

MySQL Clusterで高性能システムを 構築する際のポイント

住商情報システム株式会社, 廣濱 顕司

2008/10/31, @ MySQL ユーザカンファレンス 2008

自己紹介

- ▶ 2003年よりMySQL関連の業務を担当
 - ▶ 2003/8/4にLarry Stefonic氏とサンフランシスコでMTGしたのが全ての始まり
 - ▶ 立ち上げ時はMySQL社とのパートナーリングその他雑用全般を担当
 - ▶ 現在は以下のMySQL関連業務
 - ▶ コンサルティング
 - ▶ トレーニング講師
 - ▶ サポート
- ▶ 趣味
 - ▶ 旅行
 - ▶ ガジェット



Agenda

MySQL Clusterとは

MySQL ClusterへのSCSの取り組み

ベンチマーク

MySQL Clusterのシステム構成例

さいごに

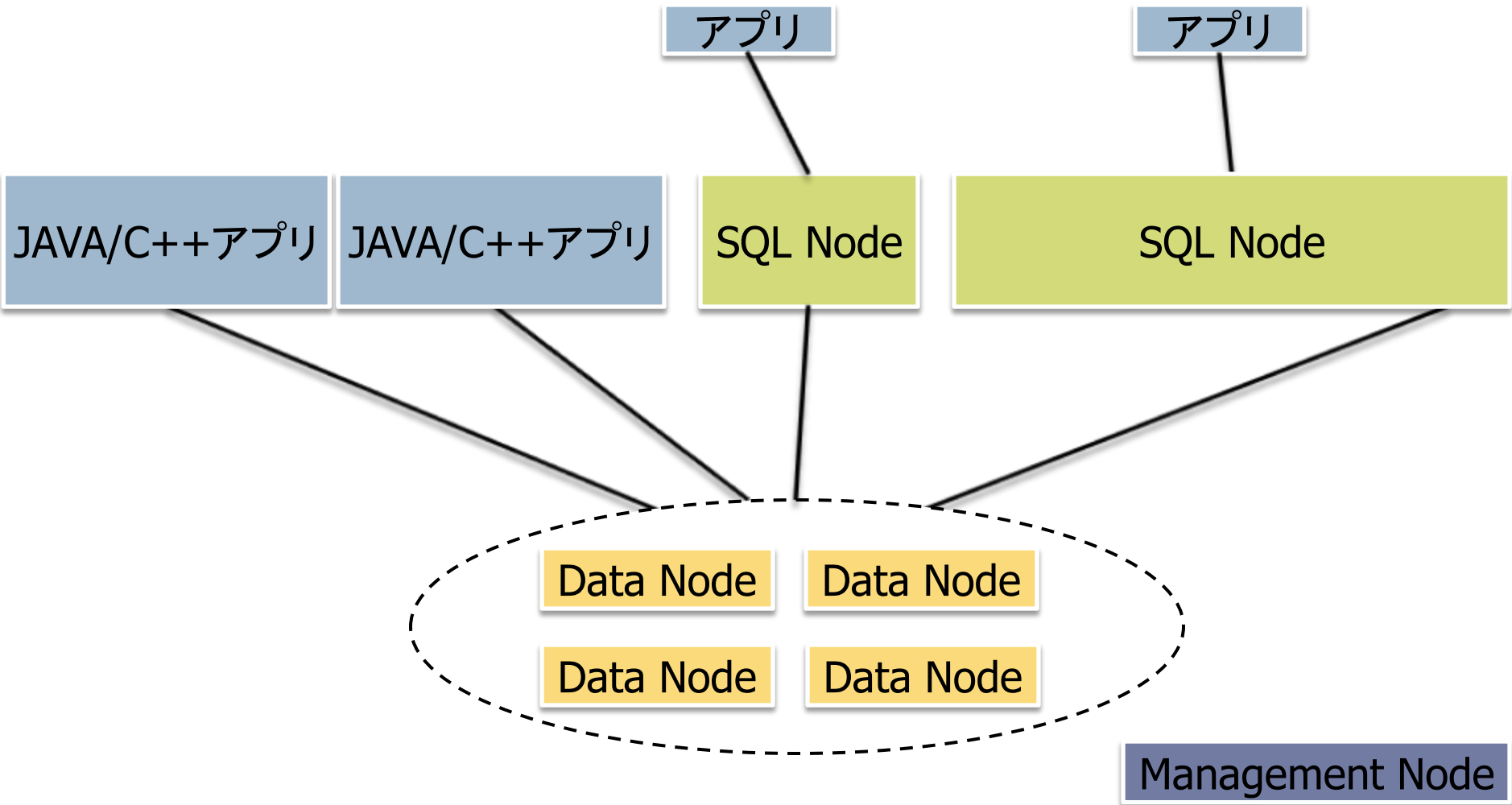
MySQL Clusterとは

MySQL Clusterとは

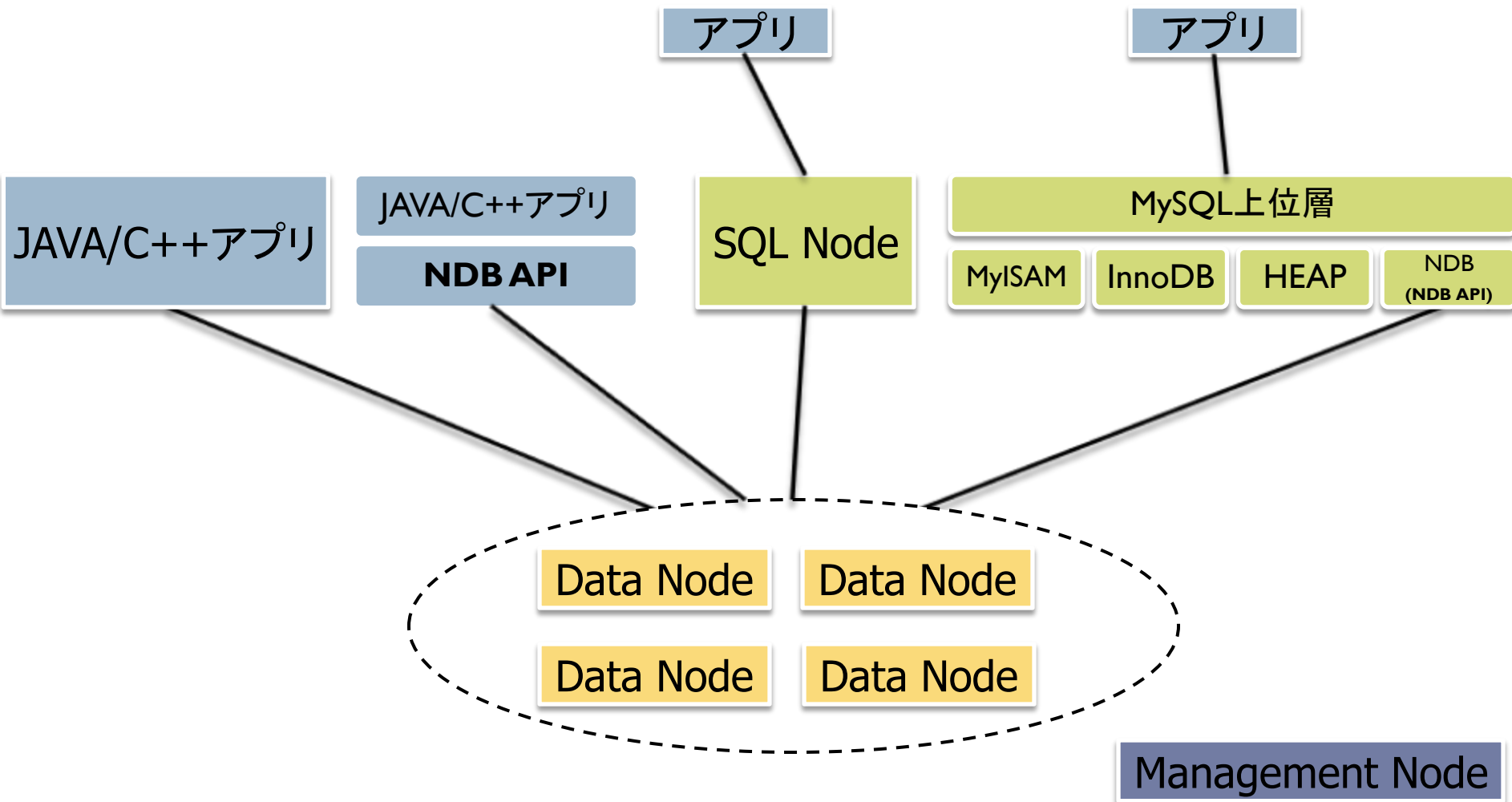
▶ 特徴

- ▶ 非共有ディスク型
 - ▶ 特殊なHWを必要としない
- ▶ アクティブ・アクティブ型
 - ▶ フェールオーバーの時間が非常に短い
- ▶ インメモリデータベース (5.1以降はディスクテーブルもサポート)
 - ▶ 高い性能
- ▶ 高い可用性
 - ▶ 各ノードの冗長構成が可能

MySQL Cluster概要図



MySQL Cluster詳細図



MySQL Clusterを利用するアプリケーション

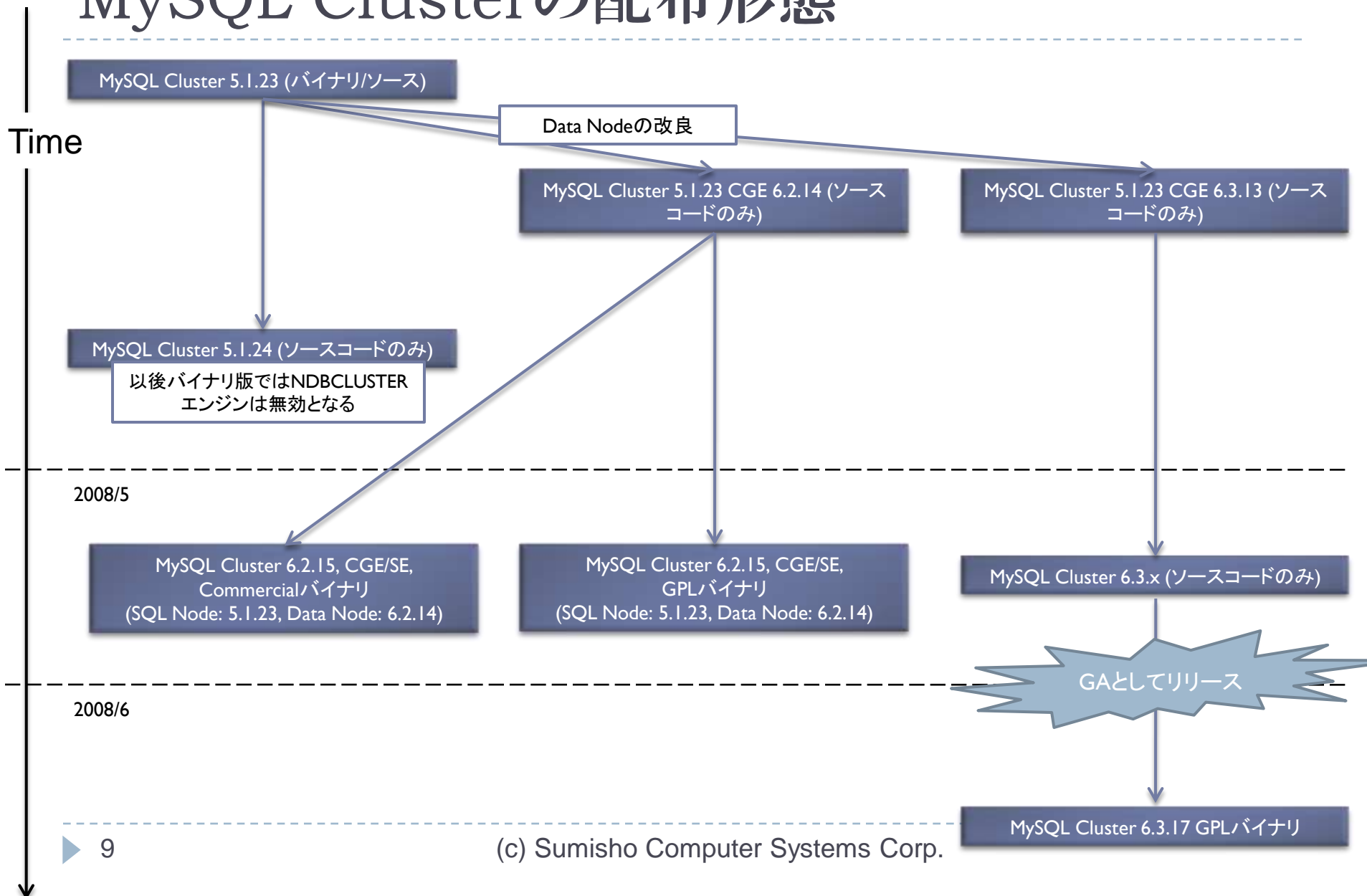
▶ SQL Node経由

- ▶ ストレージエンジンの一つとして簡単に利用できる
- ▶ MyISAM, InnoDBテーブルなどと混在できる
- ▶ MyISAM, InnoDBテーブルなどと同等に扱える
 - ▶ SQLクエリの発行など

▶ NDBアプリケーション(JAVA/C++アプリ)

- ▶ NDB APIを利用して直接Data NodeへアクセスするJAVAやC++で記述されたアプリケーション
- ▶ SQLクエリを発行することはできないが、非常に高速

MySQL Clusterの配布形態



MySQL Clusterエディションの違い

| 機能 | SE (Standard Edition) | CGE (Carrier Grade Edition) |
|---|--------------------------|--------------------------------|
| NDB Bindings (C++ - NDB API, Java - NDB/J) | なし | あり |
| Geographical Replication | なし | オプション (有料) |
| LDAP Interface | なし | オプション (有料) |

- ▶ NDB Bindings
 - ▶ JAVAやC++などのアプリケーションからNDB APIを利用して直接Data Nodeへアクセスする方法
- ▶ Geographical Replication
 - ▶ MySQL Cluster ReplicationやGlobal Replicationと呼ばれていた
 - ▶ MySQL Clusterをサイト間で非同期レプリケーションする仕組み
 - ▶ ディザスターリカバリーに対応
- ▶ SEとCGEの違い
 - ▶ バイナリ上の違いは無い
 - ▶ ライセンス上の違いだけ

MySQL ClusterへのSCSの取り組み

MySQL ClusterへのSCSの取り組み

- ▶ 2004年
 - ▶ MySQL Clusterの検証を開始
- ▶ IPAプロジェクト (2004-2005)
 - ▶ オープンソースDBMSの評価プロジェクトにMySQL Cluster担当として参加
- ▶ Linux World 2005
 - ▶ 日立ブレードシンフォニーでMySQL Clusterをデモンストレーション
- ▶ Linux World 2006
 - ▶ 「MySQL Clusterの最適構成」セッション担当
- ▶ 2007年～
 - ▶ 実案件における問い合わせ/サービス提供

IPAプロジェクトの概要

▶ 正式名称

- ▶ 2005年度上期オープンソースソフトウェア活用基盤整備事業「OSS性能・信頼性評価 / 障害解析ツール開発」DB層

▶ SCSはMySQL Cluster担当として参加

- ▶ DBT(負荷ツール)を利用したDBMS性能検証にもMySQL担当として参画

▶ 評価レポートなどはIPAのサイトで公開中

- ▶ <http://www.ipa.go.jp/software/open/forum/development/index.html>

IPAプロジェクトの結果

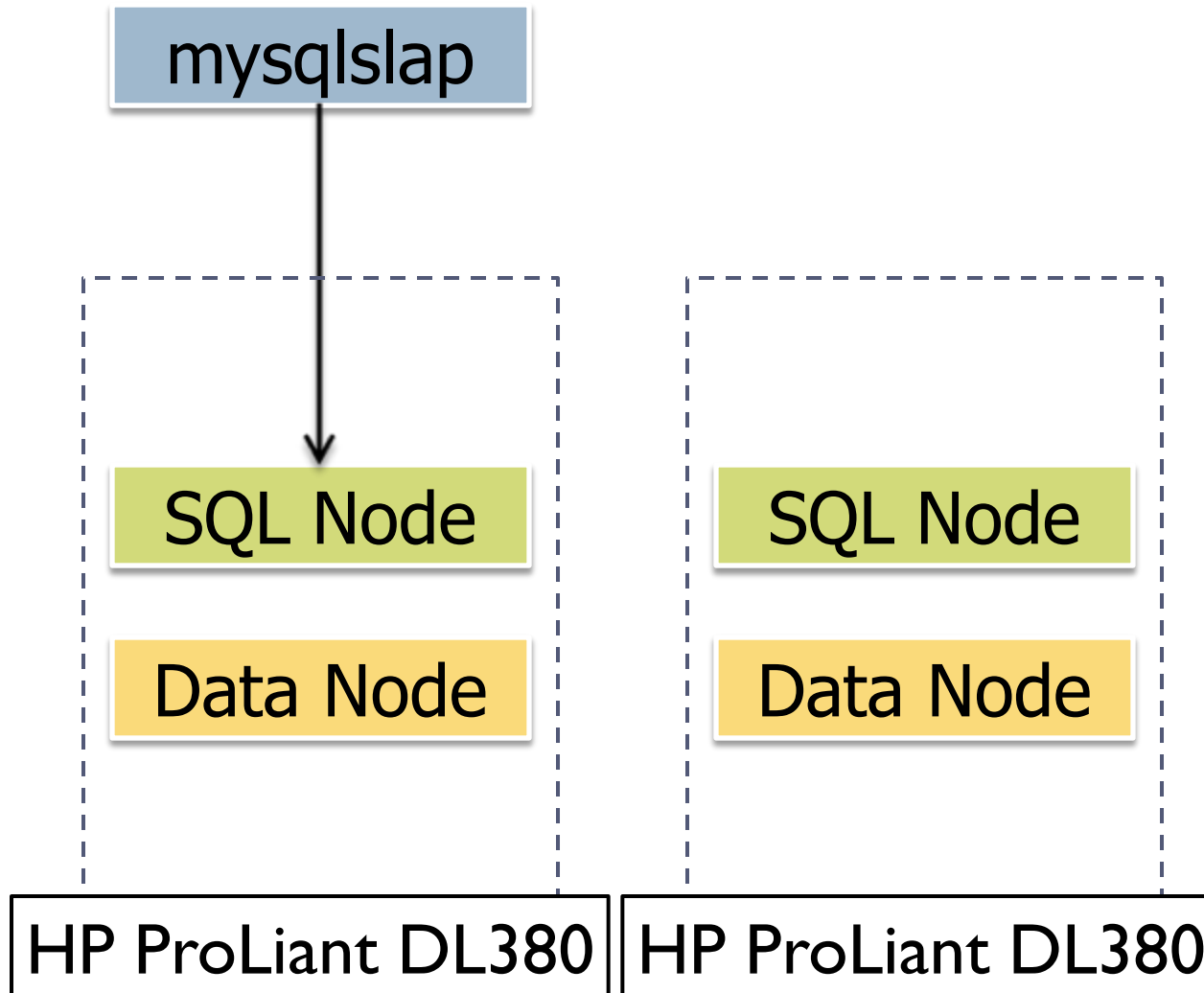
- ▶ MySQL ClusterのHA機能で不可解な挙動は無い
 - ▶ ノード障害、ネットワーク障害、サーバー障害などにも対応
 - ▶ フェールオーバーは非常に高速：数秒
- ▶ インターコネク트에SCI (Dolphin Interconnect Solutions社)を利用することで性能は最大50%向上した
- ▶ MySQL Cluster 4.1 / 5.0には高負荷時の安定性で若干の課題あり
 - ▶ 安定性は6.2で解決
 - ▶ 性能はCGEおよび6.2で改善

ベンチマーク

ベンチマーク実施時の各種スペック

- ▶ HW: HP ProLiant DL380 * 2台
 - ▶ CPU: Quad Core Xeon 2.6GHz * 2
 - ▶ Memory: 16GB
 - ▶ HDD: SAS 72GB, 15,000rpm
- ▶ SW: Red Hat Enterprise Linux 5.1 (EM64T)
- ▶ MySQL Cluster: 5.1.23
- ▶ MySQL Cluster CGE: 5.1.23 (mysql-5.1.23-ndb-6.2.14-telco)
- ▶ ベンチマークソフト: mysqlslap

サーバーの基本構成



mysqlslapの設定

- ▶ テーブルスキーマ
 - ▶ 右表のとおり
- ▶ 初期データサイズ
 - ▶ 100万件
 - ▶ 800MBほど
- ▶ クエリ種別
 - ▶ MIXED: PK SELECTとINSERTの混在

| カラム名称 | データ型 | 属性 |
|----------|--------------|-------------|
| id | VARCHAR(32) | PRIMARY KEY |
| intcol1 | INT | |
| intcol2 | INT | |
| intcol3 | INT | |
| intcol4 | INT | |
| intcol5 | INT | |
| charcol1 | VARCHAR(128) | |
| charcol2 | VARCHAR(128) | |
| charcol3 | VARCHAR(128) | |
| charcol4 | VARCHAR(128) | |
| charcol5 | VARCHAR(128) | |

a. MySQL Cluster Carrier Grade Edition

b. ディスクテーブル

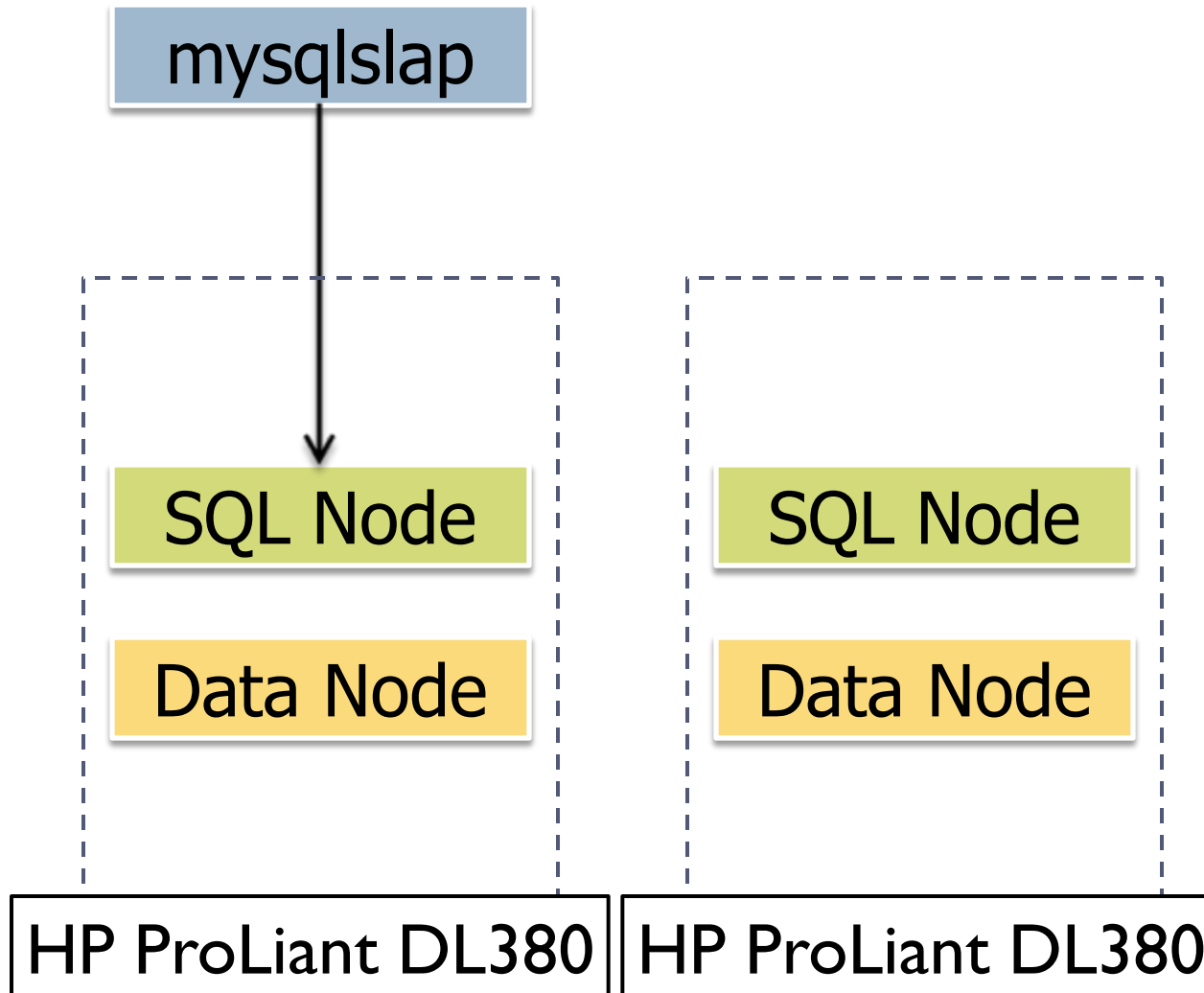
c. ノード数

d. InnoDB

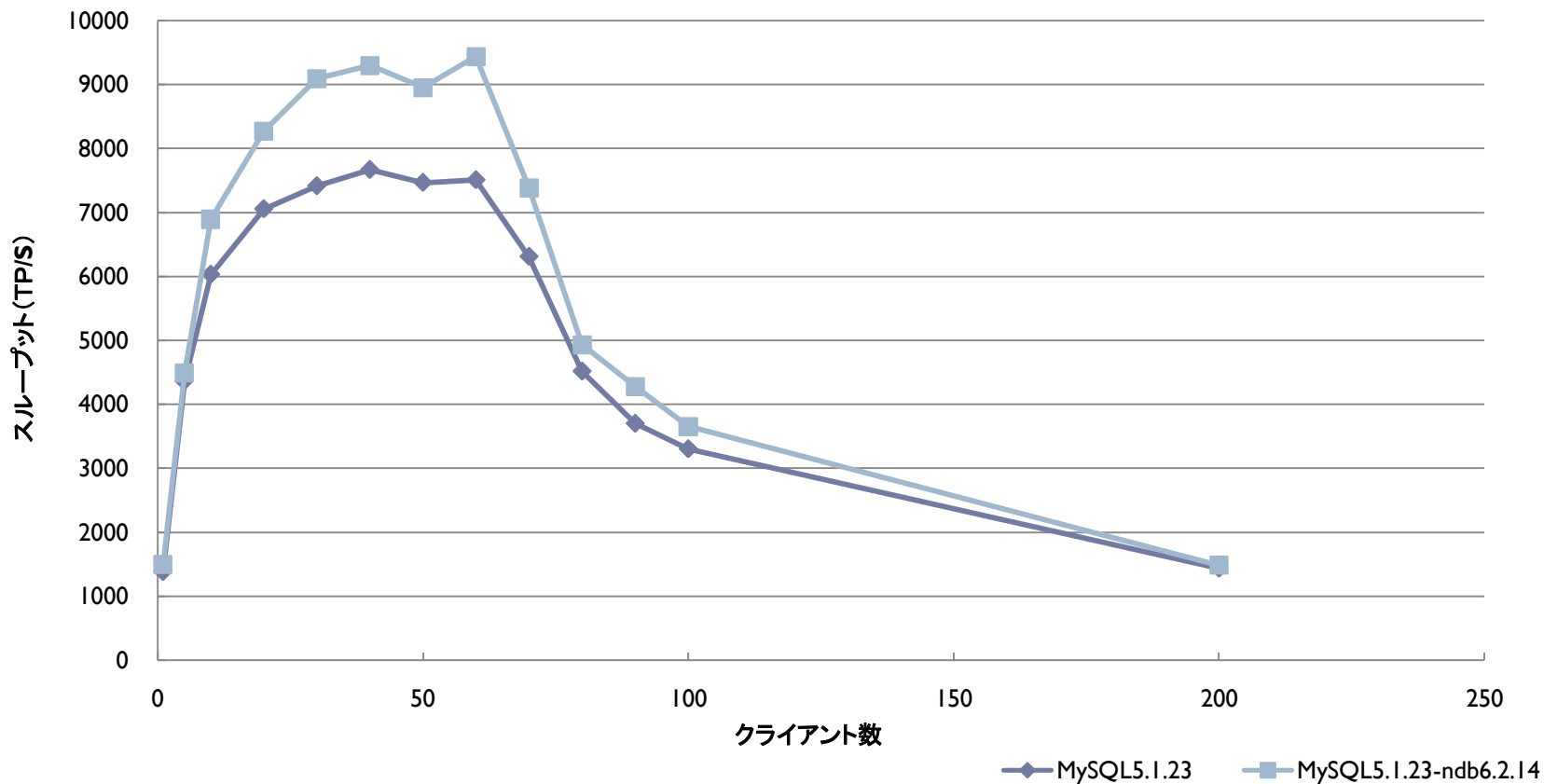
a. MySQL Cluster CGE

- ▶ **ベンチマークの目的**
 - ▶ MySQL ClusterとMySQL CGE (2008/5以前)の性能を比較したい
- ▶ **比較バージョン**
 - ▶ MySQL Cluster 5.1.23
 - ▶ MySQL Cluster 5.1.23 CGE (Data Node: 6.1.23)
- ▶ **構成**
 - ▶ Data Node: 2台
 - ▶ レプリカ(冗長性): 2
 - ▶ SQL Node: 2
 - ▶ データ: 初期データ100万(約800MB)
 - ▶ クエリ種別: MIXED (PK SELECTとINSERTのMIX)

a. MySQL Cluster CGE : 構成図



a. MySQL Cluster CGE : 結果



a. MySQL Cluster CGE：結果のまとめ

▶ 結果

- ▶ MySQL Cluster CGE (2008/5以前)では最大25%の性能改善が確認できた
 - ▶ 現在のMySQL Cluster 6.2 / 6.3は、このCGEをベースとしている
- ▶ 同時接続数=50で最も良い性能
 - ▶ 2006年に実施したベンチマーク結果と同じ (MySQL Cluster 5.0)

a. MySQL Cluster Carrier Grade Edition

b. ディスクテーブル

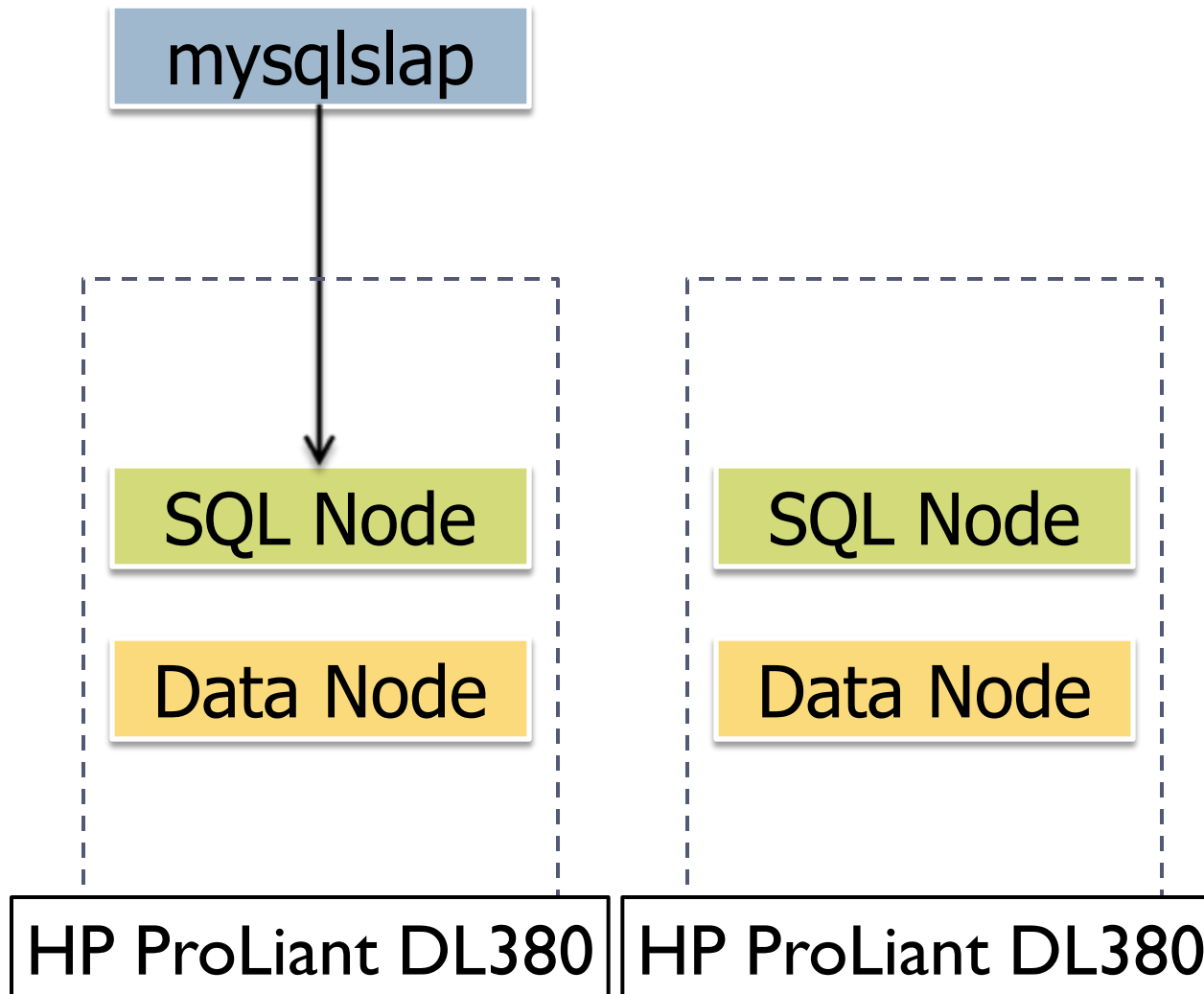
c. ノード数

d. InnoDB

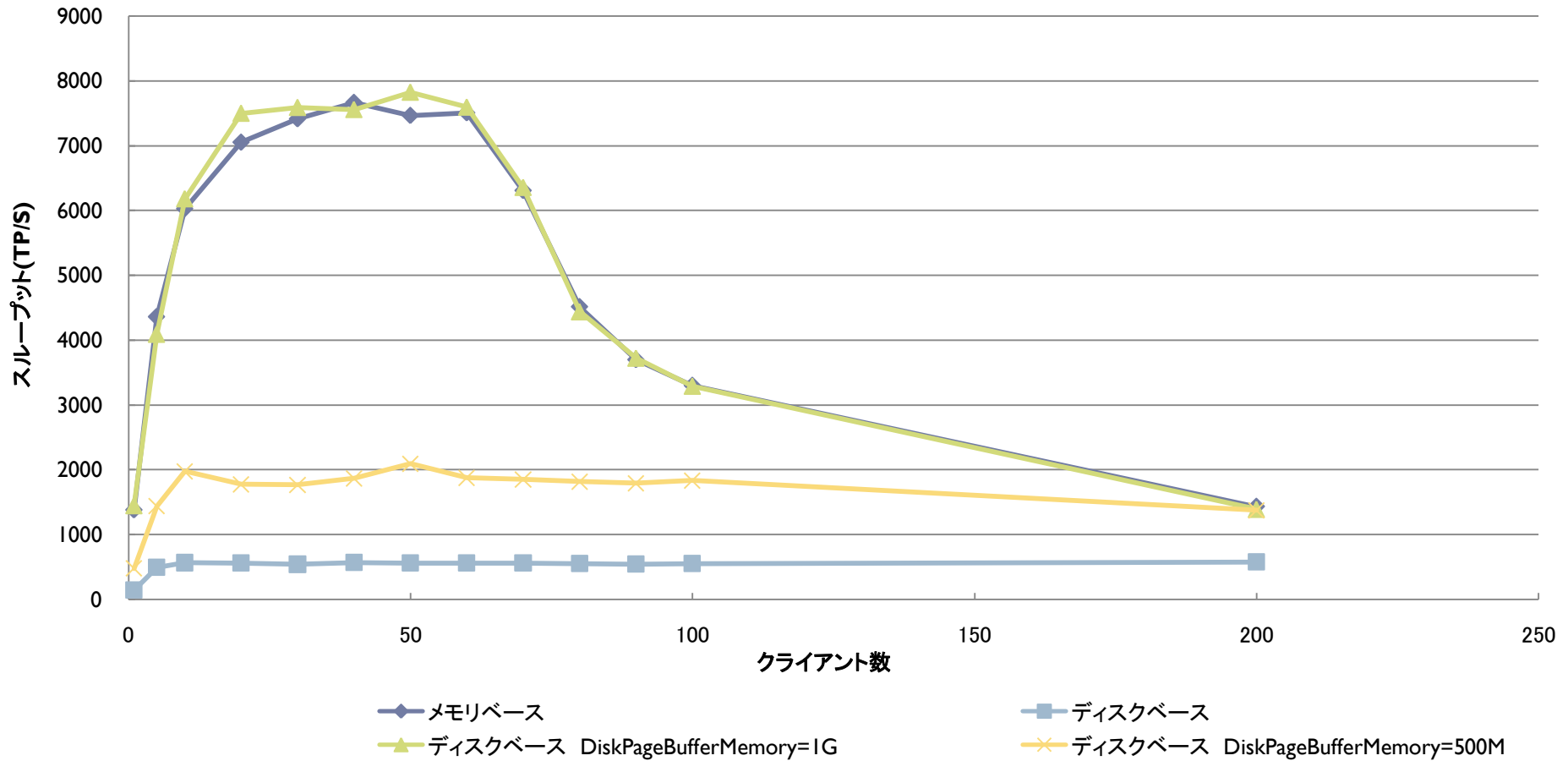
b. ディスクテーブル

- ▶ **ベンチマークの目的**
 - ▶ 5.1からサポートされたディスクテーブル(ディスクベーステーブル)が実際に「使える」レベルのものかを判断したい
- ▶ **ベンチマークの内容**
 - ▶ ディスクテーブル用のバッファの値を変えて計測した
- ▶ **構成**
 - ▶ Data Node: 2台
 - ▶ レプリカ(冗長性): 2
 - ▶ SQL Node: 2
 - ▶ データ: 初期データ100万(約800MB)
 - ▶ クエリ種別: MIXED (PK SELECTとINSERTのMIX)

b. ディスクテーブル：構成図



b. ディスクテーブル：結果



b. ディスクテーブル：結果のまとめ

▶ 結果

- ▶ ディスクテーブルの性能は低い
- ▶ DiskPageBufferMemoryをチューニングすることで性能は改善するが、効果は限定的
 - ▶ バッファに全データが乗る場合はメモリテーブルと同等の性能
 - ▶ バッファに半分以上のデータが乗る場合でもメモリテーブルの1/4ほどの性能
 - ▶ 可能な限り多く割り当てることを推奨
 - ▶ デフォルトは64MB
- ▶ mysqlslapはディスクテーブルに対応していないので手を加えた
 - ▶ SCSのサイトでパッチを公開中

a. MySQL Cluster Carrier Grade Edition

b. ディスクテーブル

c. ノード数

d. InnoDB

c-1. SQL Node

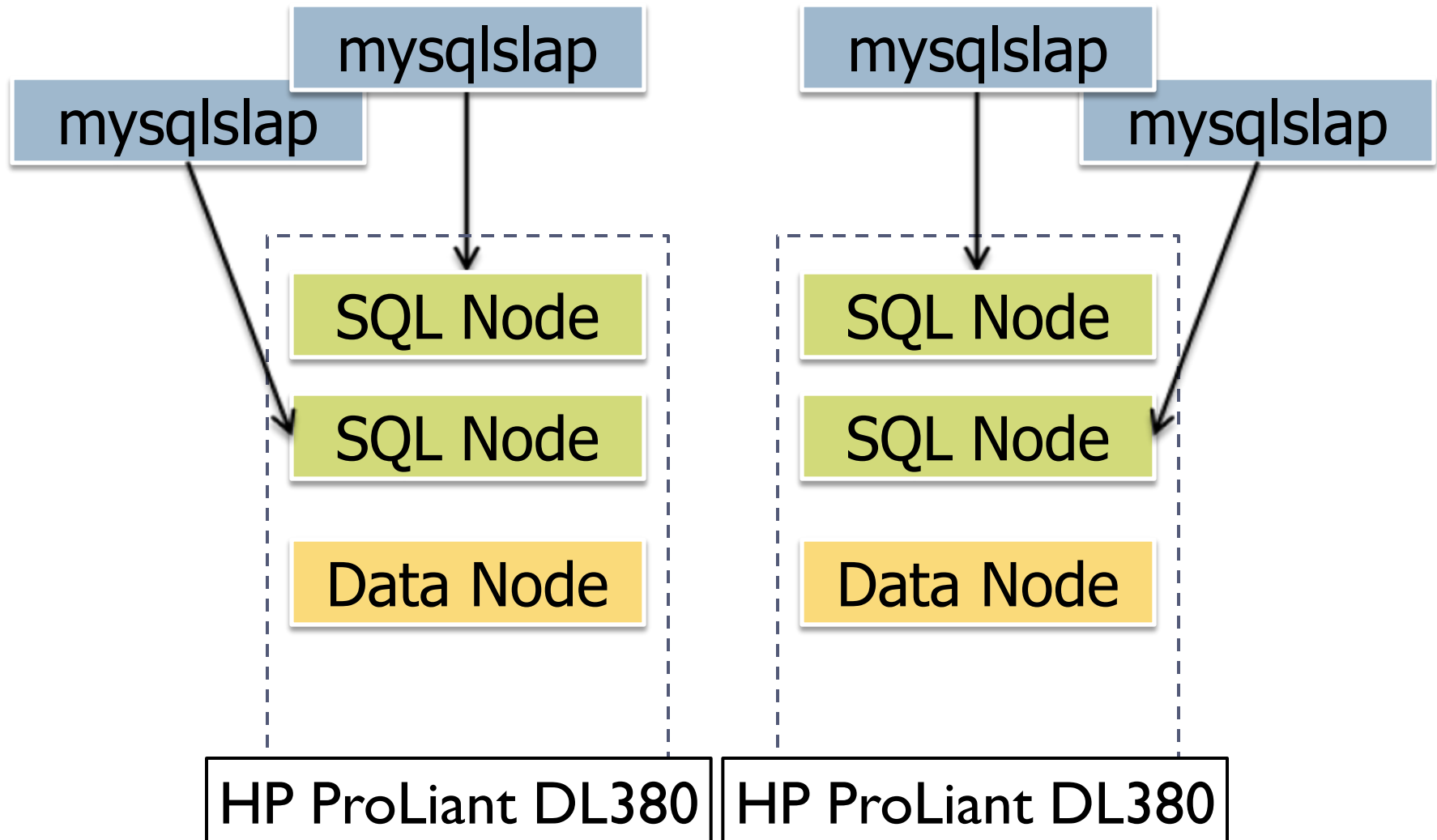
▶ ベンチマークの目的

- ▶ SQL Nodeのノード数と性能の関係を明らかにして、構成を考える際の指針とする

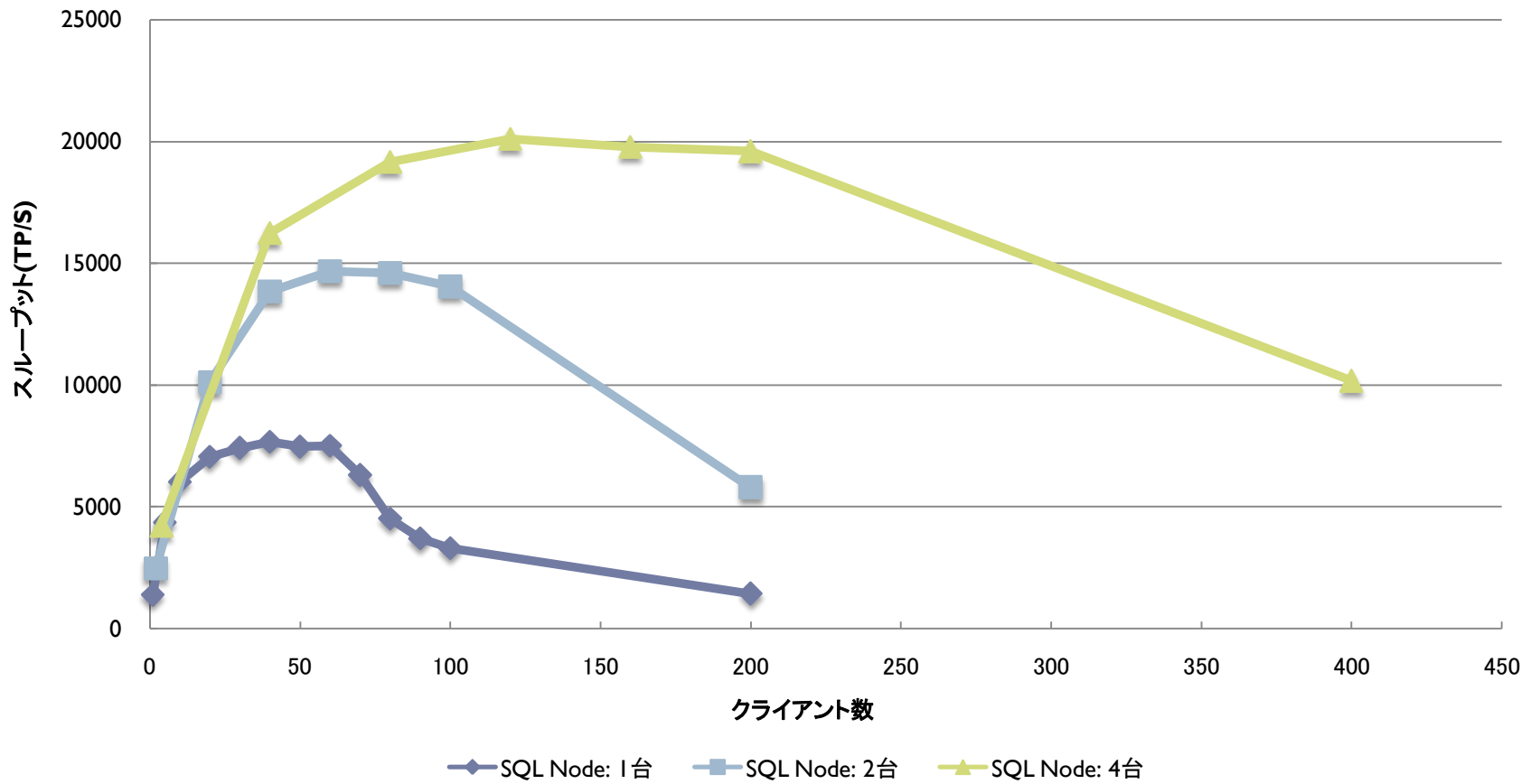
▶ 構成

- ▶ Data Node: 2台
- ▶ レプリカ(冗長性): 2
- ▶ SQL Node: 1, 2, 4
- ▶ データ: 初期データ100万(約800MB)
- ▶ クエリ種別: MIXED (PK SELECTとINSERTのMIX)

c-1. SQL Node : 構成図



c-1. SQL Node : 結果



c-1. SQL Node：結果のまとめ

▶ 結果

- ▶ SQL Node 1台あたり「50同時接続で最大性能」という傾向は変わらない
 - ▶ SQL Node 4台までは順当に性能はスケールした
- ▶ Data NodeのCPU利用率はSQL Node 4台で、初めて90%近くまで上昇
 - ▶ 6.2, 6.3ではData Nodeはシングルスレッドなので複数コアなどは有効活用できない

c-2. Data Node

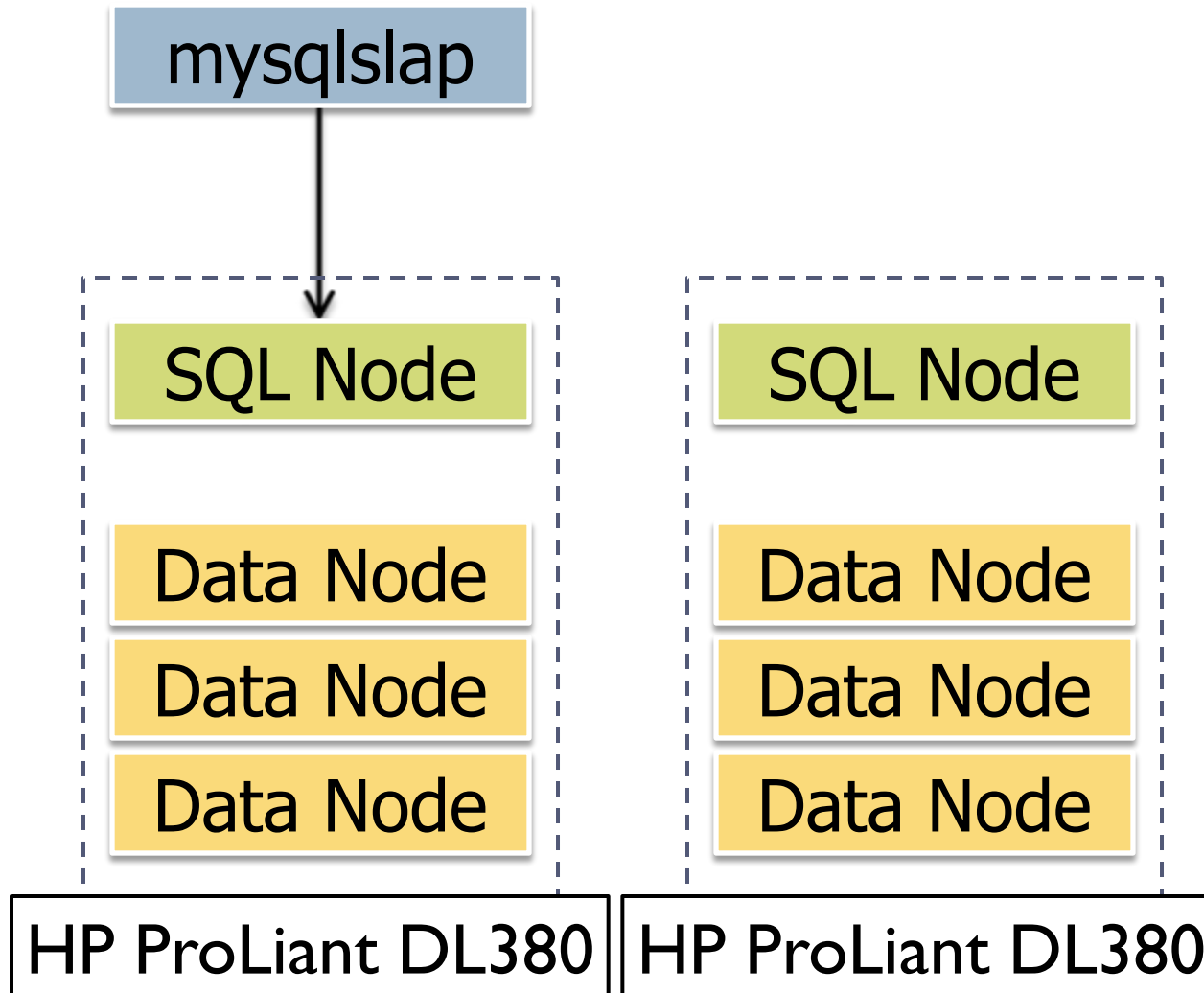
▶ ベンチマークの目的

- ▶ Data Nodeのノード数と性能の関係を明らかにして、構成を考える際の指針とする
- ▶ レプリカの数(冗長構成)は2で固定する

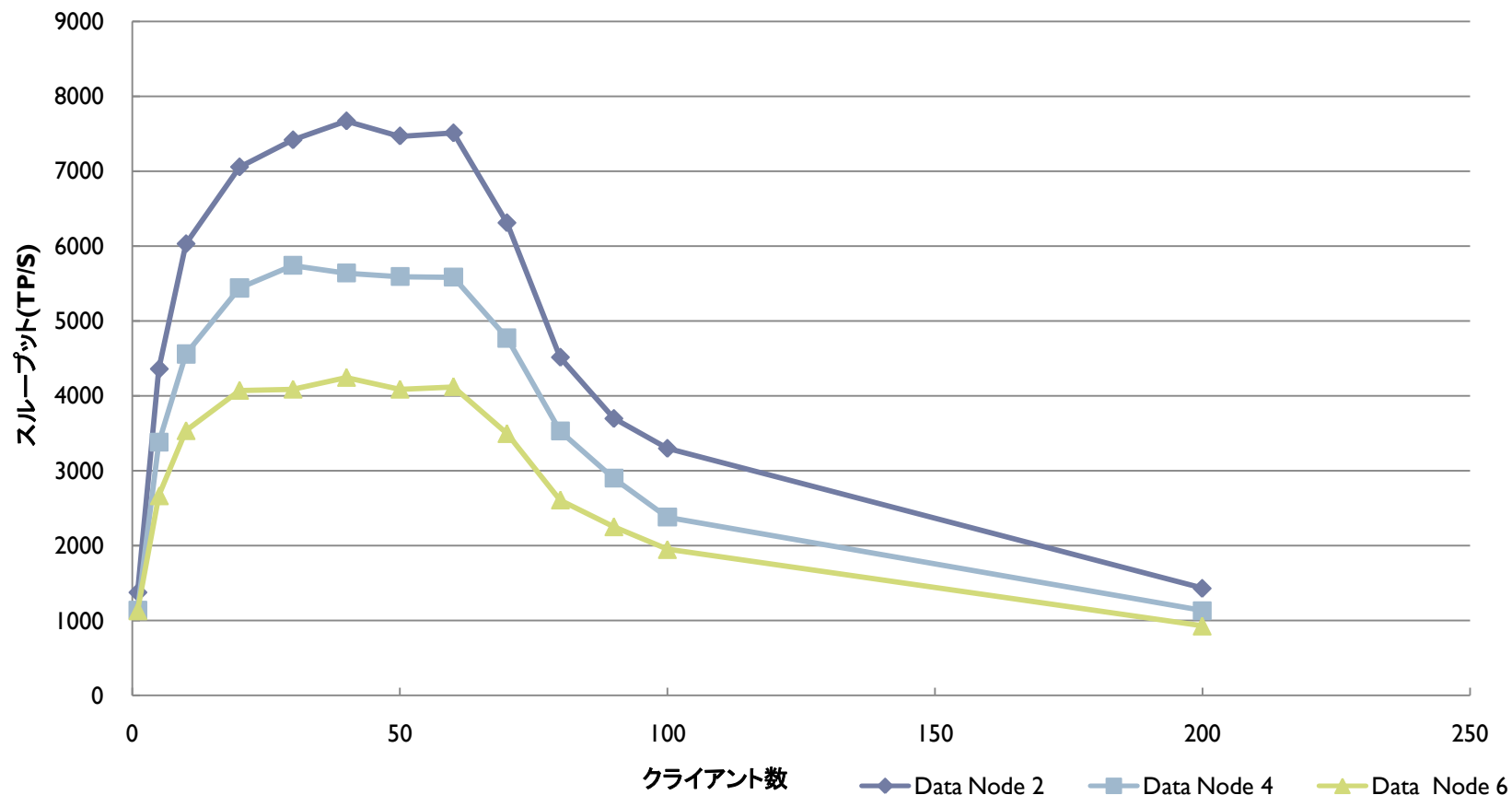
▶ 構成

- ▶ Data Node: 2, 4, 6
- ▶ レプリカ(冗長性): 2
- ▶ SQL Node: 2
- ▶ データ: 初期データ100万(約800MB)
- ▶ クエリ種別: MIXED (PK SELECTとINSERTのMIX)

c-2. Data Node : 構成図



c-2. Data Node : 結果



c-2. Data Node：結果のまとめ

▶ 結果

- ▶ Data Nodeを増やすことで性能は劣化した
- ▶ 性能劣化の原因は、ノードが増えたことによるオーバーヘッドの増加にあると思われる

a. MySQL Cluster Carrier Grade Edition

b. ディスクテーブル

c. ノード数

d. InnoDB

d. InnoDB

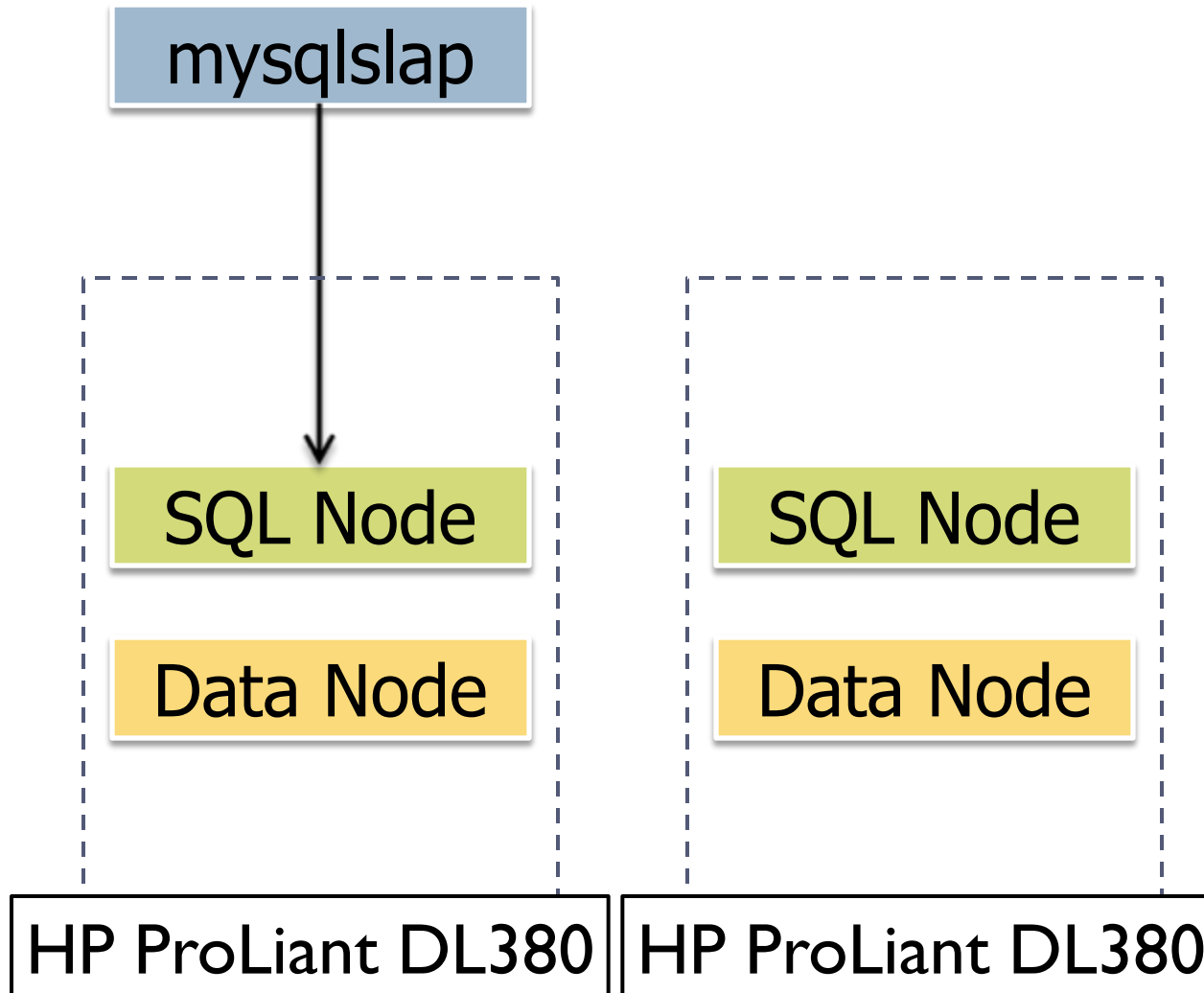
▶ ベンチマークの目的

- ▶ 「更新処理を負荷分散できる」ことを特徴の一つとしているMySQL Clusterが、最もポピュラーなInnoDBと性能面でどこまで迫れるか判断したい

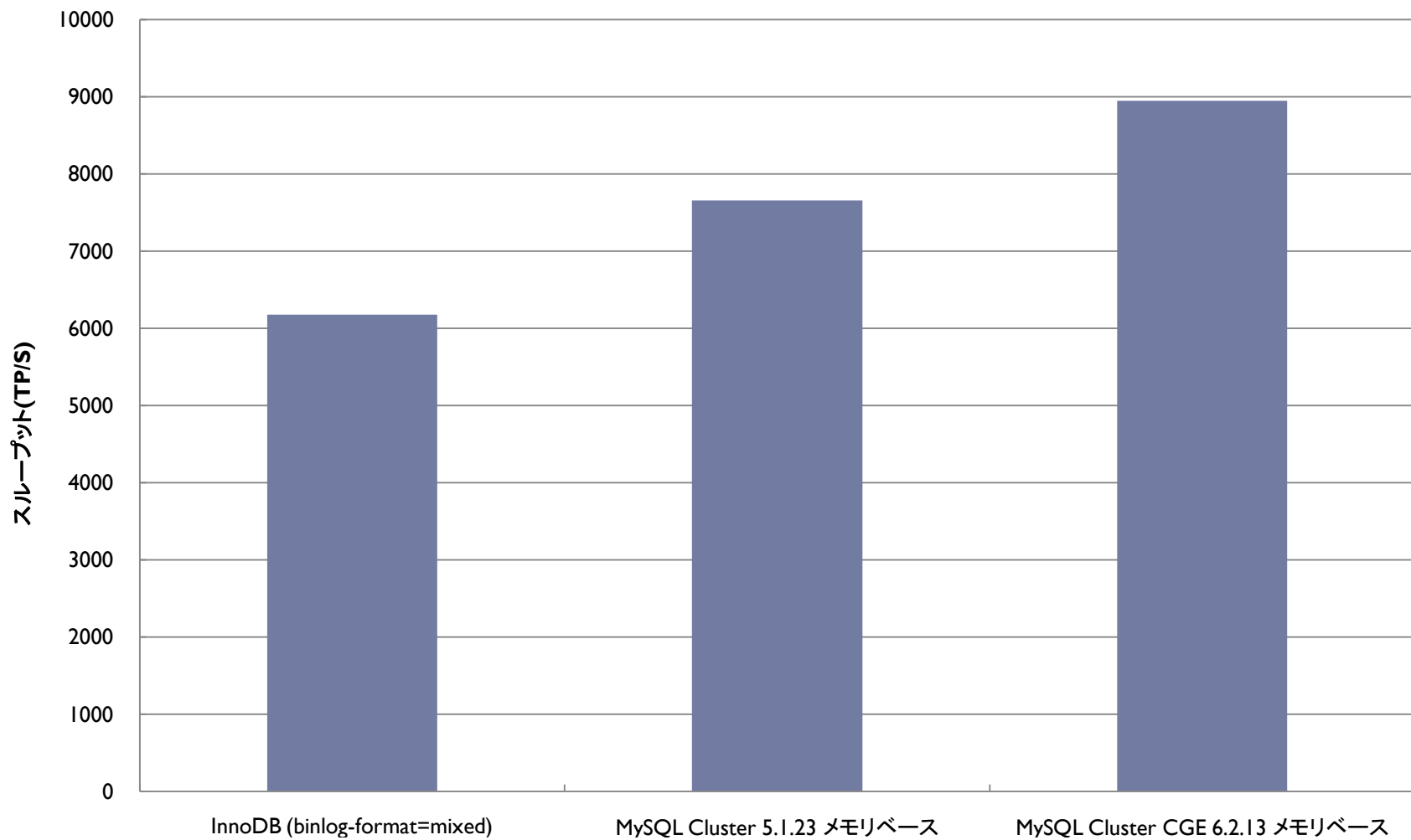
▶ 構成

- ▶ Data Node: 2
- ▶ レプリカ(冗長性): 2
- ▶ SQL Node: 2
- ▶ データ: 初期データ100万(約800MB)
- ▶ クエリ種別: MIXED (PK SELECTとINSERTのMIX)
- ▶ クライアント数: 50

d. InnoDB : 構成図



d. InnoDB：結果



d. InnoDB：結果のまとめ

▶ 結果

- ▶ InnoDB (バイナリログ有効)よりもMySQL Clusterの方が性能は良い
- ▶ ただし、バイナリログを無効にしたInnoDBは倍近い性能となったので、「書き込みキャッシュ」をサポートしたディスクコントローラでは、バイナリログ有効のInnoDBの性能も改善するはず

a,b,c,d: ベンチマーク結果のまとめ

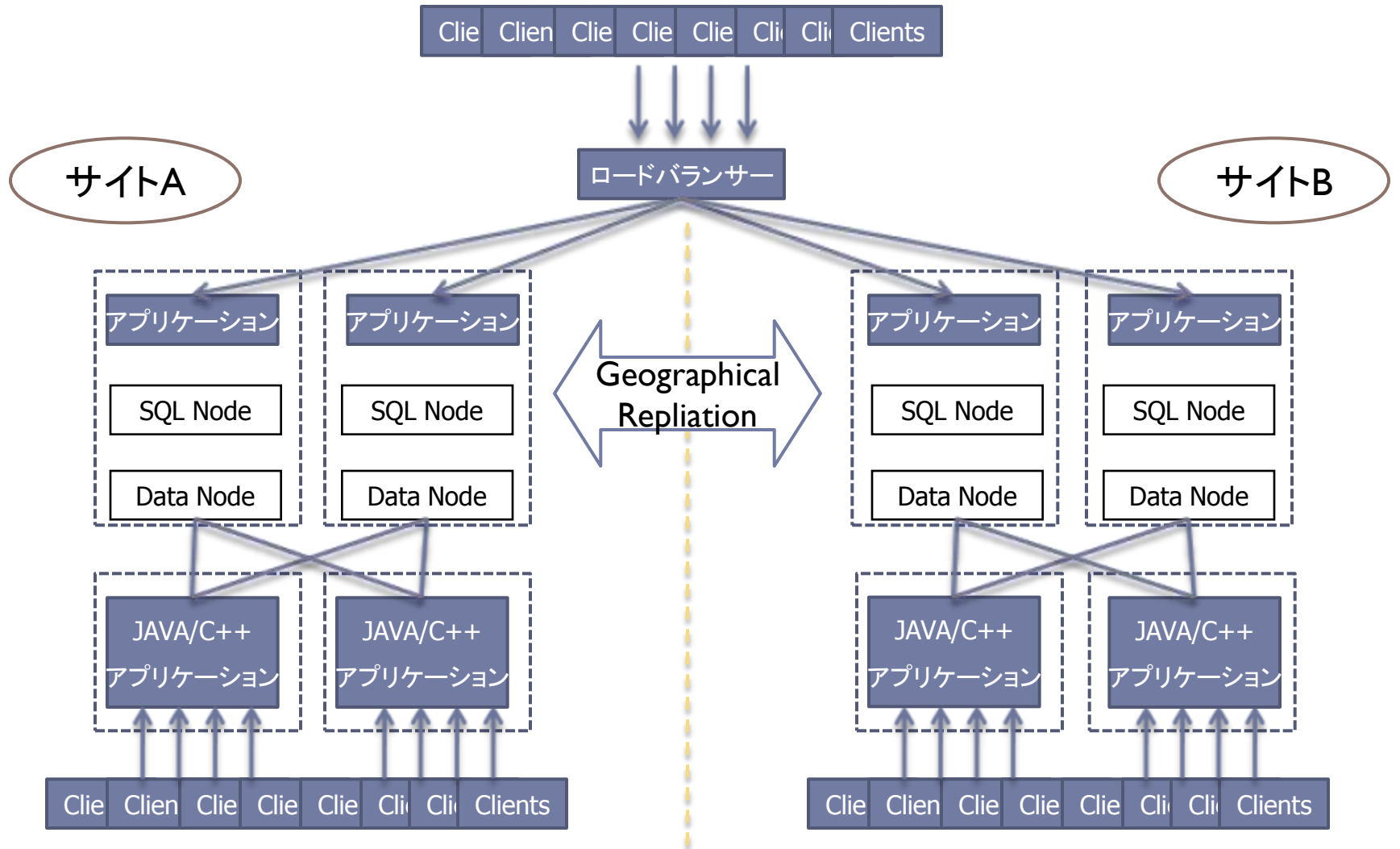
- ▶ MySQL Cluster 6.2は安定している
 - ▶ MySQL Cluster 4.1, 5.0, 5.1を利用している場合はアップグレードを強く推奨
- ▶ ディスクテーブルの全面的な採用は未だ早い
 - ▶ 性能面での懸念: 特定のテーブルのみとするべき
- ▶ Data Node数は少なく、SQL Node数は多く、レプリカ数は少なく、同時接続数は50くらい、が初期構成の目安
 - ▶ MySQL Cluster 5.0と傾向は変わらない
- ▶ Data Nodeはシングルスレッドでの動作となるので、CPUなどを増やしても性能はスケールしない
 - ▶ 6.4で改善される予定

MySQL Clusterのシステム構成例

MySQL Clusterで高性能システムを構築する際のポイント

- ▶ MySQL Cluster 6.2を利用する
- ▶ データサイズを見積もる
- ▶ ディスクテーブルの利用は控えめに
- ▶ SQL Nodeの冗長化
 - ▶ アプリケーションレベル/ロードバランサーなどで実現
- ▶ Data Node
 - ▶ 冗長性(レプリカ)は2で十分
- ▶ 性能重視の場合はNDB APIの利用も検討する

MySQL Clusterシステム構成例



MySQL Clusterシステム構成例

- ▶ サイトA、サイトBでマスター・マスターのGeographical Replication
 - ▶ ディザスターリカバリ対策
- ▶ SQL Node経由では性能が十分でないアプリケーションはNDB APIを利用して直接Data Nodeへアクセス
- ▶ SQL Nodeの冗長性はロードバランサーで実現

MySQL Clusterに適したアプリケーション

- ▶ データの見積もりが設計時にある程度可能なシステム
- ▶ JOINやSUB QUERYをほとんど利用しないアプリケーション
- ▶ 検索するカラムにはインデックスがはられている
- ▶ 単純なSQLクエリを大量に発行するアプリケーション
- ▶ 高い可用性が要求されるシステム

MySQL Clusterに適さないアプリケーション

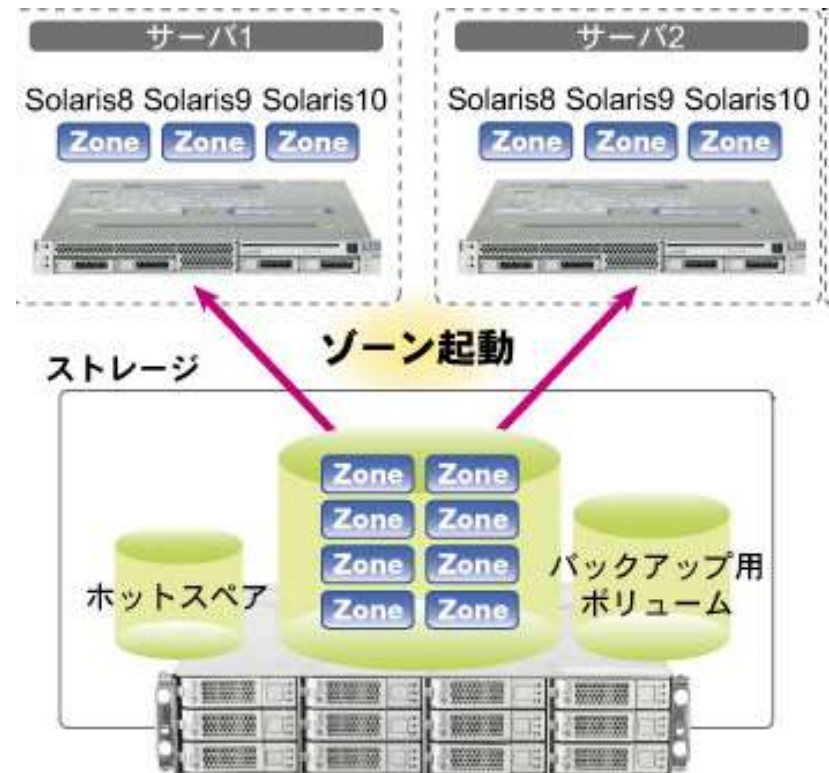
- ▶ ディスクテーブルの利用を前提とする大規模データベース
 - ▶ InnoDBをお勧めします
- ▶ データの大部分が文字列で、検索が主体のアプリケーション
 - ▶ InnoDBとMySQL+Sennaのレプリケーション構成をお勧めします
- ▶ ダウンタイムが許容されるシステム
 - ▶ MyISAMやInnoDBをレプリケーションで利用することをお勧めします
- ▶ 複雑なSQLクエリを発行するシステム
 - ▶ JOINやSUB QUERYを多用する場合には性能が劣化します

さいごに

Dolphin Zone by SCS

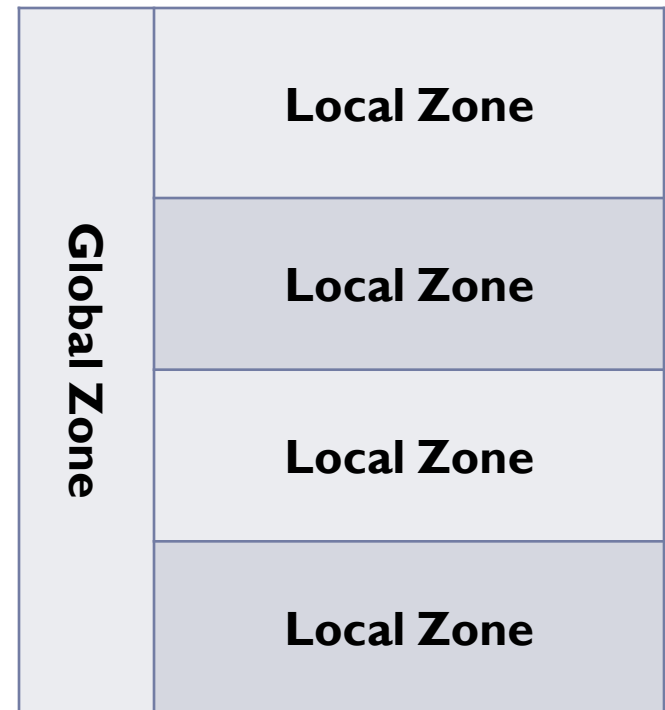
▶ Dolphin Zoneとは

- ▶ SUNの仮想化プラットフォーム「Viking Zone」にSCSのMySQL設計、構築、サポートサービスを付加したパッケージ製品
- ▶ HW構成
 - ▶ Sun SPARC Enterprise T5220 * 2台
 - ▶ Sun StorageTek 2540 FC * 1台



Zoneとは？

- ▶ SUNが提供する仮想化の仕組み
- ▶ 他の仮想化と比べてオーバーヘッドが少ない
- ▶ Global Zone
 - ▶ 従来のSolaris環境
 - ▶ Xenのdom0に相当
- ▶ Local Zone
 - ▶ Global Zone上に構成される仮想環境
 - ▶ XenのdomUに相当



MySQL Clusterの制限のひとつ

- ▶ サービスを提供している最中に、当初の見積よりもデータ容量が大きくなってしまったら
 - ▶ MySQL Clusterだけでは対応が難しい
 - ▶ メモリテーブル、ディスクテーブルともにテーブルサイズの見積が重要
- ▶ Data Nodeの構成変更にはMySQL Cluster全体の再起動が必要
 - ▶ 冗長性(レプリカ数)の変更
 - ▶ Data NodeのIPアドレスの変更
 - ▶ Data Node数の変更
 - ▶ Data Memory, Index Memoryなどの増加は可能
- ▶ 6.4でData Nodeの動的追加などが実装予定
- ▶ 6.2, 6.3ではどうするか？

Dolphin Zoneが有効なシナリオ

- ▶ 当初のデータ見積は64GBで、128GBまで拡張する可能性がある
- ▶ 128GBまでデータが増大するかどうか、サービスを開始しないと分からない
- ▶ 性能の観点からディスクテーブルは使いたくない
- ▶ 初期投資は極力抑えたい

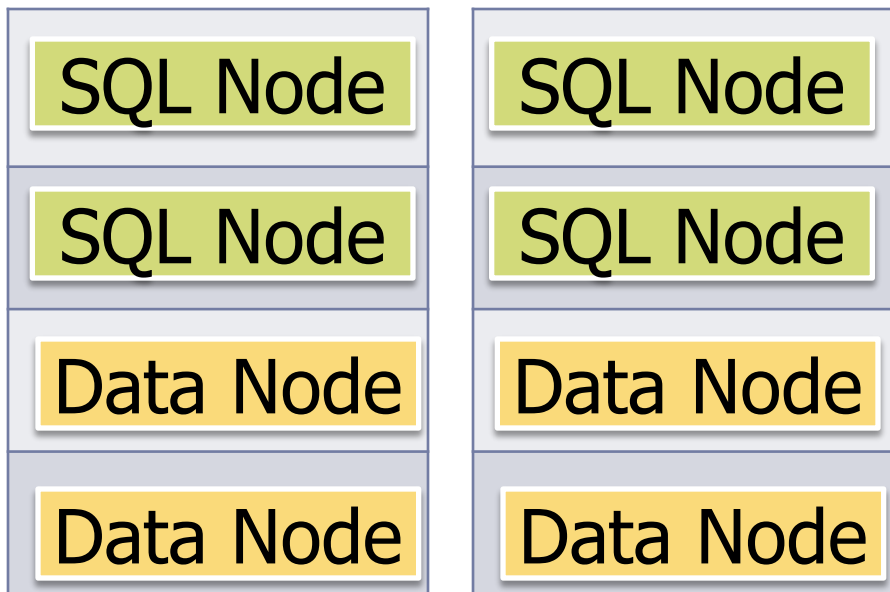
→Dolphin Zoneが解決！！！！

Dolphin ZoneでのMySQL Cluster構成

- ▶ Local ZoneにData Nodeをインストールする
- ▶ MySQL Clusterの設定
 - ▶ SQL Node : 多めに設定→性能には影響しない
 - ▶ Data Node : 多めに設定→性能に影響するので注意
- ▶ 当面のサーバーは2台
- ▶ データ増加時はData Nodeを他のサーバーのLocal Zoneへ移行/マイグレーションする
 - ▶ Data Nodeが稼働しているLocal ZoneのIPアドレスはそのままなので、オンラインで移行できる

サービス開始時

- ▶ SQL Node, Data NodeはLocal Zone上に構成する

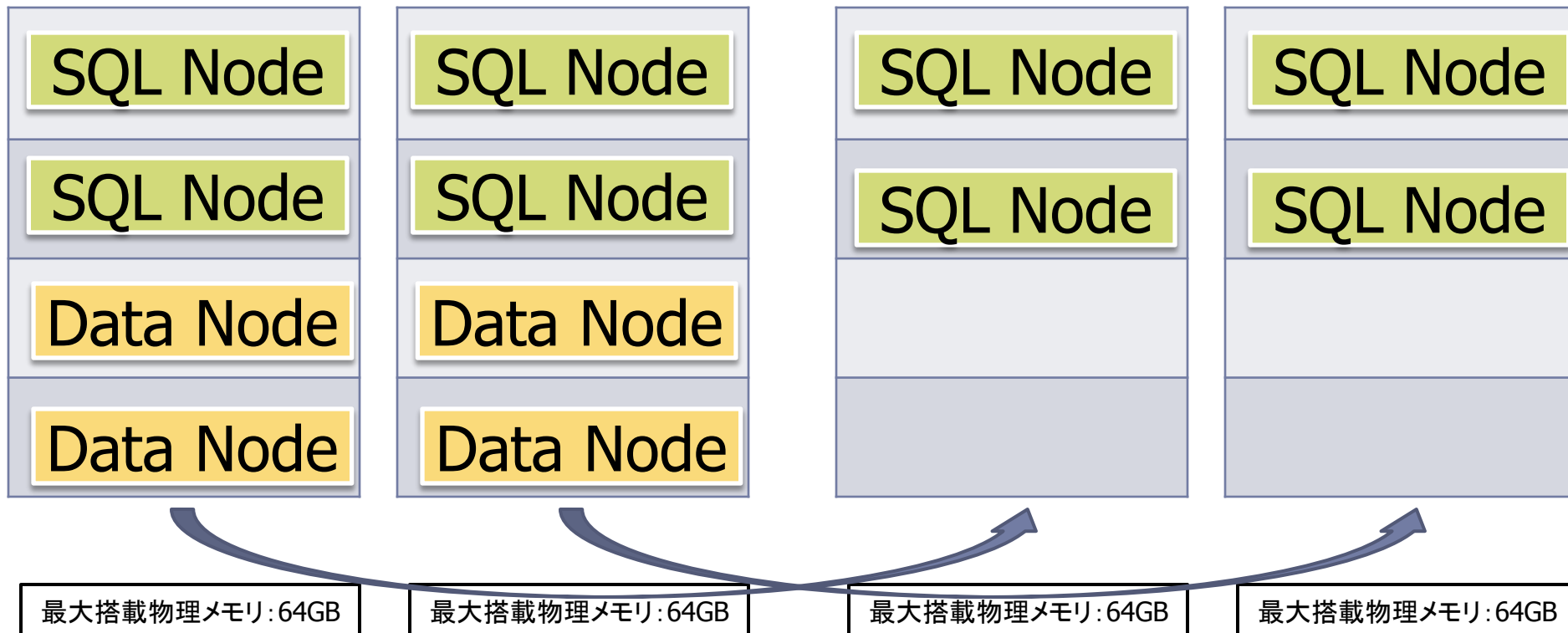


最大搭載物理メモリ: 64GB

最大搭載物理メモリ: 64GB

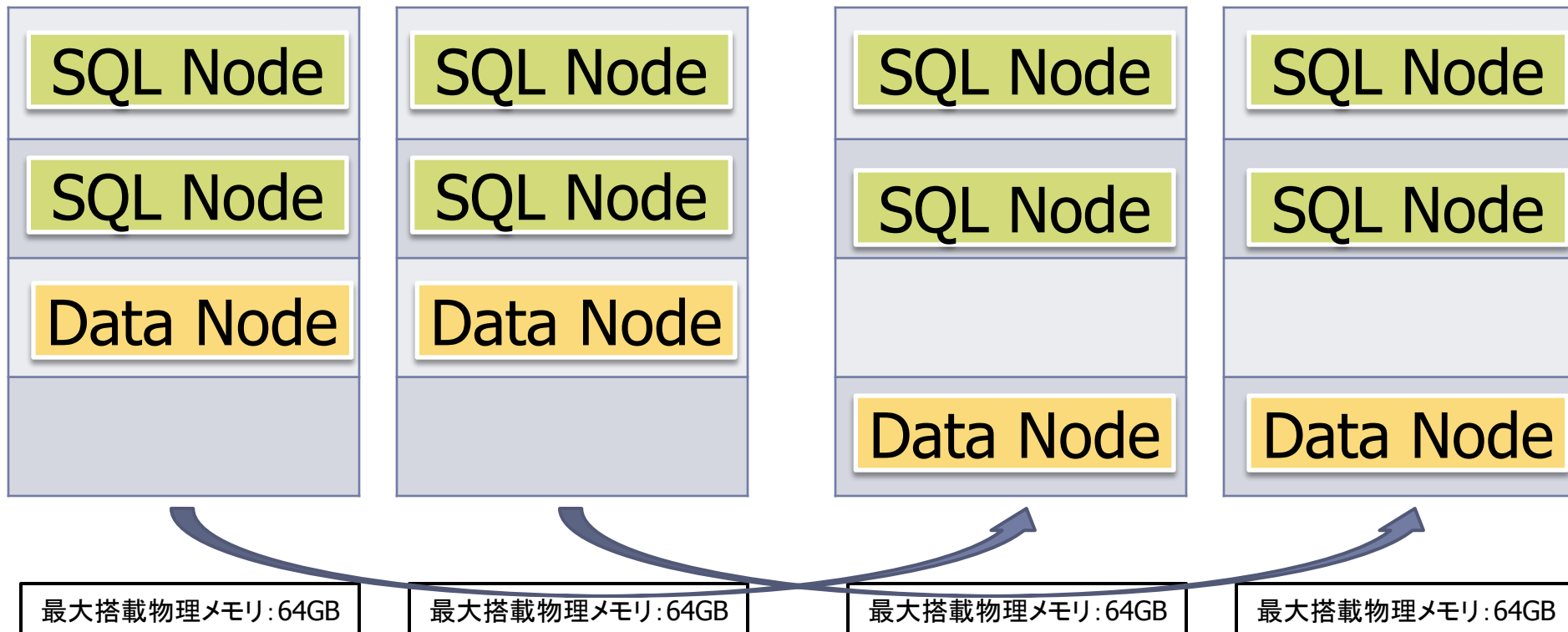
データ増加時：オンラインでマイグレーション

- ▶ Data Nodeを起動しているLocal Zoneを別のサーバーへ移行する
- ▶ Data NodeのIPアドレスは変わらないので、全てオンラインで実施できる



データ増加時の構成

- ▶ 約128GBの物理メモリをData Nodeで利用できる



住商情報システムでは

- ▶ MySQLトレーニング
 - ▶ MySQL 5.1 for DBA
 - ▶ MySQL High Availability
- ▶ MySQLサポート
 - ▶ 2005/4より開始
 - ▶ 60社以上への導入実績
- ▶ MySQLプロフェッショナルサービス(コンサルテーション)
- ▶ システム・インテグレーション
 - ▶ MySQLを利用したシステム構築
 - ▶ OracleからMySQLへのマイグレーション
 - ▶ MySQLとOracleを利用するシステム構築
 - ▶ →Oracleのプロフェッショナルも同じチームに多数在籍

参考資料

mysql scs

検索



- ▶ <http://www.scs.co.jp/mysql/>
 - ▶ 今回のプレゼン資料
 - ▶ mysqlslapパッチ
 - ▶ その他各種技術資料
- ▶ DB Magazine 2008年8月号
 - ▶ 「徹底検証:MySQL Cluster」
- ▶ Viking Zone
 - ▶ <http://www.clubscs.com/vikingzone/>

さいごに

- ▶ MySQL Cluster 6.2は「使え」ます
 - ▶ 海外での事例は多数
 - ▶ 国内でも増えてきている
- ▶ ご清聴ありがとうございました