LifeKeeper＋InfiniBand ＋IOアクセラレータを活用した PostgreSQLの HA・DR構成検証レポート

日本HP　テクノロジーサービス事業統括
テクノロジーコンサルティング統括本部
古井 佑治
yuji.furui@hp.com
2012/02/24

目次

- 本レポートの目的
- LifeKeeperとは
- 検証構成概要
- 検証内容概要
- 検証結果
- 考察
- まとめ
- Appendix

本レポートの目的

本レポートでは、PostgreSQLとLifeKeeperにおける以下の2つの検証結果を発表致します。

1. LifeKeeperのレプリケーション機能を利用したPostgreSQLのDR構成の機能・性能検証

2. PostgreSQL、LifeKeeper、InfiniBand、IOアクセラレータを組み合わせた、低レイテンシー高性能PostgreSQL HA構成のパフォーマンスおよび障害試験

ストレージ機能やストリーミングレプリケーションによるDR構成ではなく、LifeKeeperを使用することによるメリット・デメリットも考察します。

LifeKeeper概要

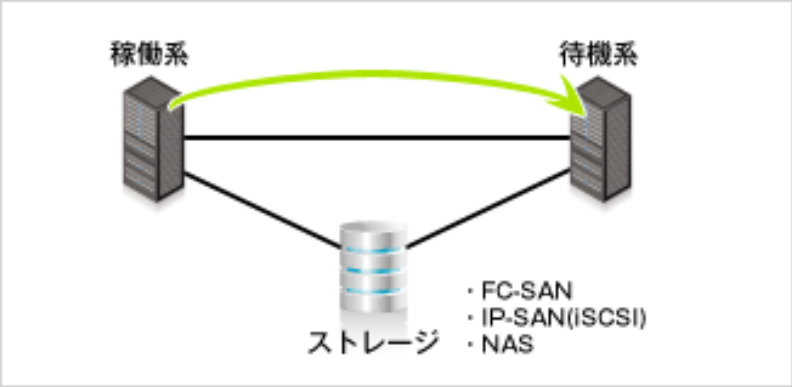
LifeKeeperは、障害時に業務を引き継ぐHAクラスター機能、データをリアルタイムに複製するデータレプリケーション機能により、ITシステムにおける事業継続を実現します。

LifeKeeperの特徴

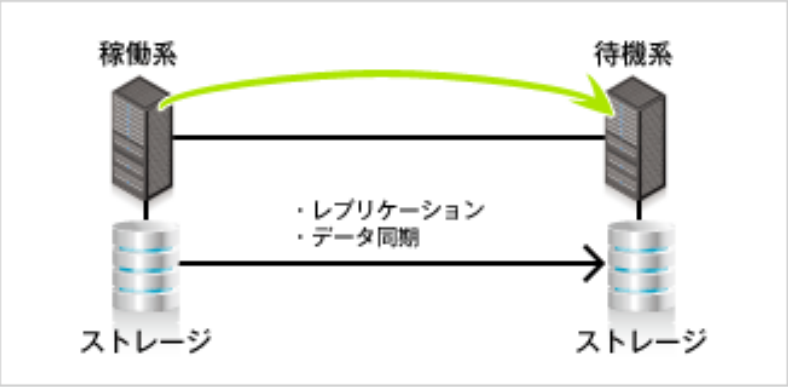
1．簡単導入
2．スムーズな運用管理
3．多様なシステム環境への対応

容易なインストール、スクリプトレスでHA化
GUIにより直感的な操作
多彩なアプリケーション、仮想環境に対応

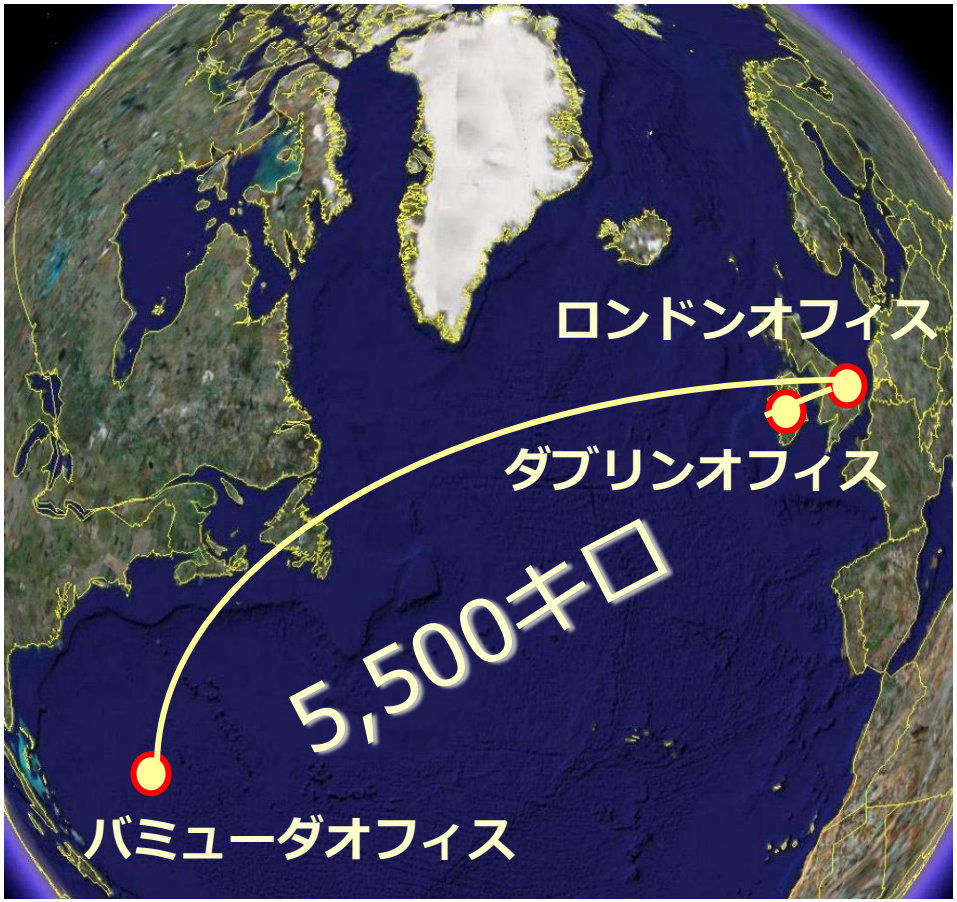
共有ディスク構成



データレプリケーション構成



事例①　5,500kmを越える遠隔地災害対策
～New Star社様導入事例～



■　背景

New Star社：英国資産運用業界の主要ブランド
Microsoft Exchangeを採用しているが、過去のシステムダウンの経験から障害時の完全復旧までの日数、復旧遅延を懸念

■　導入ステップ

ローカル環境でのレプリケーション性能検証、
他社レプリケーション製品との比較を実施
製品価格、対応アプリケーションの豊富さ、
導入の容易さ、N：1 構成が可能な点がポイントとなり、LifeKeeper の導入を決定

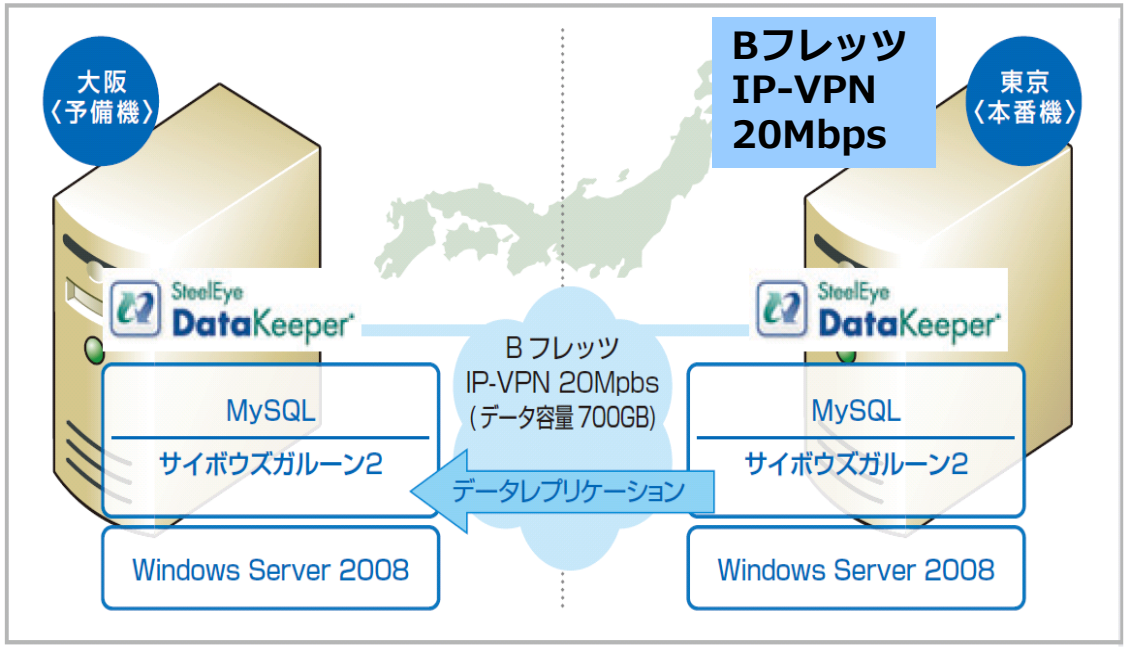
■　導入効果

LifeKeeper によって、ロンドン市内ナイツブリッジ（中核システム）、イーストロンドンのドックランズ、アイルランドのダブリン、バミューダの4拠点にて災害復旧対策を実現
専用線ではなく WAN を使用した低コストデータレプリケーション

Exchange Server の地球規模災害対策を実現

事例② グループウェア遠隔地レプリケーション
～パラカ様導入事例～

- 東京－大阪間でデータをレプリケーション
- 本番機から予備機への切替（データ容量700GB)が1時間未満で完了



低コスト導入

直近データでの復元

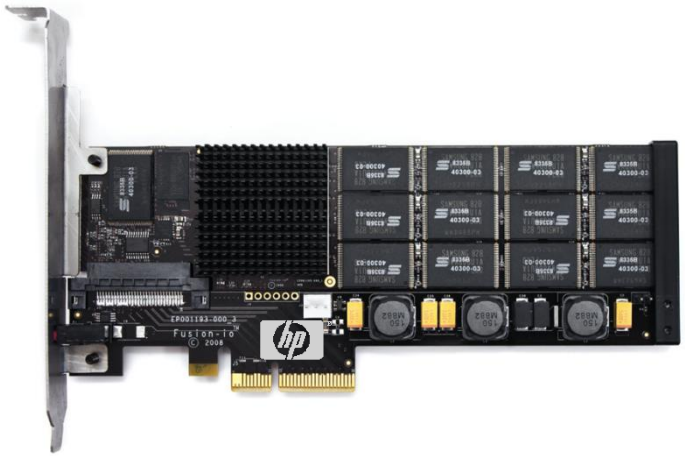
迅速な復旧

グループウェア規模拡大に合わせた
データベースの災害対策を低コストで実現

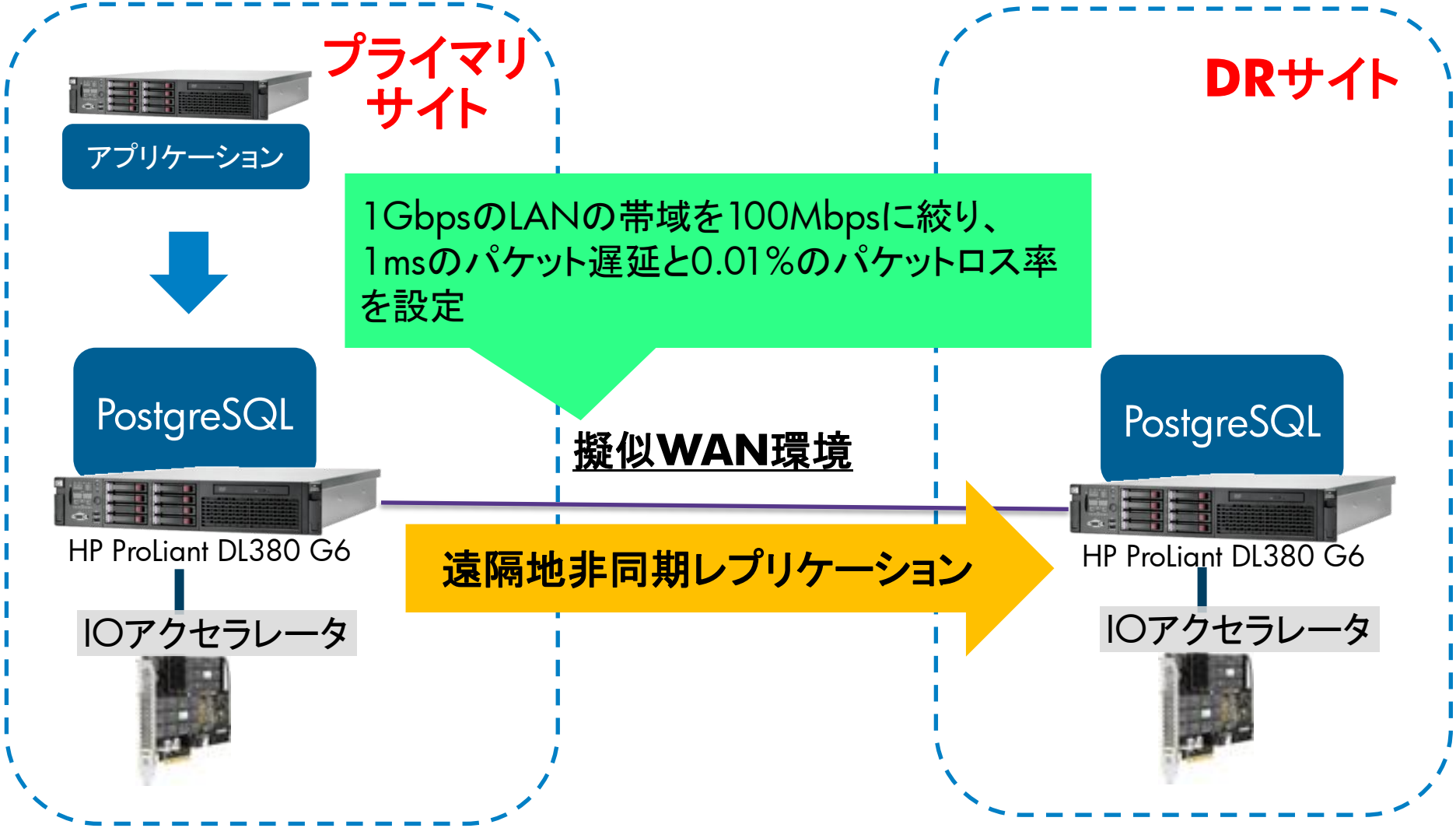
HP PCIe IOアクセラレータ

ウェブ系サービス企業で多く使用されている Fusion-io社のストレージ製品「ioDrive」のOEM製品

アプリケーションのパフォーマンスを向上させる、ソリッド ステート ストレージ テクノロジーを使用した直接接続型のPCIeカードベースのソリューション。マルチレベルセル（MLC）およびシングルレベルセル（SLC）NANDフラッシュテクノロジーに基づくこれらのデバイスは、高いトランザクションレートとリアルタイムデータアクセスを必要とする市場およびアプリケーションに最適です。



DR構成概要（擬似WAN環境非同期レプリケーション）



※ IOアクセラレータ：PCI直結の超高性能フラッシュメモリストレージ

擬似WAN環境について

RHELに同梱されているtc（traffic control）コマンドを利用し、LAN上に擬似WAN環境を構成し、実際の遅延やパケットロスをシミュレート！

•tcコマンドでは下記の制御が可能

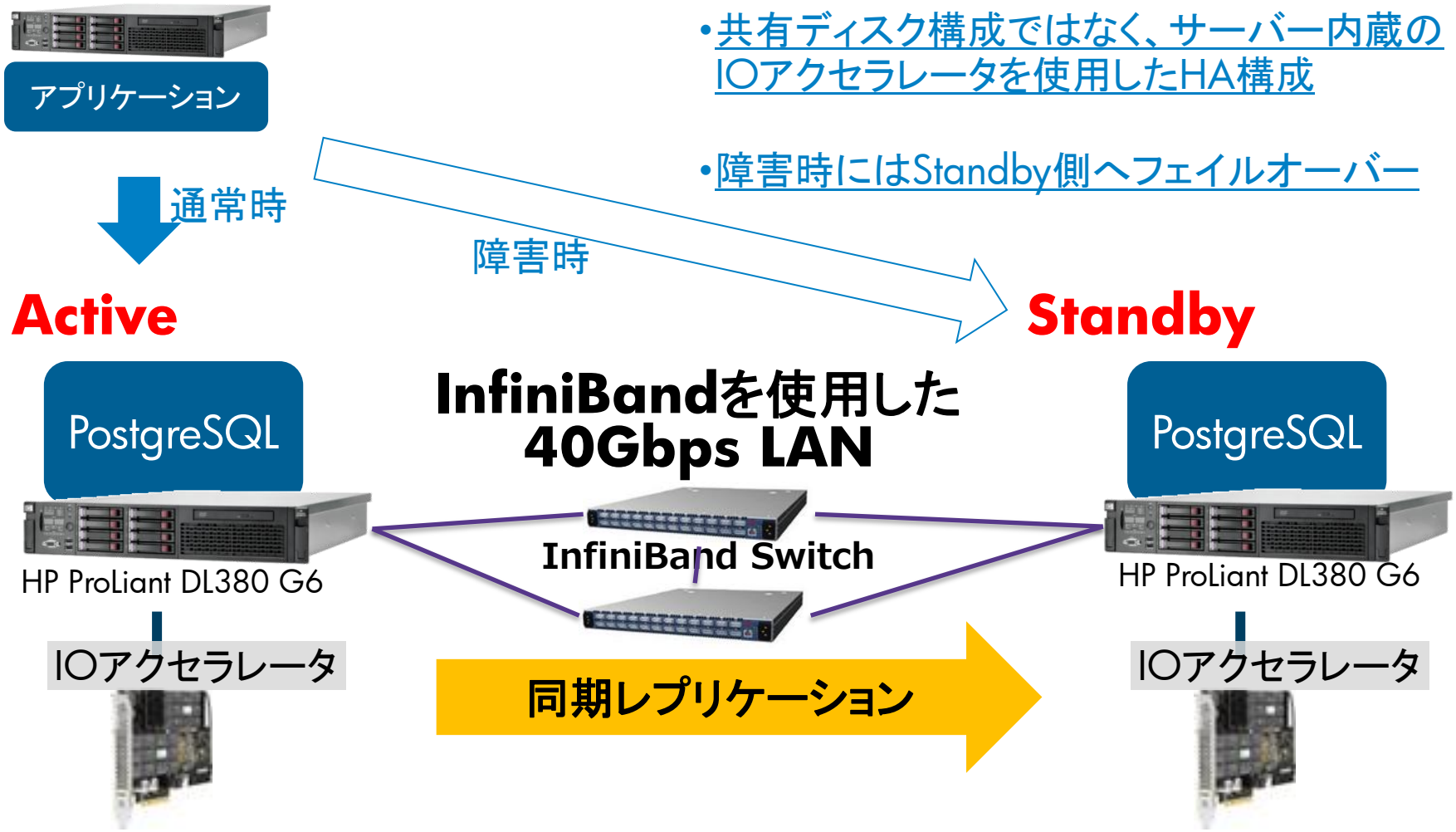
- •遅延の範囲
- •パケットロス
- •パケット破壊
- •パケット重複
- •パケット再送要求
- •帯域制御

※遅延の範囲、パケットロス、帯域制御の3つを設定することで、擬似WAN環境を作成可能！

•下記RPMに同梱

- •iproute

HA構成概要（LAN環境同期レプリケーション）



- 共有ディスク構成ではなく、サーバー内蔵のIOアクセラレータを使用したHA構成
- 障害時にはStandby側へフェイルオーバー

検証内容概要

•検証項目

DR構成とHA構成のそれぞれの構成で、PostgreSQLに対してOLTPベンチマークと障害試験

•環境

データベースサーバーにはRHEL5.7を使用

Postgres Plus 9.0.4を使用（Postgres Plus固有の機能は未使用）

IOアクセラレータ は、OSで認識する2つのデバイスをLVMでストライピング

•OLTPベンチマーク用の負荷テスト

jdbcrunnerを採用（RDBMSを対象とした負荷テストツール）

TPC-Cを簡易実装したトランザクションを実行

Write：Read = 9：1 の割合でトランザクションを実行

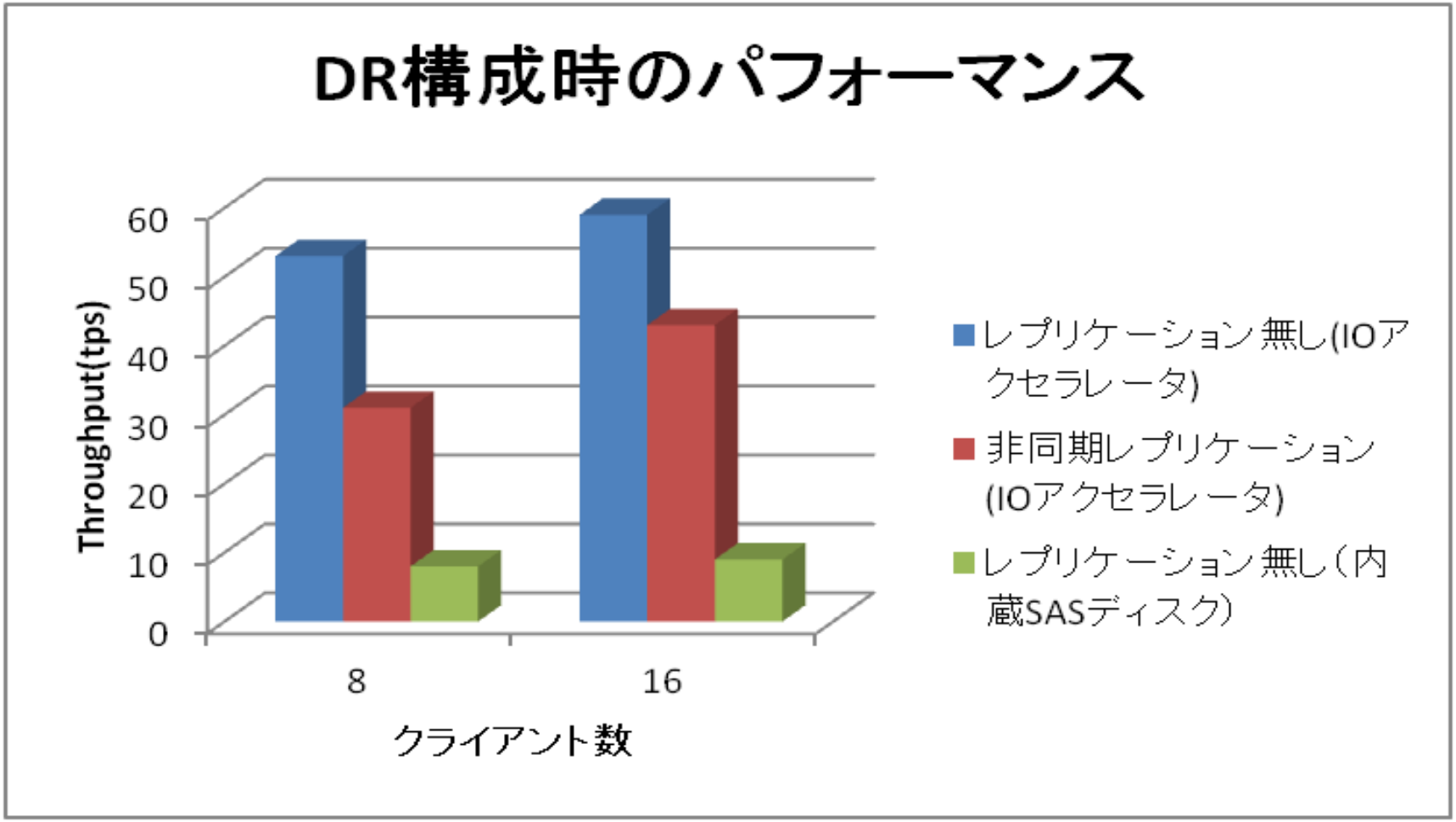
テスト用のテーブルとして、約20GBのテーブルを用意

各テストを3回ずつ実施し、平均値を結果として使用

※TPC-C とは、TPC によって策定されたOLTPベンチマーク仕様の一つ。卸売業における注文・支払いなどの業務をモデルにしたトランザクションを実行し、システムの性能を測定

検証結果

DR構成における性能比較

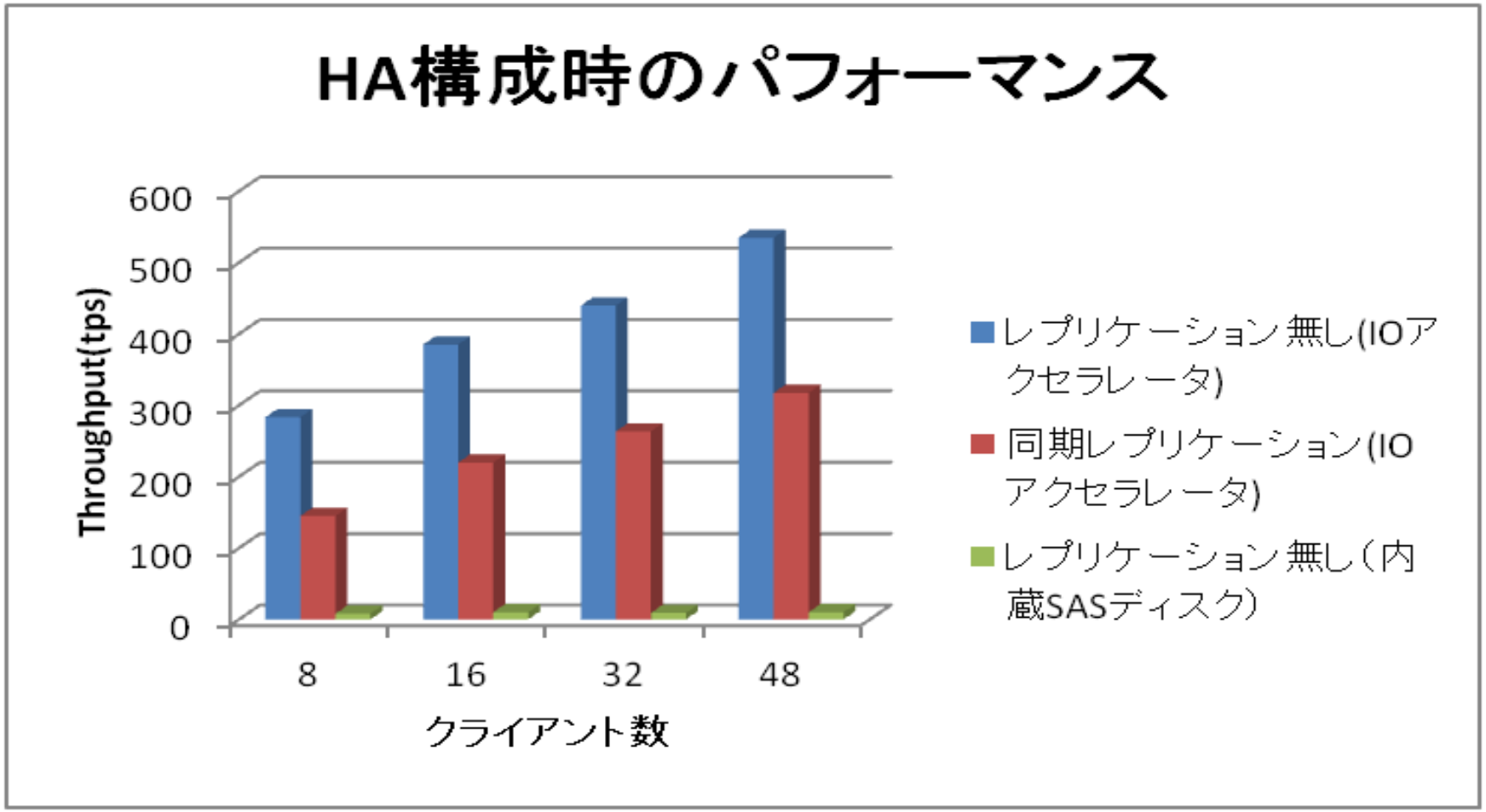


※一般的なアプリケーションからのアクセス負荷をシミュレートするため、
トランザクションごとのSleep時間を50msecに設定

障害試験結果（DR構成）

検証項目	内容	期待する挙動	結果
パス障害（レプリケーション用のセグメント）と復旧	DBへのアクセス中（ベンチマークツール実行中）に、レプリケーション用のLANケーブルを抜線	LifeKeeperはレプリケーションパスが切断した事を認識。同期を中断し、DB処理の継続。復旧後、同期の再開	○
レプリケーション再同期時間	レプリケーションを一時停止し、約20GBのファイル作成後、レプリケーション再開してから再同期が完了するまでの時間の測定	約30分で再同期完了	○
DBサーバー障害（カーネルパニック）	DBへのアクセス中（ベンチマークツール実行中）に、プライマリDBサーバー側をカーネルパニック	設定通り一旦サービス停止。10分間（既定値）手動フェイルオーバーのオペレーションがなければ、セカンダリサイトに自動フェイルオーバー	○
PostgreSQL障害	postgresのプロセスをkill	デフォルトでは、ローカルリカバリーが走り、自ノードで再起動。それにも失敗する場合は、セカンダリサイトに手動フェイルオーバー	○

HA構成における性能比較



※トランザクションごとのSleep時間を設定せず、継続的な激しい負荷の為、
DR構成に比べて高いスループットとなっている

障害試験結果(HA構成)

検証項目	内容	期待する挙動	結果
パス障害(レプリケーション用のセグメント)と復旧	DBへのアクセス中(ベンチマークツール実行中)に、レプリケーション用のLANケーブルを抜線	LifeKeeperはレプリケーションパスが切断した事を認識。同期を中断し、DB処理の継続。復旧後、同期の再開	○
レプリケーション再同期時間	レプリケーションを一時停止し、約20GBのデータ作成後、レプリケーション再開してから再同期が完了するまでの時間の測定	約2分で再同期完了	○
DBサーバー障害(カーネルパニック)	DBへのアクセス中(ベンチマークツール実行中)に、プライマリDBサーバー側をカーネルパニック	セカンダリDBサーバーへ正常に自動フェイルオーバー	○
PostgreSQL障害	postgresのプロセスをkill	デフォルトでは、ローカルリカバリーが走り、自ノードで再起動。それにも失敗する場合は、セカンダリDBサーバーに自動フェイルオーバー	○

考察

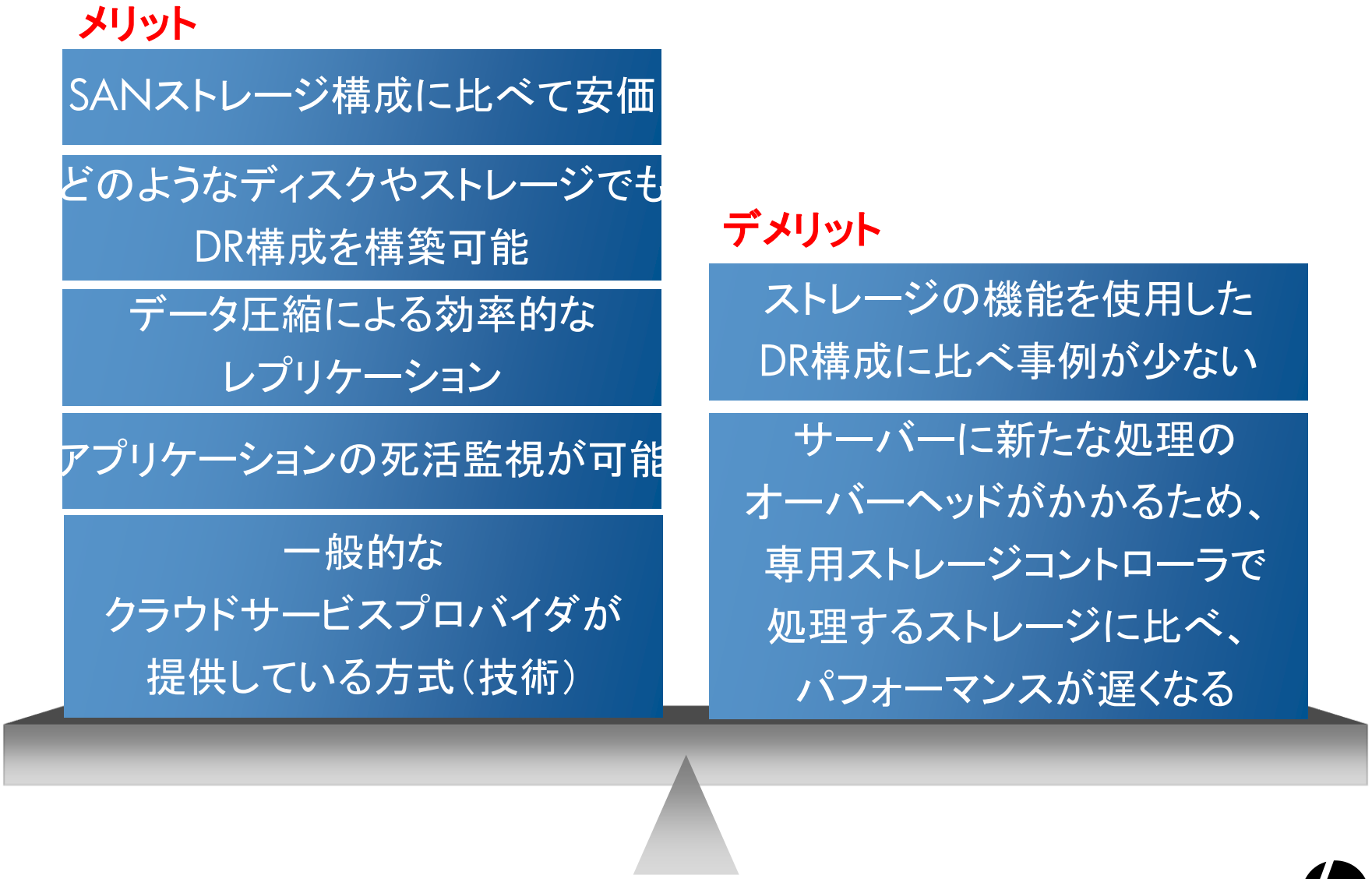
LifeKeeperを使用したDR構成考察

•DRに必要となる機能の動作確認:
SIOS社との共同検証により、DRに必要となるWAN越しでのレプリケーション機能が実用的な性能で動作することを確認

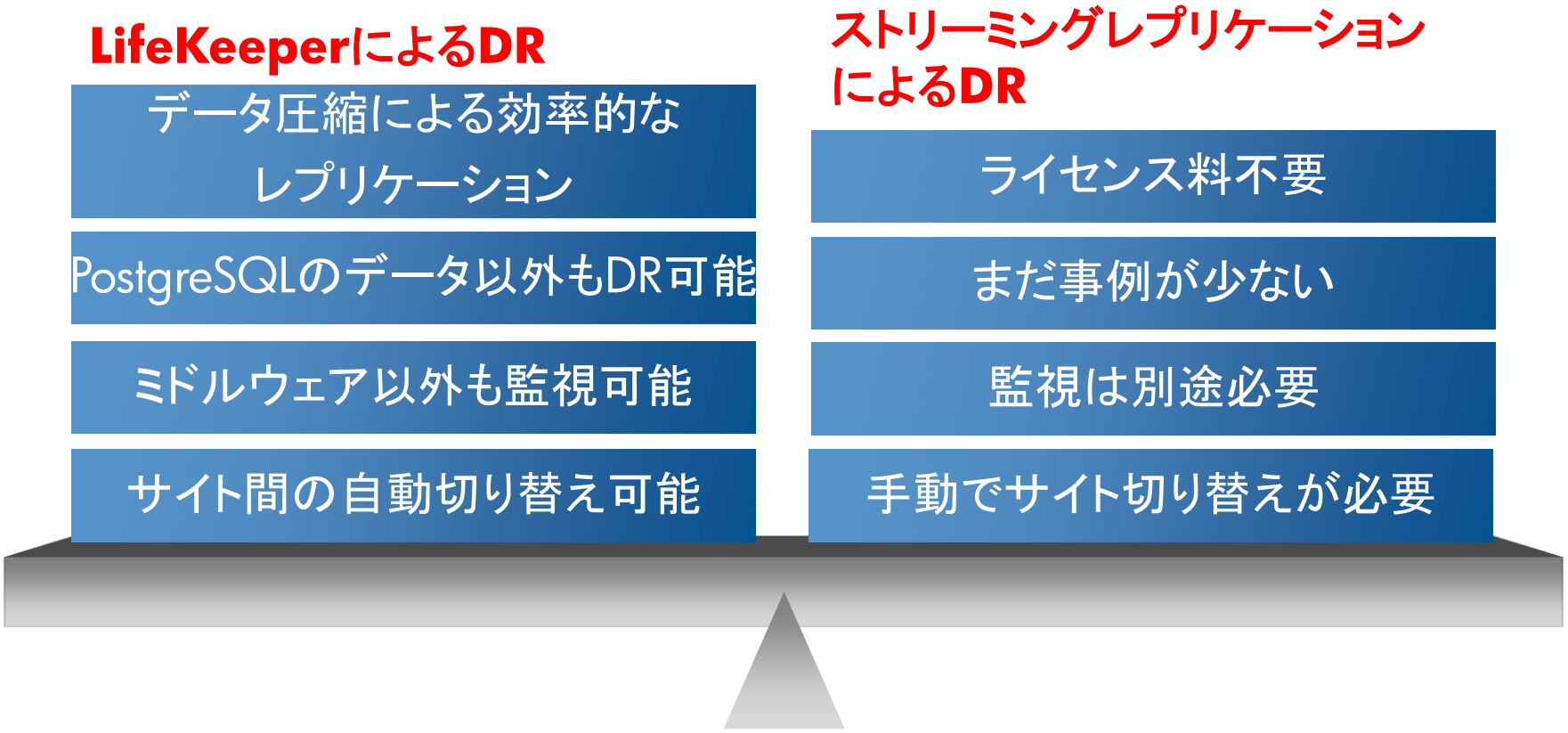
•Write性能への考慮必要:
WAN越しの非同期レプリケーションは、サーバーに対して新たな処理を実施させること、また、I/O処理量やWANの帯域によりパフォーマンスに影響がある。そのため、どれくらいのWrite性能を期待するかを設計した上で検討が必要

•PostgreSQLの監視も可能:
SANストレージによるDR構成の場合、ストレージの機能ではミドルウェアの監視をすることができないが、LifeKeeperであれば、ミドルウェアを含めた監視が可能

LifeKeeperでDR構成を構築するメリット・デメリット



DR構成におけるストリーミングレプリケーションとの比較



InfiniBand+IOアクセラレータによる超高速PostgreSQL構成でのLifeKeeperによるHA構成の考察

•HA機能の動作確認:
SIOS社との共同検証により、HA機能が想定通りに動作することを確認

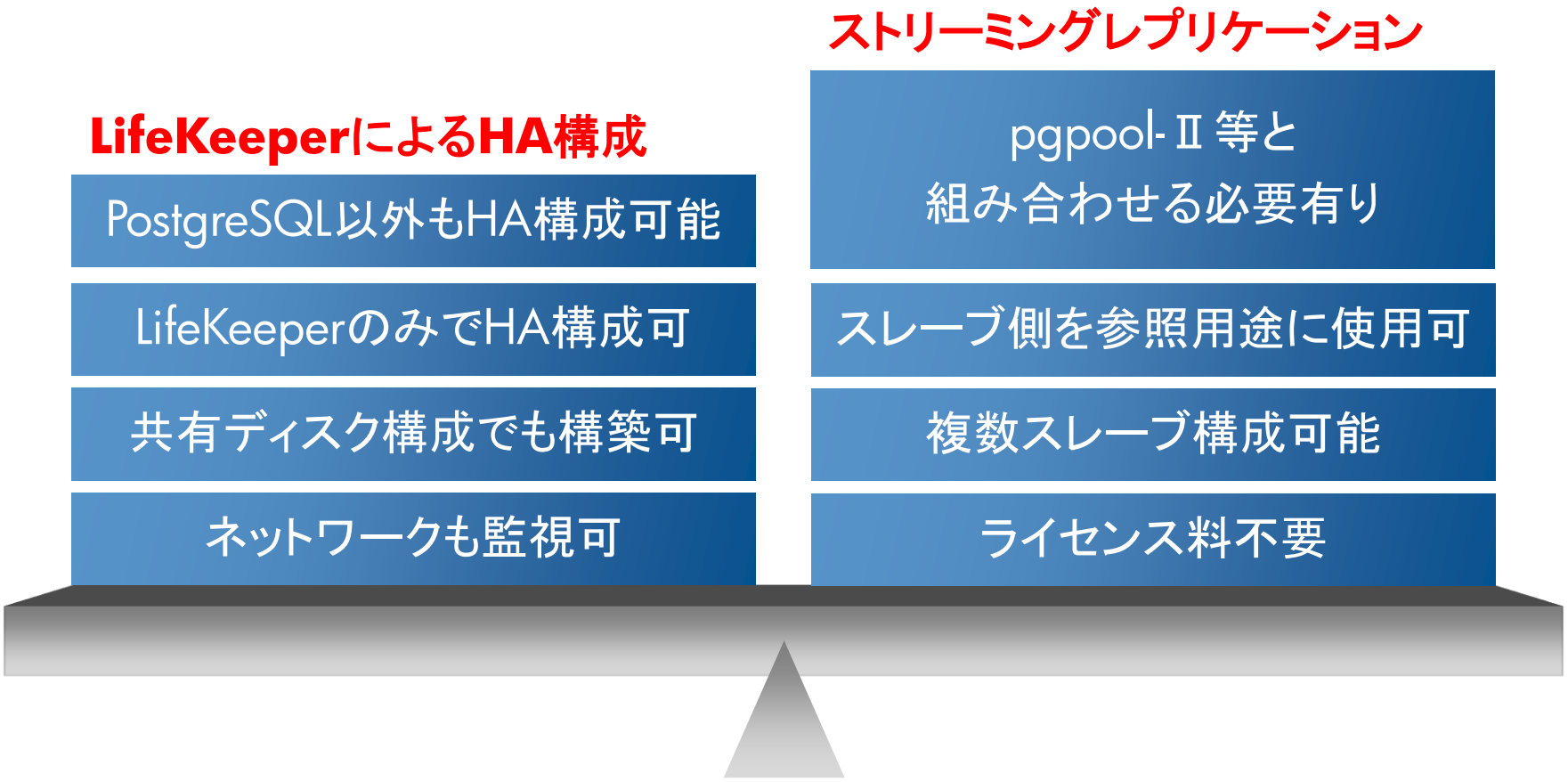
•IOアクセラレータによる高速化:
レプリケーション無しの内蔵SASディスク構成に比べて、レプリケーション無しの内蔵IOアクセラレータ構成は約35～53倍のパフォーマンス！

•Write性能への考慮必要:
同期レプリケーションは、ローカルとミラー先の2ヶ所に書き込みを実施するためにWrite性能は低下する。また、アプリケーションごとにWrite量が違う。そのため、発生するパフォーマンス低下分を考慮したシステム設計・運用設計をする必要有り

•IOアクセラレータ+InfiniBandによる高速レプリケーション:
レプリケーション無しの内蔵SASディスク構成に比べて、レプリケーション有りの内蔵IOアクセラレータ構成は約18～31倍のパフォーマンス！！

IOアクセラレータとInfiniBandを使用することで、
同期レプリケーションの性能劣化を充分に補うことが可能！！

HA構成におけるストリーミングレプリケーションとの比較



まとめ

- **LifeKeeperを利用することで高価なSANストレージを使用せずにDR構築可能！**
  - 日本HPとSIOSで十分な検証済み
  - 事例も豊富で今後も増えてくる可能性大

- **LifeKeeper + InfiniBand + IOアクセラレータで冗長性を持った低レイテンシー高性能なPostgreSQLを構築可能！**

- **LifeKeeperを利用するかストリーミングレプリケーションのどちらを使用するかは、それぞれのメリット・デメリットを考慮する必要有り**

本ソリューションについて

HP ProLiant + LifeKeeper + InfiniBand
+ IOアクセラレータを活用した
PostgreSQL
HA・DR ソリューションを提供開始

日本HPは、本ソリューションの販売を開始します。

Appendix

参考:レプリケーションシステム構築事例

お客様	システム	接続	データ容量	使用ソフトウェア
英国の資産運用会社New Star社	Exchange サーバー	ロンドン - アイルランド - バミューダ（5500km）	非公開	LifeKeeper for Windows with SDR Option
某製造業会社様	ファイル	数百メートル	（非公開：数十ギガバイト）	LifeKeeper for Windows with SDR Option
コインパーキング管理会社様	グループウェア	東京-大阪	数十ギガ～数百ギガ	DataKeeper for Windows – Standard Edition
某製造業会社様	メールサーバー	長崎-福岡	（非公開：数十ギガバイト）	LifeKeeper for Linux with SDR Option
某半導体検査装置開発企業様	ファイルサーバー	ローカル接続	2テラバイト	DataKeeper for Windows – Standard Edition
三重県某市役所様	グループウェア	ローカル接続	900ギガバイト	LifeKeeper for Linux with SDR Option
某建築会社様	グループウェア	ローカル接続	3テラバイト	LifeKeeper for Linux with SDR Option
某物流会社様	データベース	ローカル接続	60ギガバイト	DataKeeper for Windows – Standard Edition

検証環境詳細

サーバー	HP ProLiant DL380 G6（データベースサーバー）	HP ProLiant BL460cG6（クライアントサーバー）
OS	RHEL 5.7 x86_64	RHEL 5.5 x86_64
System ROM	P62 03/30/2010	I24 10/01/2009
CPUタイプ	Intel(R) Xeon(R) CPU E5540 2.53GHz	Intel(R) Xeon(R) CPU E5540 2.53GHz
CPU数	1P（4core） HT=on	1P（4core） HT=on
メモリ	6GB	6GB
InfiniBand HCA	HP IB 4X QDR CX-2 PCI-E G2 DUAL PORT HCA(MT26428)	-
OFED	MLNX_OFED_LINUX-1.5.3-30.0-rhel5.7-x86_64	-
内蔵ディスク	HP 146GB 6G SAS 10K 2.5in DP ENT HDD	HP 146GB 6G SAS 10K 2.5in DP ENT HDD
ディスク コントローラー	Smart アレイP410i/256MB コントローラ	Smart アレイP410i/256MB コントローラ
IO-A HW	HP 320GB SLC PCIe IOアクセラレータ Duo	-
IO-A firmware/driver	Firmware v101971/ driver 2.3.1	-
IO-Aフォーマット	Advertised Capacity	-
LifeKeeper	7.5	7.5

擬似WAN環境構築手順(1/4)

サーバーAとサーバーBの両方で実行します

- RPMの確認

```
# rpm -q iproute
iproute-2.6.32-16.el6.x86_64
```

- 規則をeth1に適用すると宣言

```
# tc qdisc add dev eth1 root handle 1:0 htb
```

tc qdisc add dev eth1
eth1にこのキューイング規則を適用

root handle 1:0
このインターフェイスのルートノードidを1:0に指定
このインターフェイスに作成されるキューは、このidをparentとして指定することになる

htb
キューイングアルゴリズムにHTB（Hierarchical Token Bucket）を指定

擬似WAN環境構築手順(2/4)

サーバーAとサーバーBの両方で実行します

- 100mbpsに帯域制限

```
# tc class add dev eth1 parent 1:0 classid 1:0 htb rate 100Mbit
```

parent 1:0
このキューをインターフェイス1:0に作成する

classid 1:0
このキューのidを1:0に指定

rate
帯域幅閾値を指定する

擬似WAN環境構築手順(3/4)

サーバーAとサーバーBの両方で実行します

- 帯域制限されていることの確認

```
# iperf -c 192.168.10.235 -i 10 -t 60
------------------------------------------------------------
Client connecting to 192.168.10.235, TCP port 5001
TCP window size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.10.234 port 40077 connected with 192.168.10.235 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec   135 MBytes   113 Mbits/sec
[  3] 10.0-20.0 sec   117 MBytes  98.4 Mbits/sec
[  3] 20.0-30.0 sec   123 MBytes   103 Mbits/sec
[  3] 30.0-40.0 sec   117 MBytes  98.0 Mbits/sec
[  3] 40.0-50.0 sec   122 MBytes   102 Mbits/sec
[  3] 50.0-60.0 sec   117 MBytes  98.0 Mbits/sec
[  3]  0.0-60.2 sec   731 MBytes   102 Mbits/sec
```


擬似WAN環境構築手順(4/4)

サーバーAとサーバーBの両方で実行します

- 遅延の範囲(15ms +/- 3ms)とパケットロス(0.1%)の設定

```
# tc qdisc add dev eth1 parent 1:0 netem delay 15ms 3ms loss 0.1%
```

- 遅延が起こることの確認
  - 通常時

```
# ping 192.168.10.235
PING 192.168.10.235 (192.168.10.235) 56(84) bytes of data.
64 bytes from 192.168.10.235: icmp_seq=1 ttl=64 time=0.184 ms
64 bytes from 192.168.10.235: icmp_seq=2 ttl=64 time=0.155 ms
```

  - 設定時

```
# ping 192.168.10.235
PING 192.168.10.235 (192.168.10.235) 56(84) bytes of data.
64 bytes from 192.168.10.235: icmp_seq=1 ttl=64 time=13.9 ms
64 bytes from 192.168.10.235: icmp_seq=2 ttl=64 time=15.2 ms
64 bytes from 192.168.10.235: icmp_seq=3 ttl=64 time=12.7 ms
64 bytes from 192.168.10.235: icmp_seq=4 ttl=64 time=17.1 ms
```


Thank you