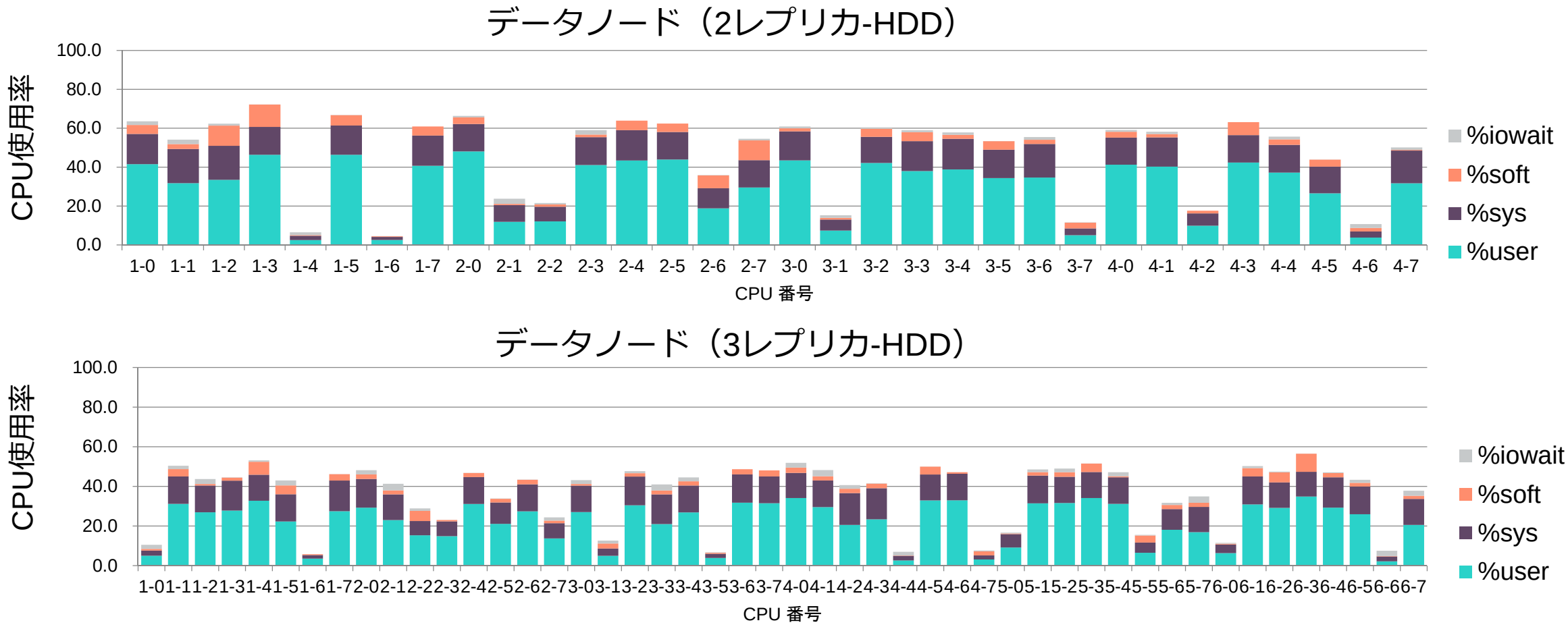
Hewlett Packard Enterprise

検証1-1（ Read Only ） 結果



- レプリカ数を3にすると約10%の性能ダウン
- Read Onlyではストレージ種別は性能にほぼ影響しない

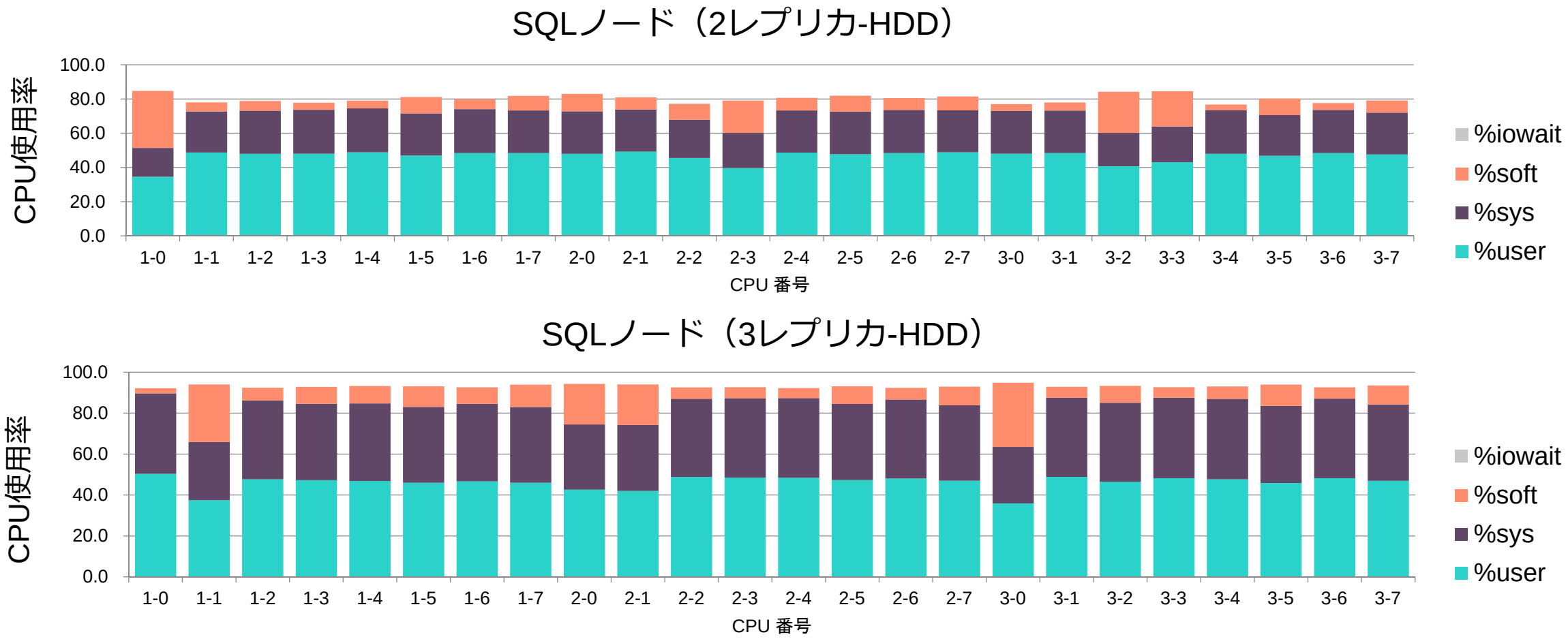
検証1-1（Read Only） CPU使用状況 ～データノード～



データノードのCPUは高負荷ではない

Hewlett Packard Enterprise

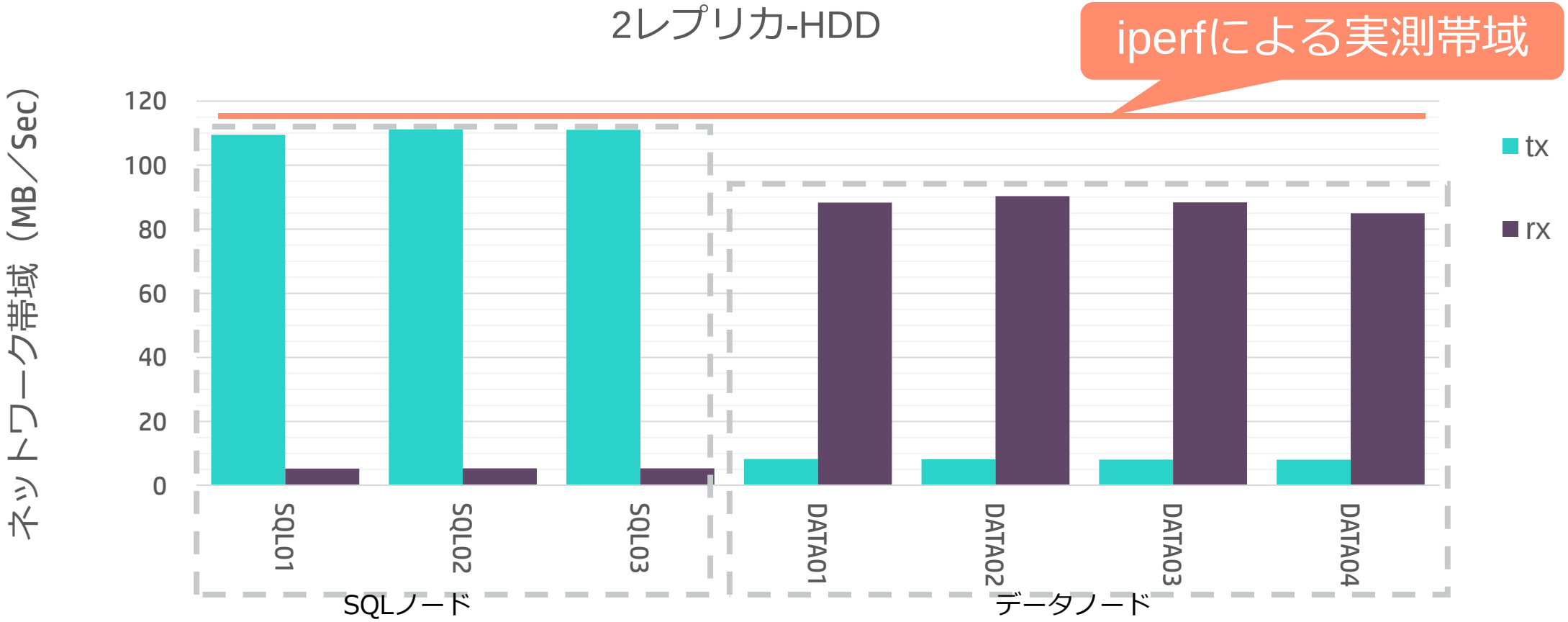
検証1-1（Read Only） CPU使用状況 ～SQLノード～



3レプリカ構成のSQLノードのCPU使用率が90%以上の高負荷状態

Hewlett Packard Enterprise

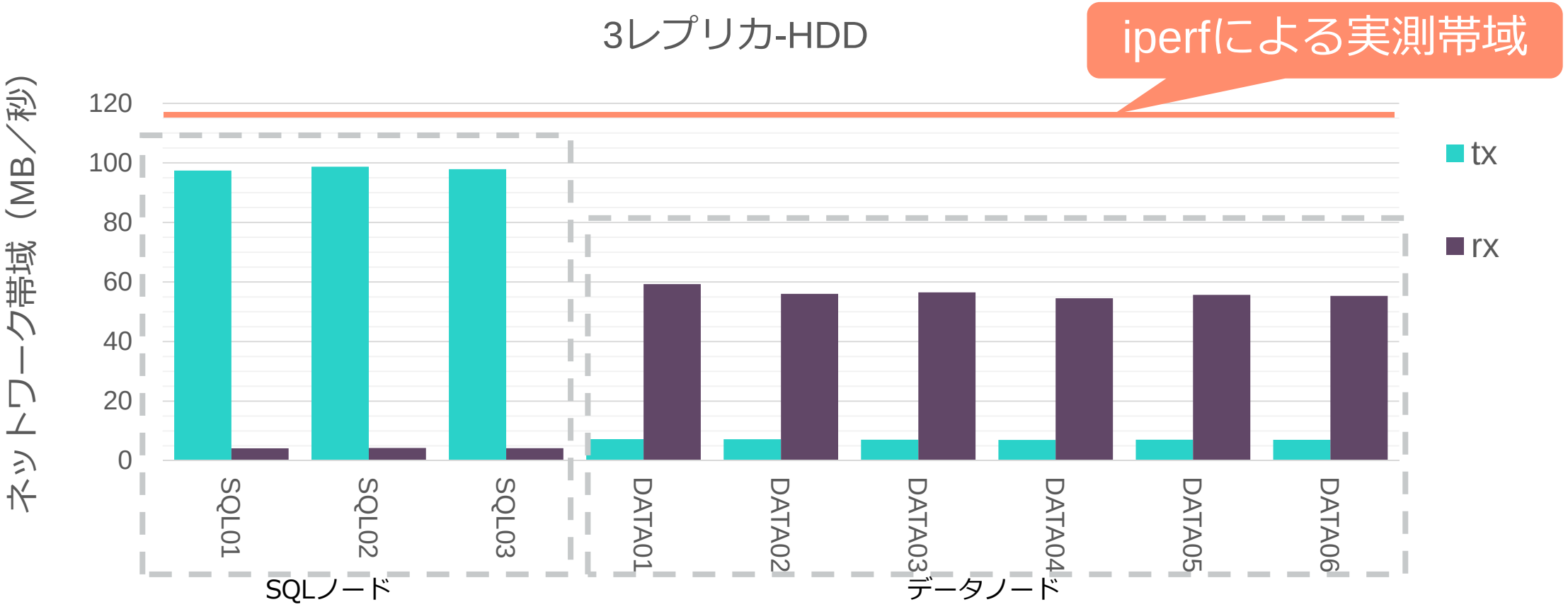
検証1-1（Read Only） ネットワーク使用状況 ～2レプリカ-HDD構成～



SQLノードではiperfによる測定結果の約117MB/secとほぼ等しい帯域を使用

Hewlett Packard Enterprise

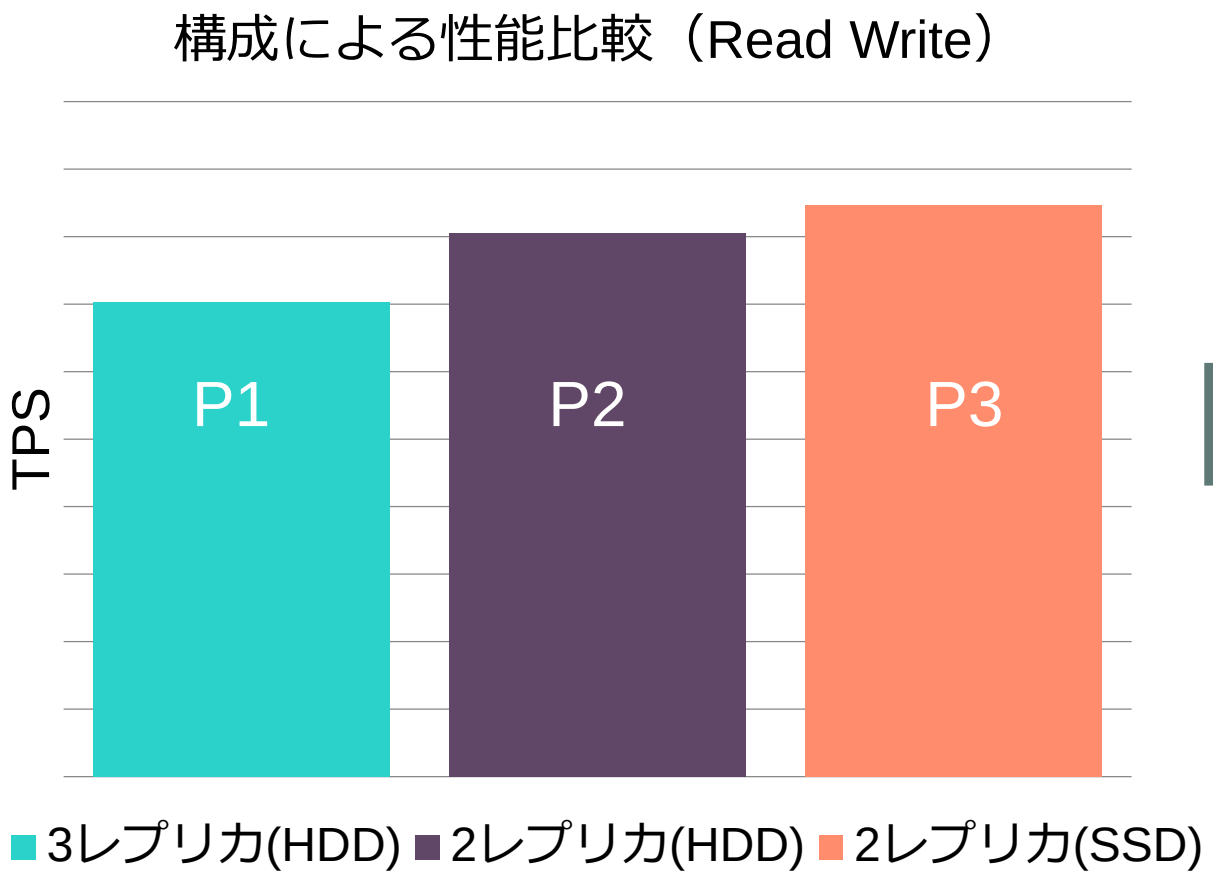
検証1-1（Read Only） ネットワーク使用状況 ～3レプリカ-HDD構成～



SQLノードのネットワークは高負荷ではあるが、
iperfによる測定結果の約117MB/秒と比較すると若干の余裕

Hewlett Packard Enterprise

検証1-2 （Read Write） 結果



- レプリカ数を3にすると約10%の性能ダウン
- Read WriteではストレージをSSDにした場合メモリテーブルでも5%程度性能向上

検証1 考察

MySQL Clusterの構成による性能比較

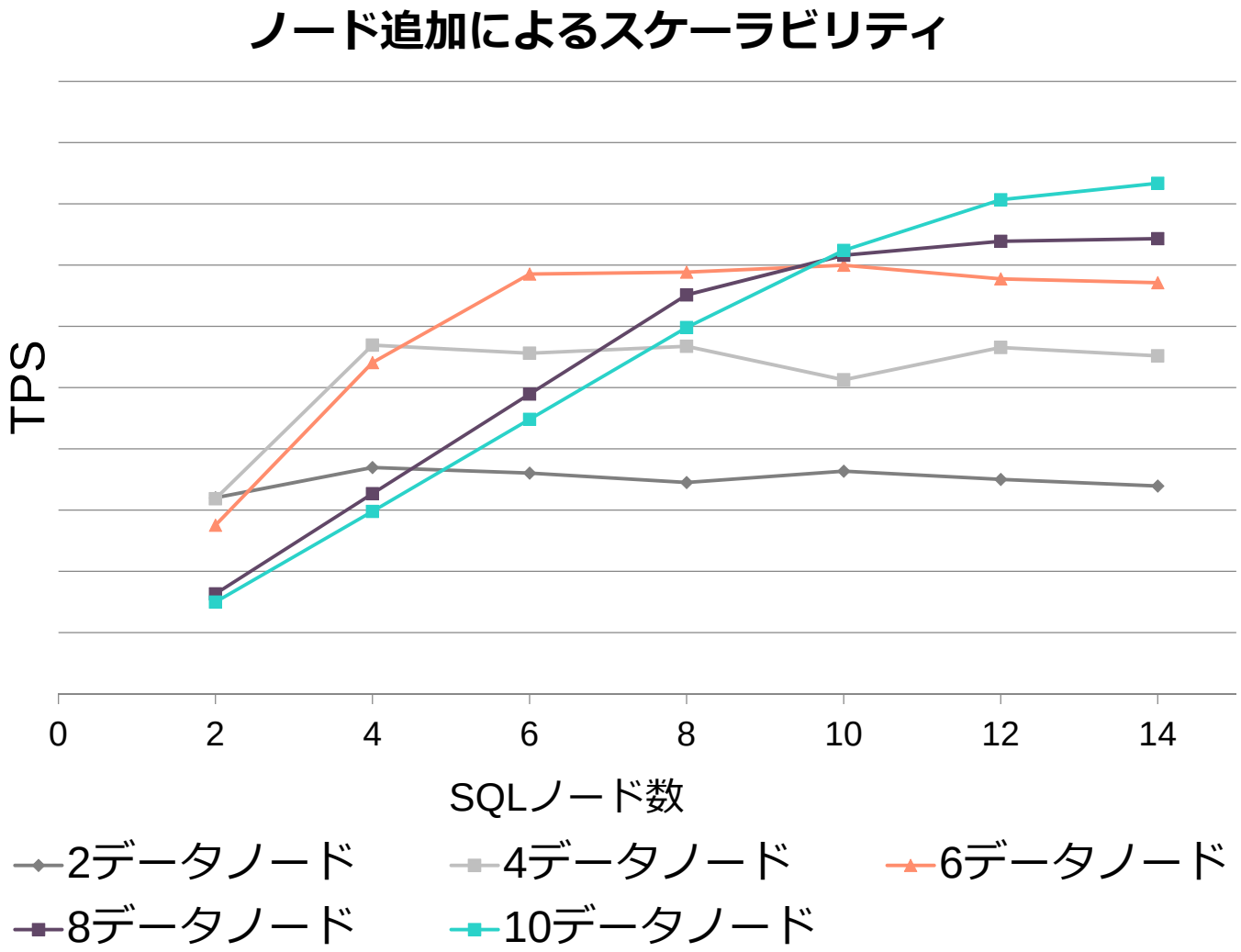
- SQLノードもCPUリソースが重要
- パフォーマンスを求めるのであれば、ネットワーク帯域は10Gbps
  最新のMoonshotでは10Gbps対応のスイッチとサーバーカートリッジも
  - m400（AppliedMicro X-Gene / ARM 64bit v8 8 cores 2.4GHz / 64GB)
  - m710（Xeon E3 1284Lv3 4 cores 1.8-3.2GHz/ 32GB)
- レプリカ数を多くするのは慎重に検討

Hewlett Packard Enterprise

検証2 ノード追加によるスケーラビリティ確認

- データノードやSQLノードを追加してスループットが向上するかどうかを確認
- レプリカ数は2に固定し、データノード数を2台～10台、SQLノード数を2台～14台に変化させて性能測定
- sysbench OLTP complexモードを使用してスループット（TPS）を測定
- レコード数は100万
- それぞれの構成についてRead Onlyの性能を測定

検証2 結果



- ノードを追加することにより、スループットは向上
- データノードとSQLノードを適切な割合で増加させることが重要
- データノード数が多い場合、MySQL Clusterが苦手とする範囲検索の影響のためスループットがのびない（？）

Hewlett Packard Enterprise

4. まとめ

まとめ

- ＪＲ東日本情報システム様がMySQL Clusterを導入した背景
  - 新しい技術へのチャレンジ：OSSの採用
  - データの分散配置、高可用性を備えたDBMS：MySQL Cluster
- MySQL Clusterの特徴
  - インメモリ、高速、高可用性、スケーラビリティ
  - 管理ノード、SQLノード、データノードから構成
- MySQL Cluster 検証結果
  - CPU性能、ネットワーク帯域、その後にディスク性能
  - SQLノードとデータノードを適切に配置することでスケール

Hewlett Packard
Enterprise

Thank you

ご清聴ありがとうございました

※パラメータ解説や本日割愛した検証結果などを記載した詳細版は、弊社ブログサイトからダウンロード可能です。
短縮URL：http://goo.gl/EhEZI4

Appendix 1. MySQL Cluster概要補足

MySQL Cluster略歴

- もともとはNetwork DataBase（NDB）と呼ばれていた携帯通信網用データベースで、以下のような特徴があった
  - インメモリで高速
  - 比較的シンプルな処理向き（SQLは使用できなかった）
  - 加入者増加にあわせた拡張性
  - 高可用性
- MySQLと統合されSQLによるアクセスも可能となった
- バージョンが進むにつれ、複雑なSQLの高速化や、一般的なRDBMSが備えている機能を実装（現在の最新版は7.4）
  - v7.0でデータノードのマルチスレッド化
  - v7.2、7.3でjoinの性能が大幅に改善
  - v7.3で外部キーをサポート

Hewlett Packard Enterprise

MySQL Clusterのパラメータ(1/4)

CPU関連パラメータ

- MaxNoOfExecutionThreadsパラメータでデータノードで動作するプロセス(ndbmtd)のスレッド数を指定することが可能
  例

```
MaxNoOfExecutionThreads = 8
```

- さらにThreadConfigパラメータを使用すると、役割の異なるスレッドに割り当てるCPUを細かく設定できる
  例

```
ThreadConfig=main={cpubind=0},ldm={count=4,cpubind=1,2,5,6},io={cpubind=3}
```

MySQL Clusterのパラメータ(2/4)

レプリカ数、チェックポイント関連パラメータ

- レプリカ数
  - NoOfReplicas：デフォルト2(1 ～ 4)

- チェックポイント関連
  - DiskCheckpointSpeed
  - TimeBetweenLocalCheckpoints
  - TimeBetweenGlobalCheckpoints
  - TimeBetweenEpochsTimeout

後述

MySQL Clusterのパラメータ(3/4)

オブジェクト数関連パラメータ

- MaxNoOfTables:デフォルト128(8 ～ 20320)
- MaxNoOfOrderedIndexes:デフォルト128(0 ～ 4294967039)
- MaxNoOfUniqueHashIndexes:デフォルト64(0 ～ 4294967039)

MySQL Clusterのパラメータ(4/4)

同時実行数関連パラメータ

- MaxNoOfConcurrentScansパラメータの最大値が500に注意
  - MaxNoOfConcurrentScansパラメータはデータノード毎の同時実行可能なテーブルスキャンまたはインデックススキャンの数
  - この値を超える「ALERT: MySQL error: Got temporary error 488 'Too many active scans' from NDBCLUSTER」というエラーが発生。
  - 最大の500に設定してもエラーが出る場合は、データノードの増設が必要

- その他に、MaxNoOfConcurrentTransactions、MaxNoOfConcurrentOperationsなど。

Hewlett Packard Enterprise

データの永続化(1/2)

LCPとGCPによるデータ永続化

- LCP(Local Check Point)
  - LCPはある時点における全データをディスクに書き出す処理
  - TimeBetweenLocalCheckpointsパラメータで指定した間隔ごとに行われる(デフォルトは20(4MB))
  - DiskCheckpointSpeedパラメータでディスクへの書き込み速度を制御(デフォルトは10MB/s)
  - ディスクテーブルのデータは更新されたデータが全てディスクにフラッシュされる(いわゆるチェックポイント)

データの永続化(2/2)

LCPとGCPによるデータ永続化

- GCP(Global Check Point)
  - 更新された内容をRedoログへ出力する処理
  - TimeBetweenGlobalCheckpointsパラメータで指定した間隔ごとに行われる(デフォルトは2秒)
  - GCPよりも高い頻度で、データノード間の同期処理を行うmicro-GCPが実行される。
  - micro-GCPがTimeBetweenEpochsTimeoutパラメータで設定された時間以内に完了しないとデータノードがクラスタから切り離される。v7.1までのデフォルト値は4秒。7.2以降は0(タイムアウトを検知しない)となっている。

ディスクテーブル

ディスクにデータを格納する機能とその注意点

- ディスクにデータを格納する機能
- 注意点
  - メモリテーブルと比較すると遅い
  - メモリ上にメモリテーブルとは別のバッファ領域が必要である。そのためディスクテーブルの性能を向上させるために、バッファ領域を大きくすると、メモリテーブル用の領域が少なくなってしまう。
- BLOB型、TEXT型のようなサイズの大きなデータを格納するために使用するケースが多い

MySQL Clusterの主なメモリ領域

No	メモリ領域	主な用途
1	DataMemory	メモリテーブルのデータを格納。Ordered （B-Tree ） インデックスのデータもこの領域に格納される。 ディスクテーブルのインデックスとインデックスを作成した列も格納される
2	IndexMemory	ハッシュインデックスのデータを格納
3	RedoBuffer	Redoログに出力するデータのバッファ領域。COMMITされるまでの変更は全てこの領域に保持する必要がある
4	SharedGlobalMemory	外部キーのチェックなどの各種操作で使用するメモリ領域。ディスクテーブルを使用する場合、この領域にUNDOバッファが作成される
5	DiskPageBufferMemory	ディスクテーブルデータ用のキャッシュ領域

Hewlett Packard Enterprise

Appendix 2. MySQL Cluster検証補足

sysbenchで実行されるトランザクション

テーブル構造

No	列名	データ型	備考
1	id	integer	PK
2	k	integer	
3	c	char(120)	
4	pad	char(60)	

更新SQL
Read Onlyでは実行されない

sysbenchは検証1,2,4で使用

トランザクション

PKによる単純な検索

```
SELECT c FROM sbtest WHERE id = :1;    × 10回

SELECT c FROM sbtest WHERE id BETWEEN :1 AND :2;
SELECT SUM(k) FROM sbtest WHERE id BETWEEN :1
AND :2;
SELECT c FROM sbtest WHERE id BETWEEN :1 AND :2
ORDER BY c;
SELECT DISTINCT c FROM sbtest WHERE id BETWEEN :1
AND :2 ORDER BY c;
```

範囲検索およびソート

```
UPDATE sbtest SET k = k + 1 WHERE id = :1;
UPDATE sbtest SET c = :1 WHERE id = :2;
DELETE FROM sbtest WHERE id = :1;
INSERT INTO sbtest (id, k, c, pad) VALUES (:1, :2, :3, :4);
COMMIT;
```

Hewlett Packard Enterprise

検証環境MySQL Cluster主要パラメータ設定

- データノード用パラメータ（config.ini）

パラメータ	設定値	備考
DataMemory	20G	
IndexMemory	4G	
DiskPageBufferMemory	64M	デフォルト
RedoBuffer	64M	
MaxNoOfExecutionThreads	8	

- SQLノード用パラメータ

パラメータ	設定値	備考
ndb-cluster-connection-pool	8	
max_connections	1000	

Hewlett Packard Enterprise

検証3 単一SQL処理時間性能比較

- 以下の条件のクエリの実行時間をMySQL ClusterとMySQL（InnoDB）で比較
  ① プライマリーキーによる1レコード取得
  ② インデックスによる複数レコード取得
  ③ インデックスによる複数レコード取得 + ソート
  ④ 非インデックス列への中間一致
- クエリ用テーブルは以下のような単純なテーブル

	列名	列定義	備考
1	col1	char(8)	primary key
2	col2	char(8)	index
3	col3	char(200)	
4	col4	char(200)	

- レコード数は100万行（3-1）と1億行（3-2）の2パターンで測定（MySQLのInnoDBバッファプールに全データがのるケースとのらないケースを想定）

Hewlett Packard Enterprise

検証3-1（ 100万行テーブル検索）結果

No	クエリタイプ	取得行数	処理時間（ミリ秒）		性能比
			MySQL Cluster	MySQL	
1	プライマリーキーによる1レコード取得	1	0.65	0.25	0.39
2	プライマリキー以外のインデックスによる複数レコード取得	10,000	149.76	71.30	0.48
3	プライマリキー以外のインデックスによる複数レコード取得 + ソート	10,000	3,369.47	75.61	0.02
4	非キー項目の中間一致検索	15,741	396.36	3,087.59	7.79

検証3-2（ 1億行テーブル検索）結果

No	クエリタイプ	取得行数	処理時間（ミリ秒）		性能比
			MySQL Cluster	MySQL	
1	プライマリーキーによる1レコード取得	1	0.67	21.04	31.45
2	プライマリキー以外のインデックスによる複数レコード取得	1,000,000	14,764.93	467,372.91	31.65
3	プライマリキー以外のインデックスによる複数レコード取得 ＋ ソート	1,000,000	279,945.79	467,381.68	1.67
4	非キー項目の中間一致検索	1,574,101	27,955.42	480,301.26	17.18

検証4 テーブル配置種別による性能比較

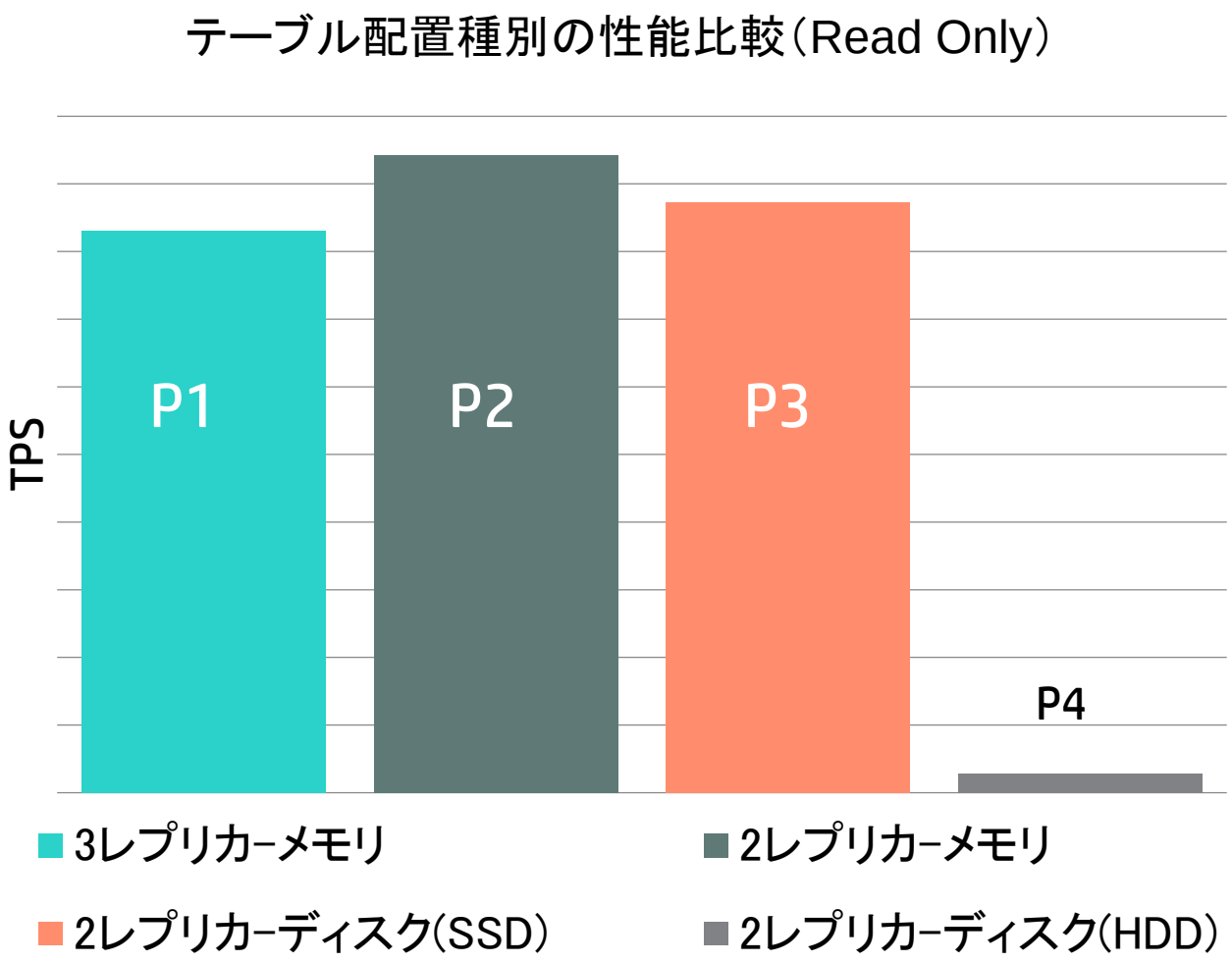
- 下記パターンの性能測定を実施し、メモリテーブル、ディスクテーブル（SSD）、ディスクテーブル（HDD）の性能を比較

構成No	SQLノード数	データノード数	レプリカ数	テーブル	ストレージ
P1	3	6	3	メモリ	HDD
P2	3	4	2	メモリ	HDD
P3	3	4	2	ディスク	SSD
P4	3	4	2	ディスク	HDD

- sysbench OLTP complexモードを使用してスループット（TPS）を測定
- レコード数は1000万
- それぞれの構成についてRead Only（4-1）とRead Write（4-2）の2ケースを測定

Hewlett Packard Enterprise

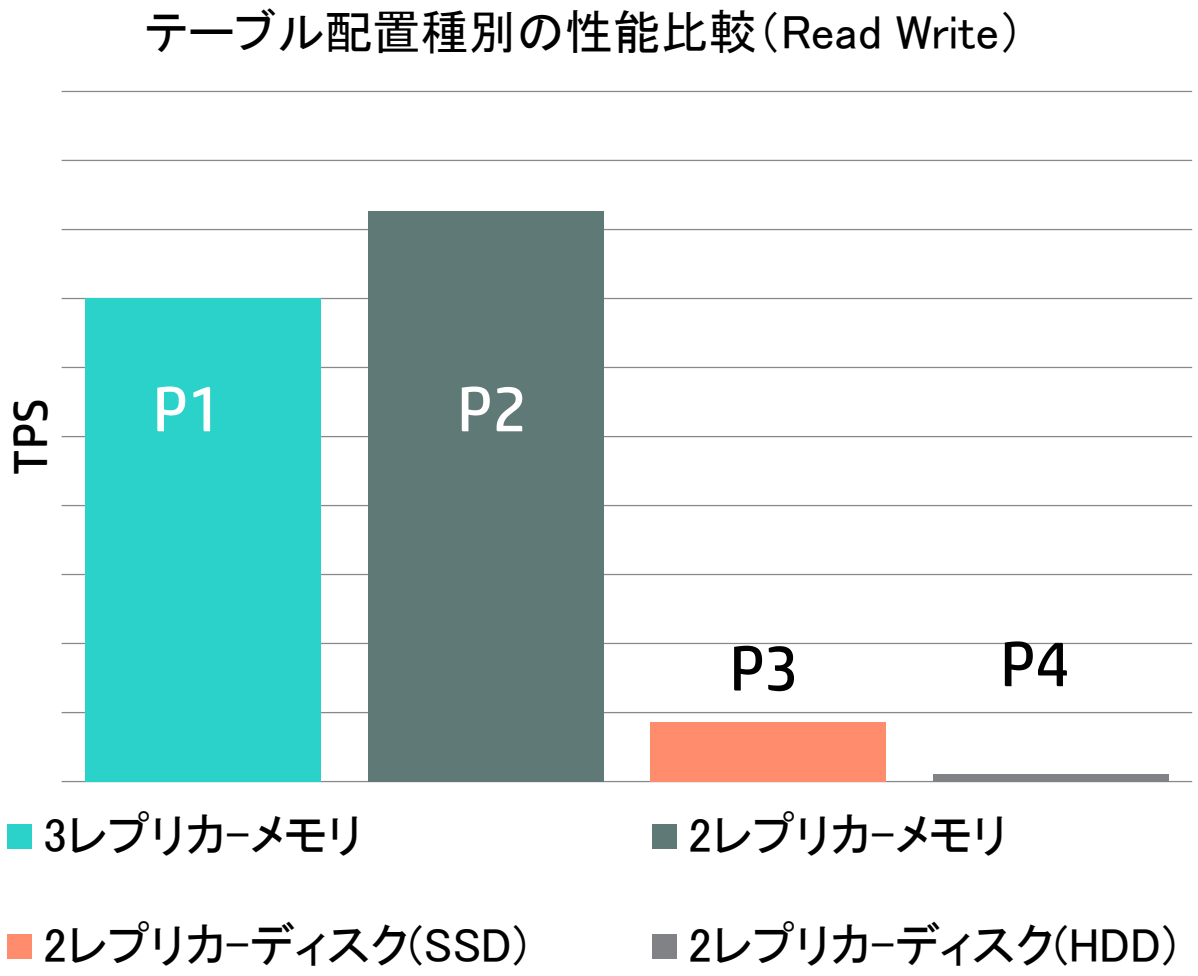
検証4-1（Read Only）結果



- Read OnlyではSSDを使用したディスクテーブルはメモリテーブルとほぼ同等の性能（環境によって異なるので注意）
- HDDのディスクテーブルの性能は極端に低い

Hewlett Packard Enterprise

検証4-2（Read Write）結果



- Read WriteではSSDを使用したディスクテーブルの性能も劣化
- SSDの性能特性にも影響されると考えられる。（使用したSSDのWrite性能はそれほど高性能ではなかった）
  - ランダムRead：64k IOPS
  - ランダムWrite：8k IOPS

Hewlett Packard Enterprise