



OSSのカラム型データベースエンジン

MariaDB ColumnStore

ビッグデータ分析などに適した大規模並列処理に対応する

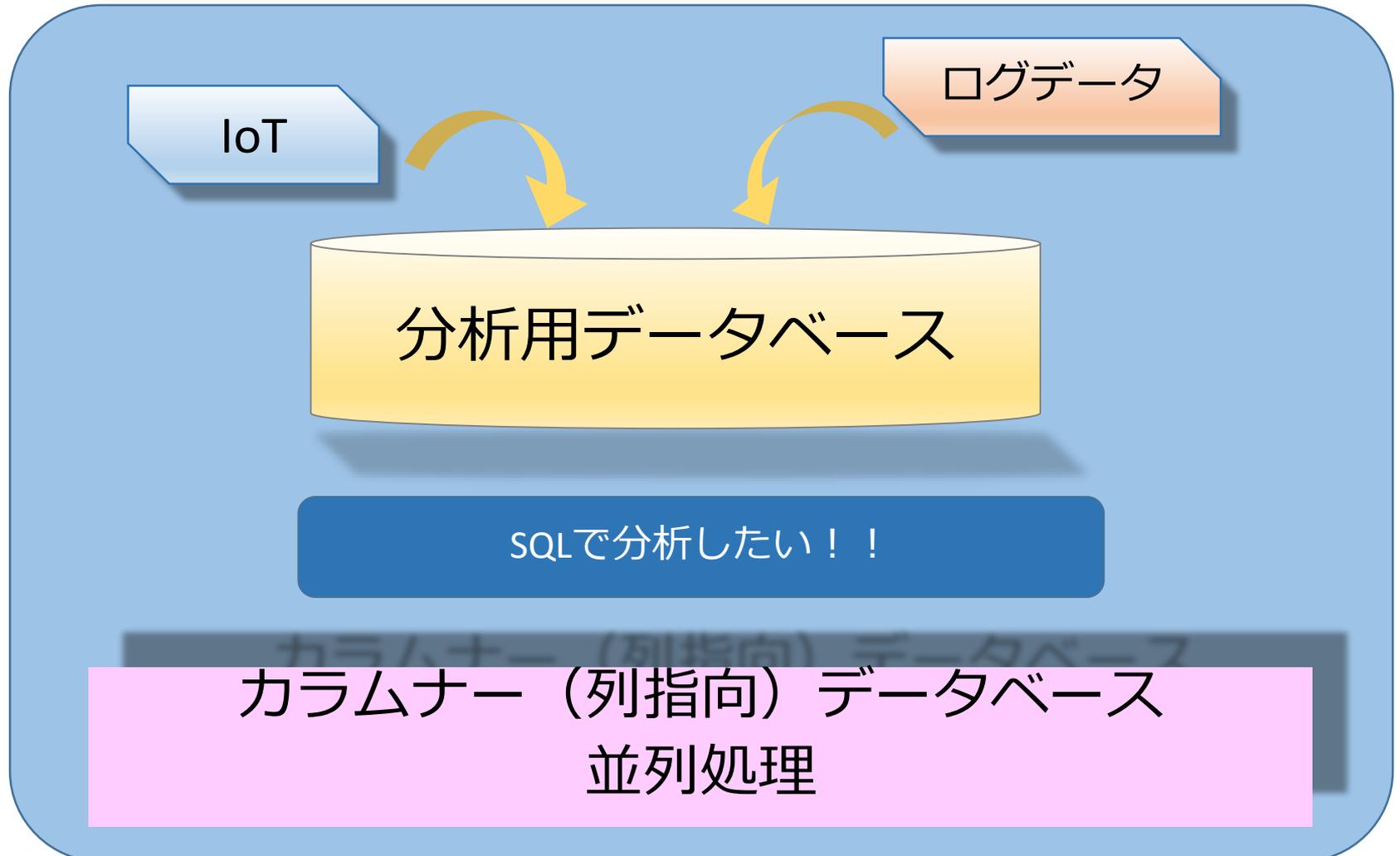
データベースエンジン

MariaDB について

- 「MySQL」から派生したオープンソースリレーショナルデータベース。
- MariaDBは、MySQLのオリジナルコード開発者である「Michael “Monty” Widenius」氏によって開発されている。
- 「MySQL」と「MariaDB」は、基本的にはほぼ同じ機能。
- 異なる点としては、ストレージエンジンの追加/変更がされている。
(オプティマイザーとか、他にもあるけど。。。別な機会に)
- MariaDBは、MySQLの標準ストレージエンジンに加え、多くのストレージエンジンを採用。
Aria、InnoDB/XtraDB、MyISAM、NDB、Archive、Memory、Federated-X、**ColumnStore**、Galera、Sphinx、TokuDB、Connect、Cassandra、Spider

本日は、MariaDB Column Store とSpider についての解説。

昨今のDWH



列指向データベースのメリット

- 列指向とは?
- メリットは?

(シーケンシャルな)アクセス効率 + データ圧縮 = 分析基盤に向いている

→

Prod ID	Prod Name	Date
Prod ID	Prod Name	Date
Prod ID	Prod Name	Date
Prod ID	Prod Name	Date
Prod ID	Prod Name	Date



1ブロックには様々なタイプのデータが格納されているので**圧縮効率を上げにくい**

↓

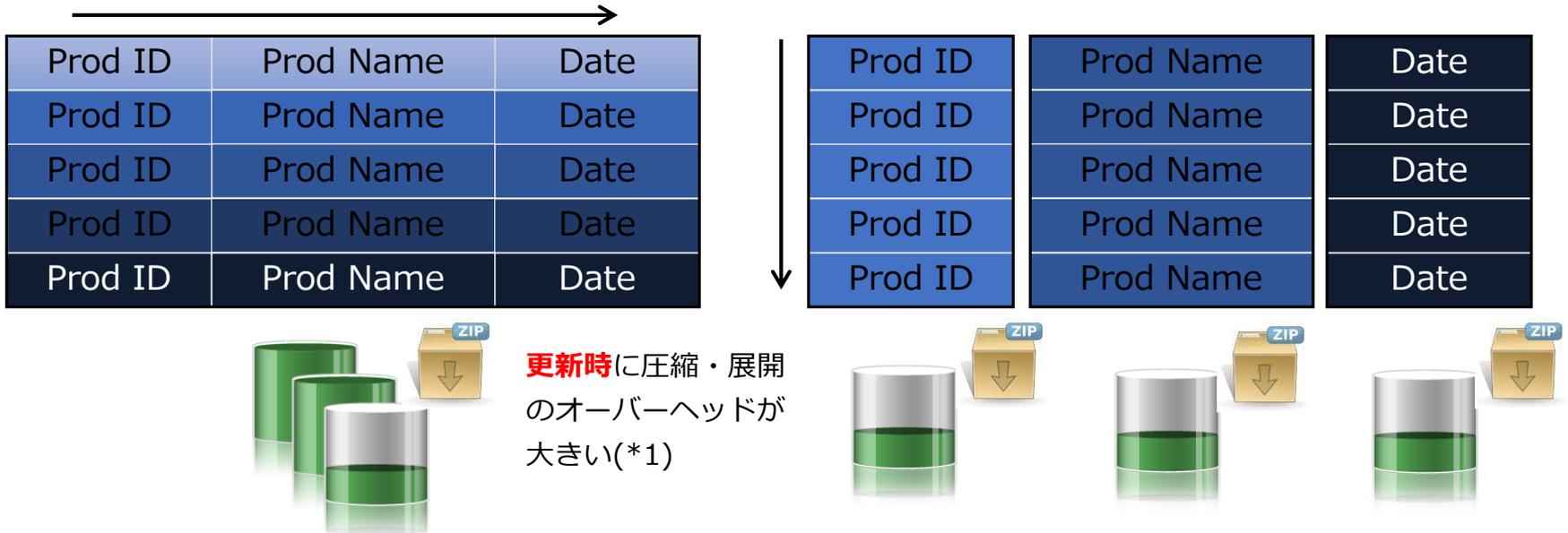
Prod ID	Prod Name	Date
Prod ID	Prod Name	Date
Prod ID	Prod Name	Date
Prod ID	Prod Name	Date
Prod ID	Prod Name	Date



1ブロックには同様のタイプのデータが格納されているので**圧縮効率を上げやすい**

列指向データベースのデメリット

- デメリットは?
一般的に
(ランダムな)アクセスが非効率 + データ圧縮 = OLTP基盤に向いていない



データベースを稼働させるサーバの 並列化システム構成といえは・・・

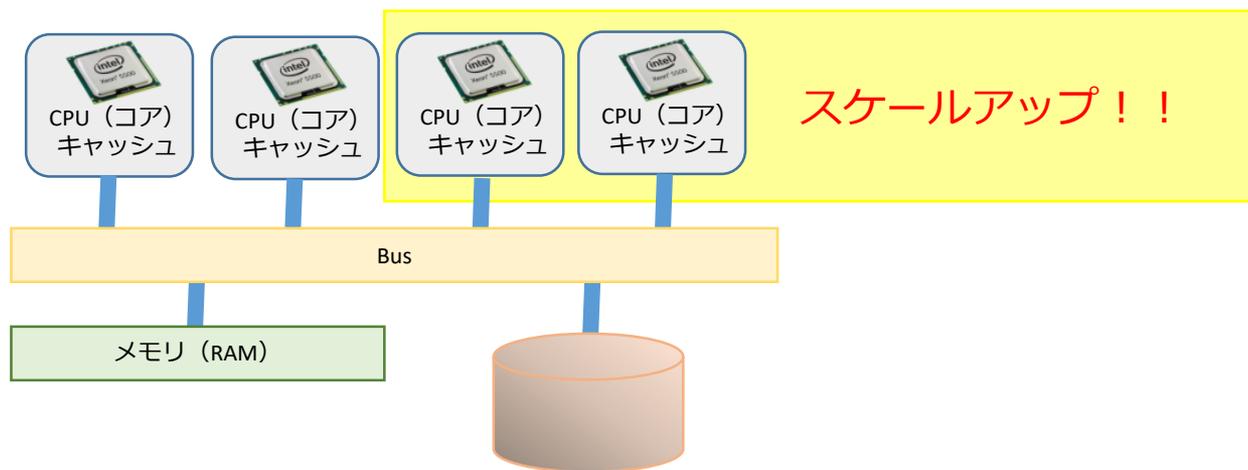
- SMPシステム
- MPPシステム

SMP

Symmetric Multi-processing

「対称型マルチプロセッシング」

- 物理メモリ共有型の並列コンピューティング方式
- 一台の筐体内に複数のCPU（コア）を搭載したサーバ
- CPU(コア)を増設することで処理能力を向上させる

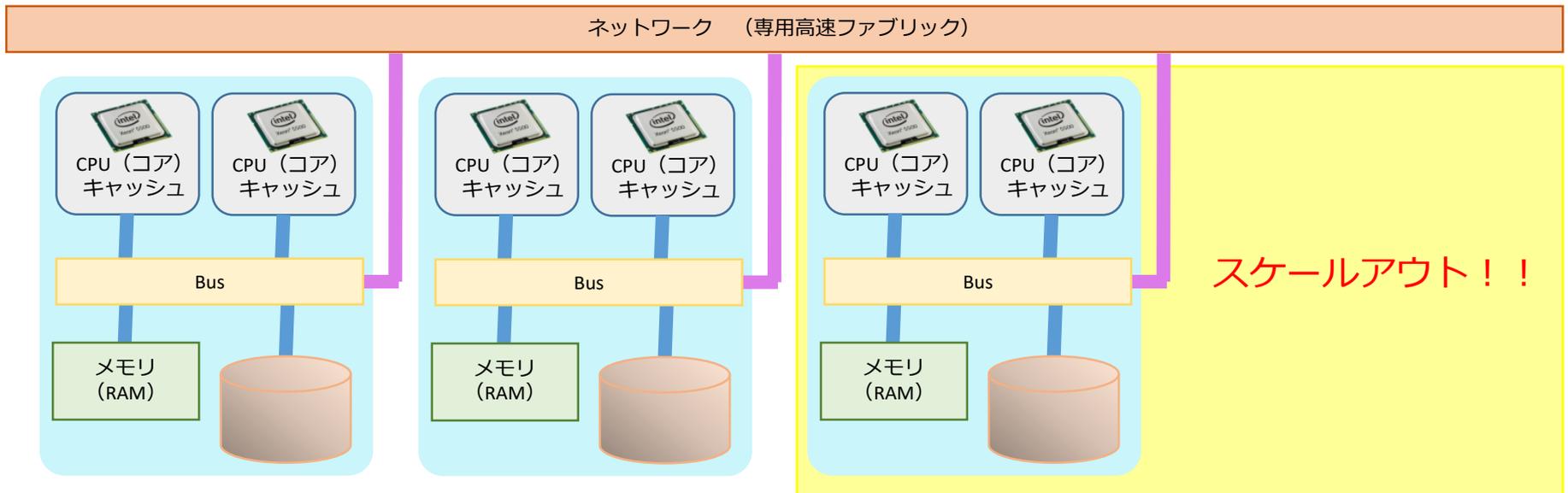


MPP

Massively Parallel Processing

「超並列処理」

- 複数の筐体（ノード）をネットワークで接続して並列計算するクラスターシステム
- それぞれのノードで独立したCPU、メモリ、OSが動作している
- ノードを増設することで処理能力を向上させる

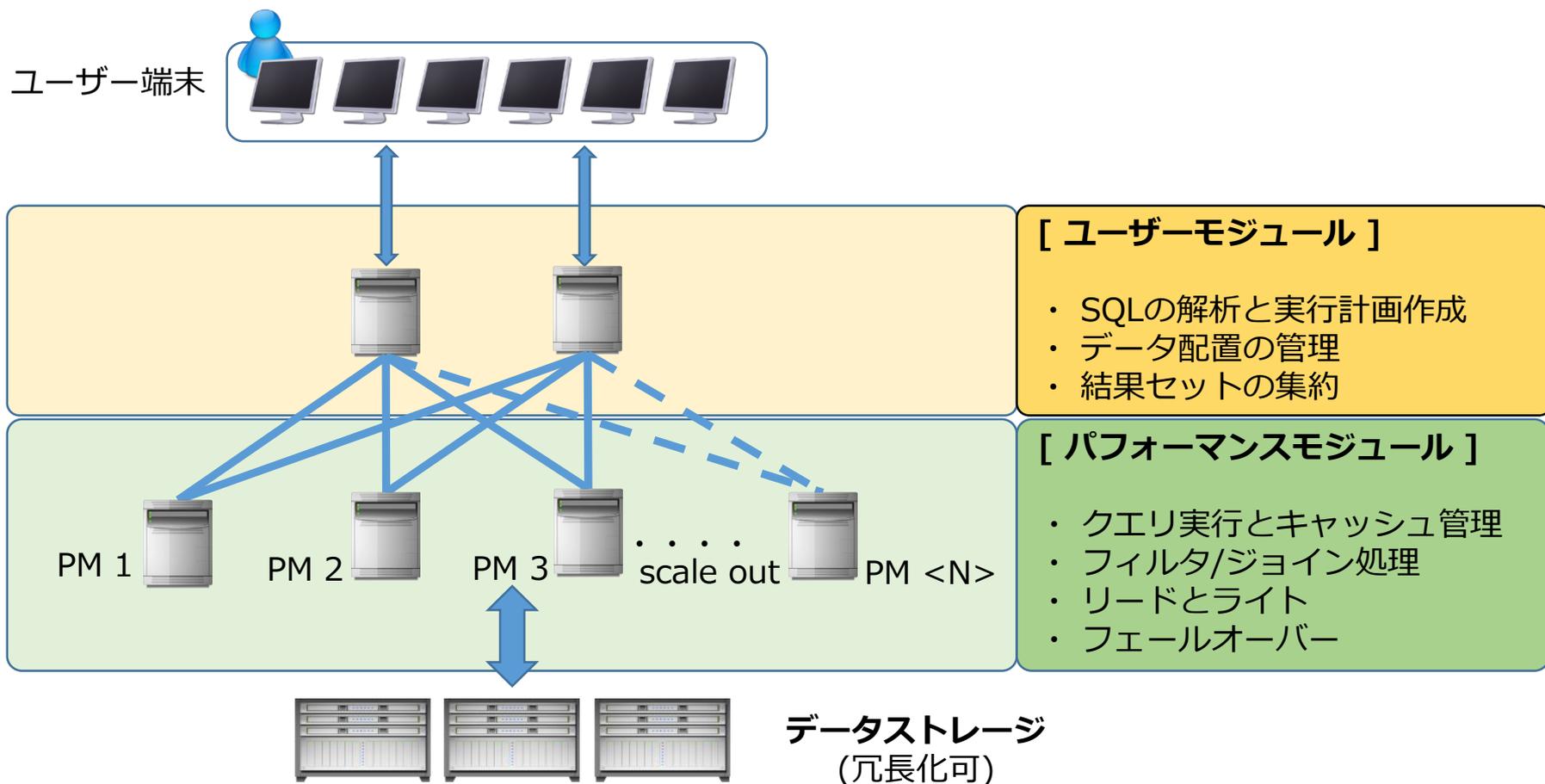


MariaDB Column Storeの特徴 Insight Technology no challenge, no life

- MariaDB ColumnStore はMPP向けに設計されたカラム型ストレージエンジン
- リニアにスケールアウト
- 分析クエリーのリアルタイム処理に優れている
- カラム型ストレージエンジンの効率的なI/O
- データの自動圧縮、解凍

MariaDB ColumnStoreの構成 Insight Technology no challenge, no life

MariaDB Column Store は分散超並列処理(MPP)向けに設計されたカラム型ストレージエンジンで、複数の異なるコンポーネントで構成され、連携して動作します。

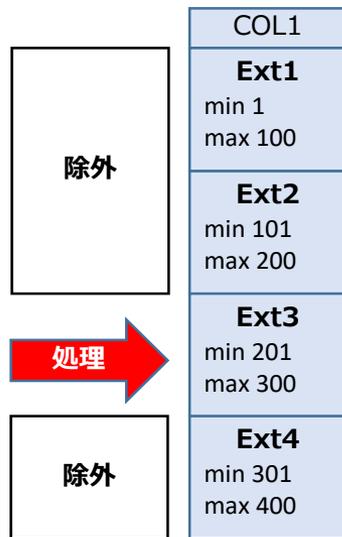


ColumnStoreストレージアーキテクチャ

- ・エクステントは物理的なセグメントファイル内に存在する論理ブロック
- ・エクステントマップにはエクステント及び対応するブロックを管理
- ・データの抽出と配置は、エクステントマップにより高速で処理される
- ・リアルタイム解凍と圧縮
- ・バージョンバッファファイル (UNDO)

— エクステントマップの仕組み —

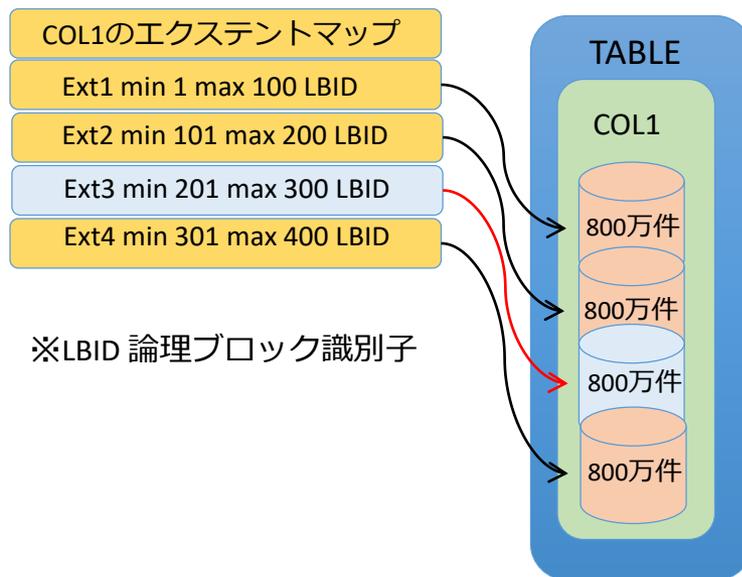
```
SELECT COL1 FROM TABLE  
WHERE COL1 BETWEEN 220 AND 250;
```



COL1フィルター：エクステント1,2,4を除外

エクステント3のみI/Oが発生する。

エクステントマップのイメージ

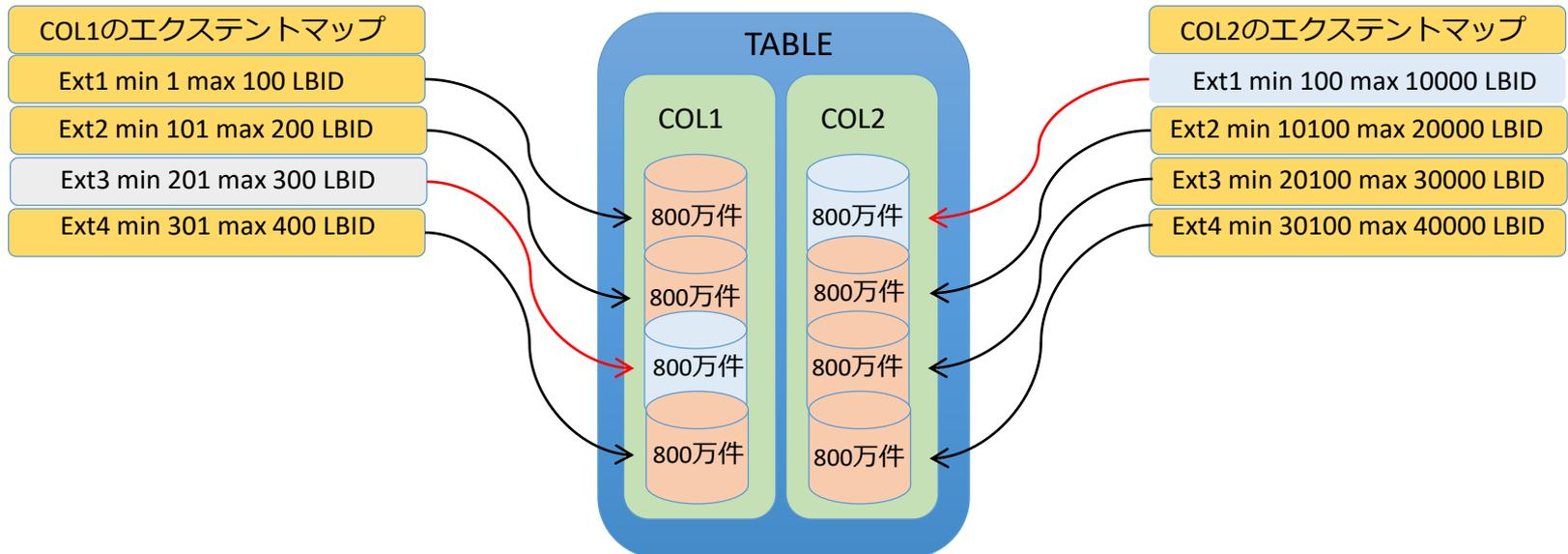


SELECT COL1, COL2 FROM TABLE
 WHERE COL1 BETWEEN 220 AND 250
 AND COL2 < 10000;

COL1フィルター：エクステント1,2,4を除外
 COL2フィルター：エクステント3を除外
 I/Oは発生しない。

除外	COL1	COL2
	Ext1 min 1 max 100	Ext1 min 100 max 10000
	Ext2 min 101 max 200	Ext2 min 10100 max 20000
	Ext3 min 201 max 300	Ext3 min 20100 max 30000
除外	Ext4 min 301 max 400	Ext4 min 30100 max 40000
除外		

エクステントマップのイメージ



利用者TBL

Row	氏名	住所	年齢	性別	利用費
1	横井 雅則	高知県土佐郡土佐町瀬戸5483	33	男	¥24,000
2	山田 裕恵	千葉県山武郡大網白里町小西9-3-3-502号室	42	女	¥12,000
3	長尾 純子	高知県南国市白木谷6-13-1	28	女	¥26,000
4	西田 次男	島根県雲南市三刀屋町六重2-12-100	22	男	¥18,000
5	大城 雅則	福井県坂井市丸岡町上久米田9-1-5	44	男	¥28,000
6	杉本 泰美	新潟県佐渡市新穂井内1-15-1	29	女	¥9,400
7	古田 裕仁	沖縄県中頭郡西原町津花波7203	19	男	¥14,000
8	小倉 真由美	神奈川県横浜市青葉区緑山8-15-5	41	女	¥25,000
9	中島 真知子	京都府京都市左京区新東洞院町1-4-9	55	女	¥9,600
10	荒井 敬志	長野県松本市刈谷原町7-10-9	63	男	¥8,800
11	寺田 仁美	岐阜県瑞浪市和合町6-6-9 アスコットヴィラ和合町 612	52	女	¥5,500
12	白石 彩	東京都千代田区神田西福田町5-10-2	38	女	¥800
13	小西 直秋	茨城県笠間市寺崎7900	55	男	¥1,000
14	細川 一人	神奈川県藤沢市桐原町5-15-7-2階	81	男	¥900
15	大竹 美緒	東京都三鷹市野崎8-12-9	44	女	¥1,200
16	菊池 浩一郎	東京都新宿区単箭町1-15-8	41	男	¥26,000
17	早川 真佐文	長崎県長崎市南町2-2-7	29	男	¥8,800
18	森山 良秋	福島県福島市仲間町3-9-2	24	男	¥6,000
19	齋藤 寛	千葉県長生郡睦沢町森1566	56	男	¥2,500

エクステントマップ

```
select 年齢, 費用 from 利用者TBL
where 年齢 < 30
and 利用費 > 10000
```

Ext1 Min: 19 Max: 44	Ext1 Min: 12000 Max: 28000
Ext2 Min: 41 Max: 63	Ext2 Min: 800 Max: 26000
Ext3 Min: 29 Max: 56	Ext3 Min: 2500 Max: 8800

andなので両方の条件に該当するExt1の範囲のデータ(Row1~8)に対してのみI/Oが発生する。

デモ

- InnoDB
- ColumnStore Single
- ColumnStore PM x 2

デモ環境 : InnoDB

デモマシン

- CPU 24 CORE
- メモリ 48G
- SSD



TPCH 10G
1 User

Query
1, 5, 7, 14, 15, 16

MariaDB
Engine : InnoDB

デモ環境 : ColumnStore Single

デモマシン

- CPU 24 CORE
- メモリ 48G
- SSD



TPCH 10G
1 User

Query
1, 5, 7, 14, 15, 16

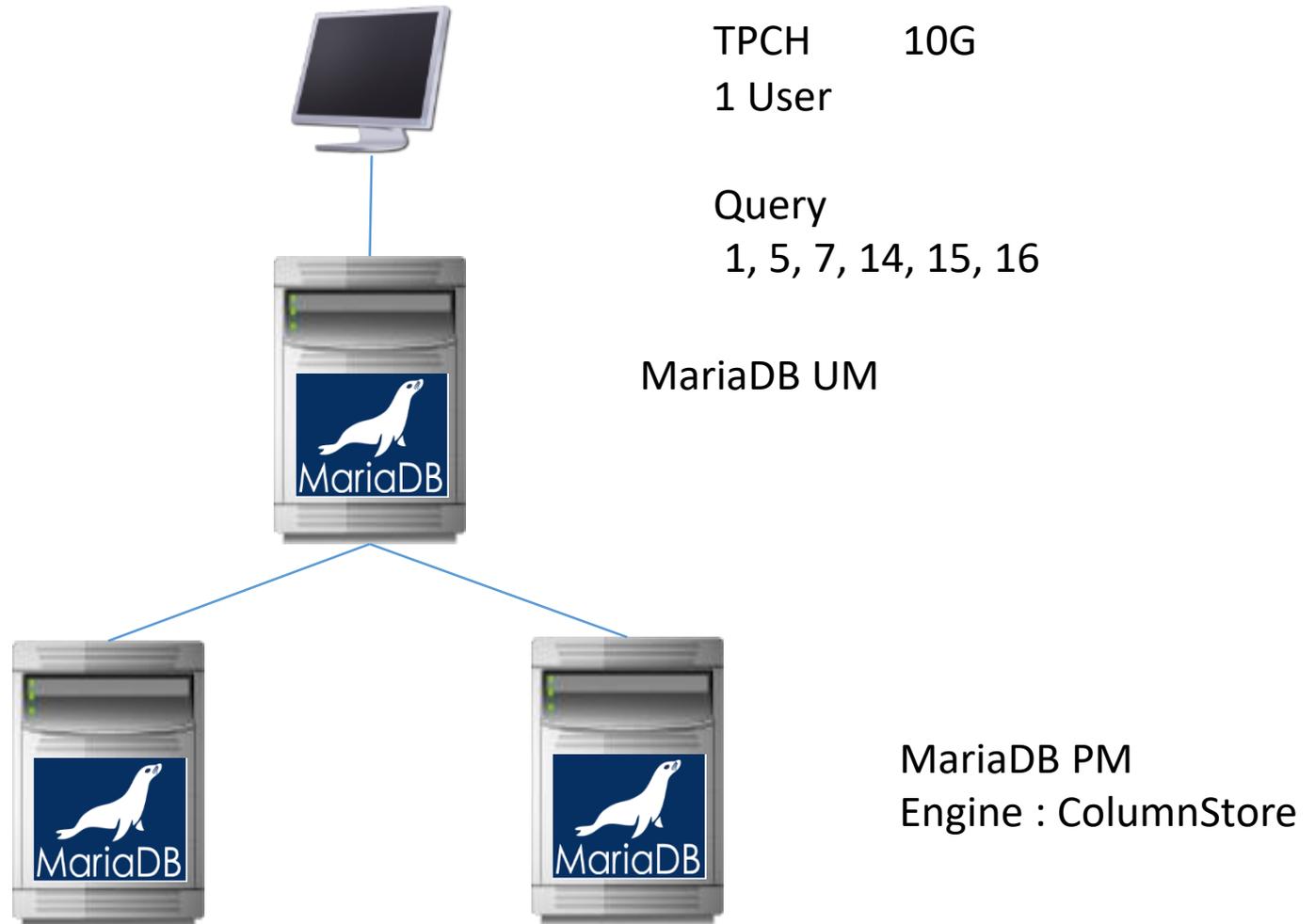


MariaDB UM,PM
Engine : ColumnStore

デモ環境 : ColumnStore PM x 2

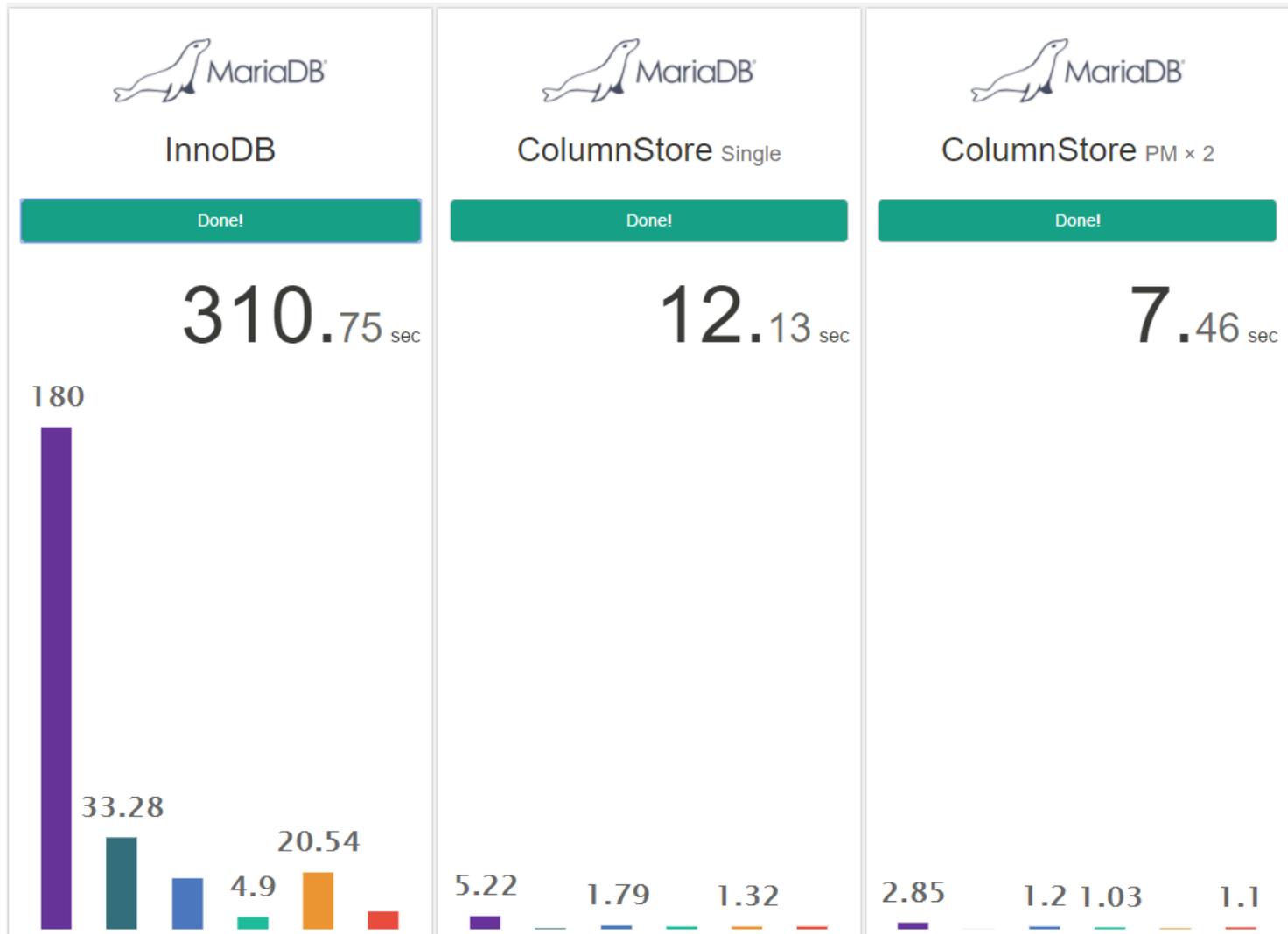
デモマシン

- CPU 24 CORE
- メモリ 48G
- SSD

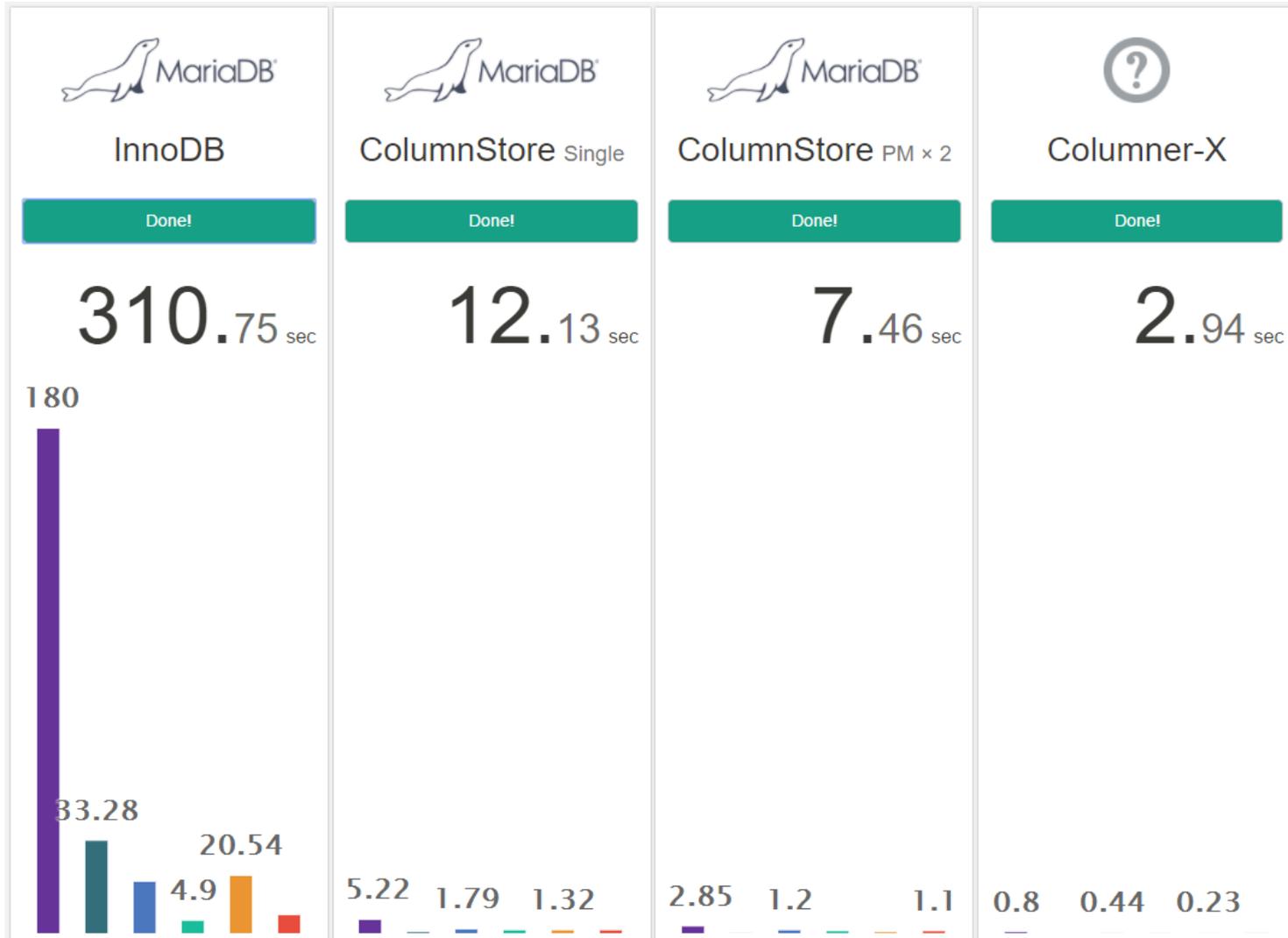


Let's DEMO !!

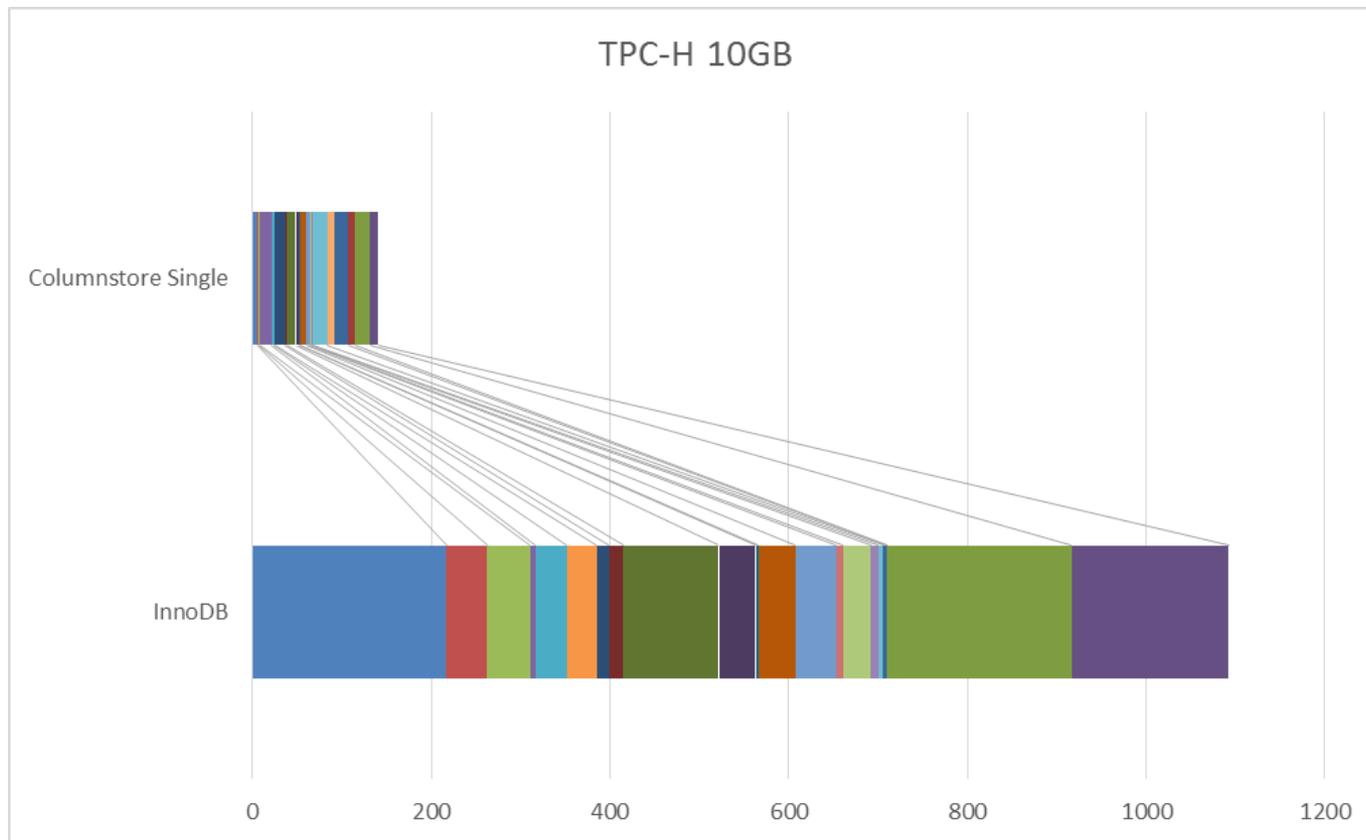
デモ結果 (実行例)



デモ結果 (実行例)

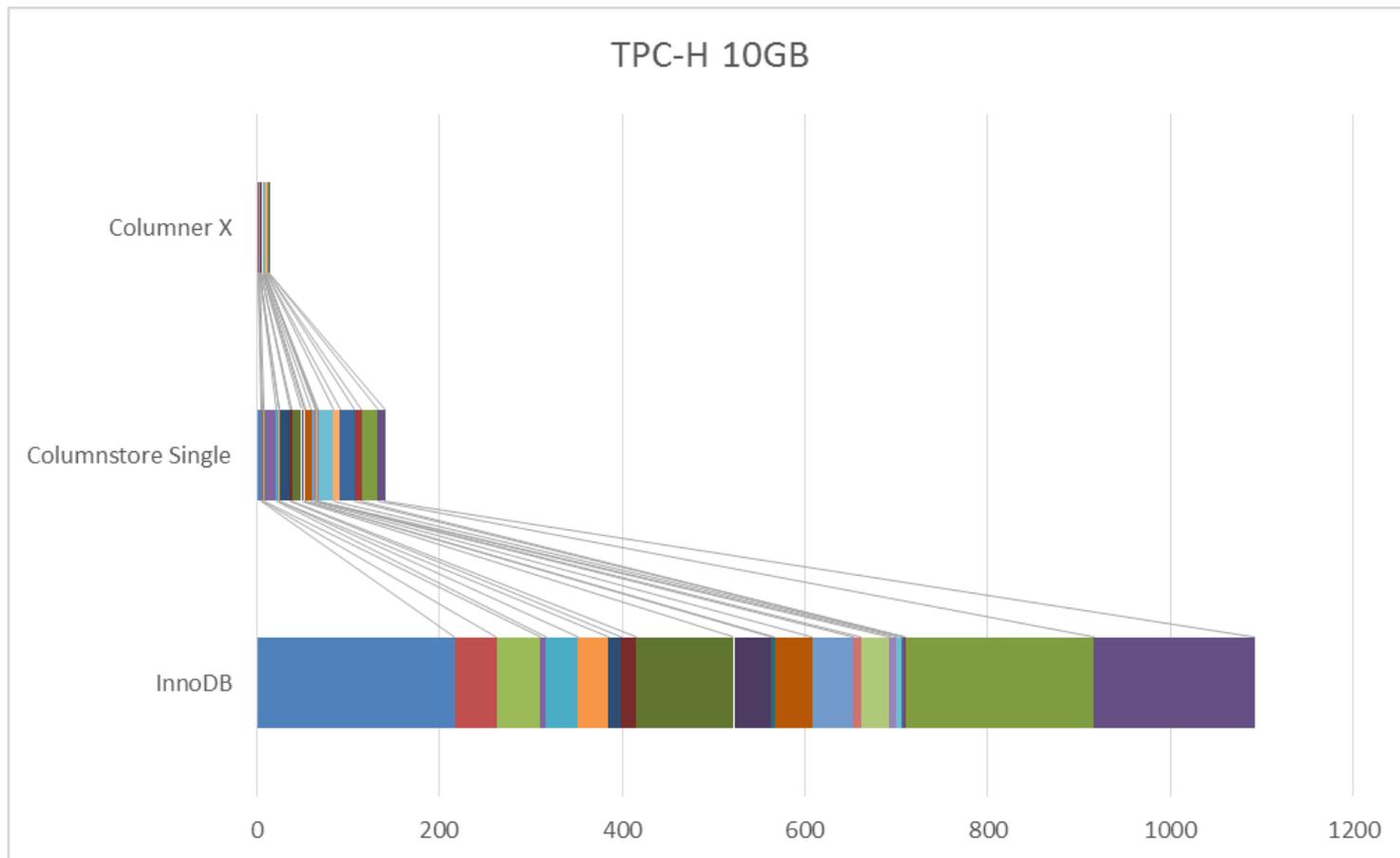


TPC-H 全クエリでの比較



InnoDBではクエリ18が返ってこなかったため除外
ColumnStoreはInnoDBの約8倍！！

TPC-H 全クエリでの比較

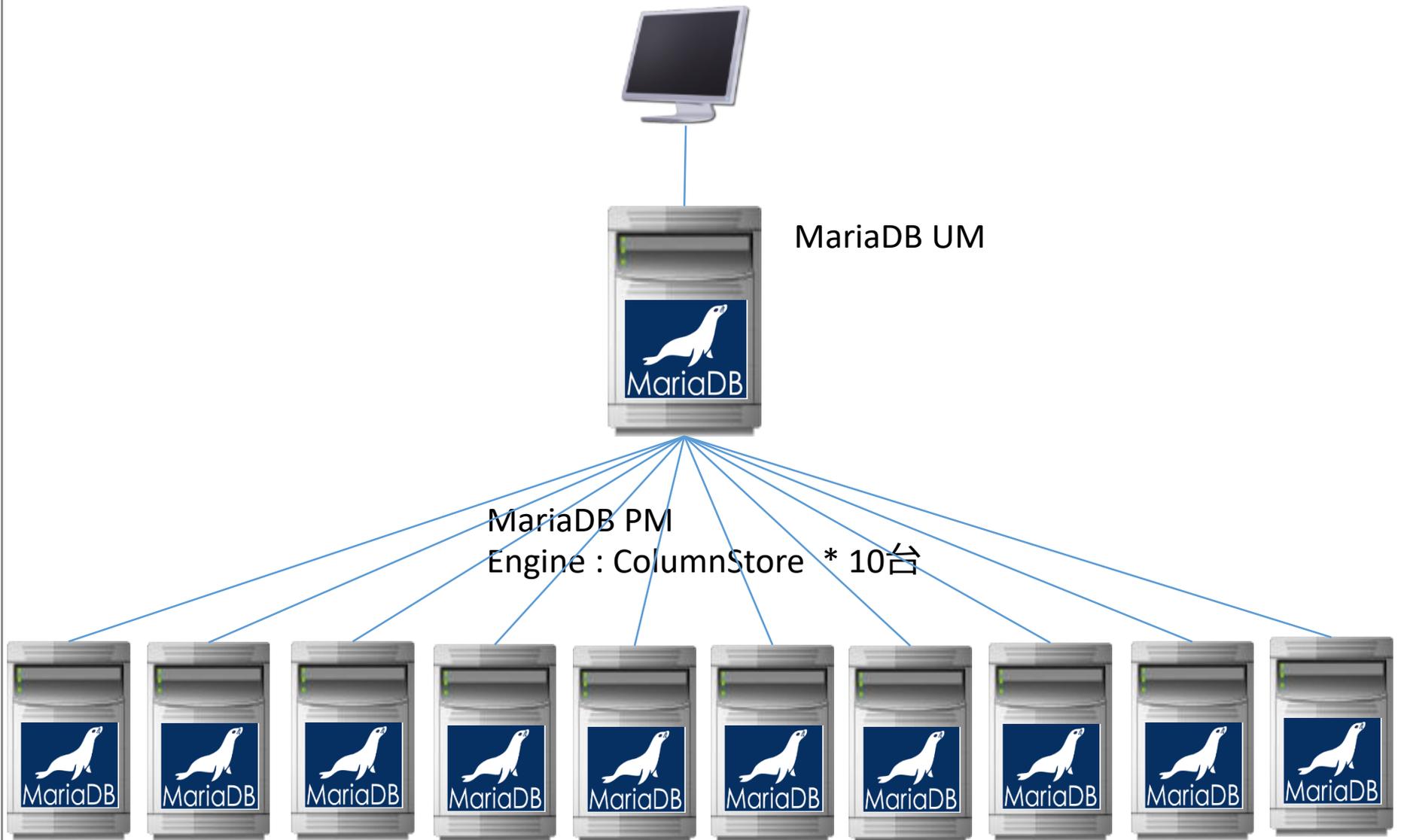


ColumnStoreはInnoDBの約8倍！！

Columnner XはColumnStore Singleの約10倍！！ (InnoDBの約80倍)

(Columnner X とは Actian Vector のこと)

次回は、こんなのをやってみたいな。



Spider ストレージエンジンについて

- Spider は、MariaDB、MySQLのストレージエンジン。
- 他のサーバにある、MariaDB,MySQL,Oracle DB、他のテーブルを、あたかもそのデータベースのにあるテーブルのように利用可能。
- テーブルのパーティションを定義する事により、複数のサーバにデータを分割する事が可能。
- MariaDB 10.0.4 から標準でバンドルされている。

Spiderで出来る事

- フェデレーション
他のサーバのテーブルをローカルサーバのテーブルとして使用可能。
- シャーディング
テーブルを複数のサーバに分散する事が出来る。

フェデレーション

MariaDB
Spider Engine

<- select empno,ename,job,deptno from emp;

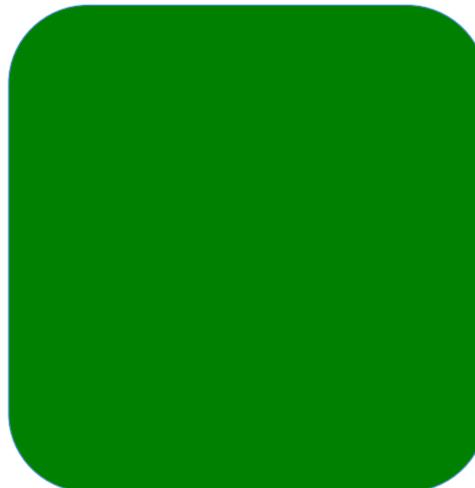
アプリケーションは、1つのデータベースに接続するだけで、データの場所を気にせず利用可能。

EMPNO	ENAME	JOB	DEPTNO
7369	SMITH	CLERK	20
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7566	JONES	MANAGER	20
7654	MARTIN	SALESMAN	30
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7788	SCOTT	ANALYST	20
7839	KING	PRESIDENT	10
7844	TURNER	SALESMAN	30
7876	ADAMS	CLERK	20
7900	JAMES	CLERK	30
7902	FORD	ANALYST	20
7934	MILLER	CLERK	10

MariaDB

EMP			
EMPNO	ENAME	JOB	DEPTNO
7369	SMITH	CLERK	20
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7566	JONES	MANAGER	20
7654	MARTIN	SALESMAN	30
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7788	SCOTT	ANALYST	20
7839	KING	PRESIDENT	10
7844	TURNER	SALESMAN	30
7876	ADAMS	CLERK	20
7900	JAMES	CLERK	30
7902	FORD	ANALYST	20
7934	MILLER	CLERK	10

MariaDB



MariaDB

DEPT		
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

シャーディング

MariaDB +
Spider Engine

<- select empno,ename,job,deptno from emp;

アプリケーションは、1つのデータベースに接続するだけで、全てのデータベースを透過的に利用可能

EMPNO	ENAME	JOB	DEPTNO
7369	SMITH	CLERK	20
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7566	JONES	MANAGER	20
7654	MARTIN	SALESMAN	30
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7788	SCOTT	ANALYST	20
7839	KING	PRESIDENT	10
7844	TURNER	SALESMAN	30
7876	ADAMS	CLERK	20
7900	JAMES	CLERK	30
7902	FORD	ANALYST	20
7934	MILLER	CLERK	10

MariaDB

EMP			
EMPNO	ENAME	JOB	DEPTNO
7782	CLARK	MANAGER	10
7839	KING	PRESIDENT	10
7934	MILLER	CLERK	10

MariaDB

EMP			
EMPNO	ENAME	JOB	DEPTNO
7369	SMITH	CLERK	20
7566	JONES	MANAGER	20
7788	SCOTT	ANALYST	20
7876	ADAMS	CLERK	20
7902	FORD	ANALYST	20

MariaDB

EMP			
EMPNO	ENAME	JOB	DEPTNO
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7654	MARTIN	SALESMAN	30
7698	BLAKE	MANAGER	30
7844	TURNER	SALESMAN	30
7900	JAMES	CLERK	30

シャードッキングでのJOIN

MariaDB +
Spider Engine

```
<- select e.empno,e.ename,e.job,d.deptno,d.dname
from emp e,dept d
where e.deptno=d.deptno;
```

empno	ename	job	deptno	dname
7782	CLARK	MANAGER	10	ACCOUNTING
7839	KING	PRESIDENT	10	ACCOUNTING
7934	MILLER	CLERK	10	ACCOUNTING
7369	SMITH	CLERK	20	RESEARCH
7566	JONES	MANAGER	20	RESEARCH
7788	SCOTT	ANALYST	20	RESEARCH
7876	ADAMS	CLERK	20	RESEARCH
7902	FORD	ANALYST	20	RESEARCH
7499	ALLEN	SALESMAN	30	SALES
7521	WARD	SALESMAN	30	SALES
7654	MARTIN	SALESMAN	30	SALES
7698	BLAKE	MANAGER	30	SALES
7844	TURNER	SALESMAN	30	SALES
7900	JAMES	CLERK	30	SALES

アプリケーションは、1つのデータベースに接続するだけで、全てのデータベースを透過的に利用可能

MariaDB

MariaDB

MariaDB

EMP			
EMPNO	ENAME	JOB	DEPTNO
7782	CLARK	MANAGER	10
7839	KING	PRESIDENT	10
7934	MILLER	CLERK	10

DEPT		
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

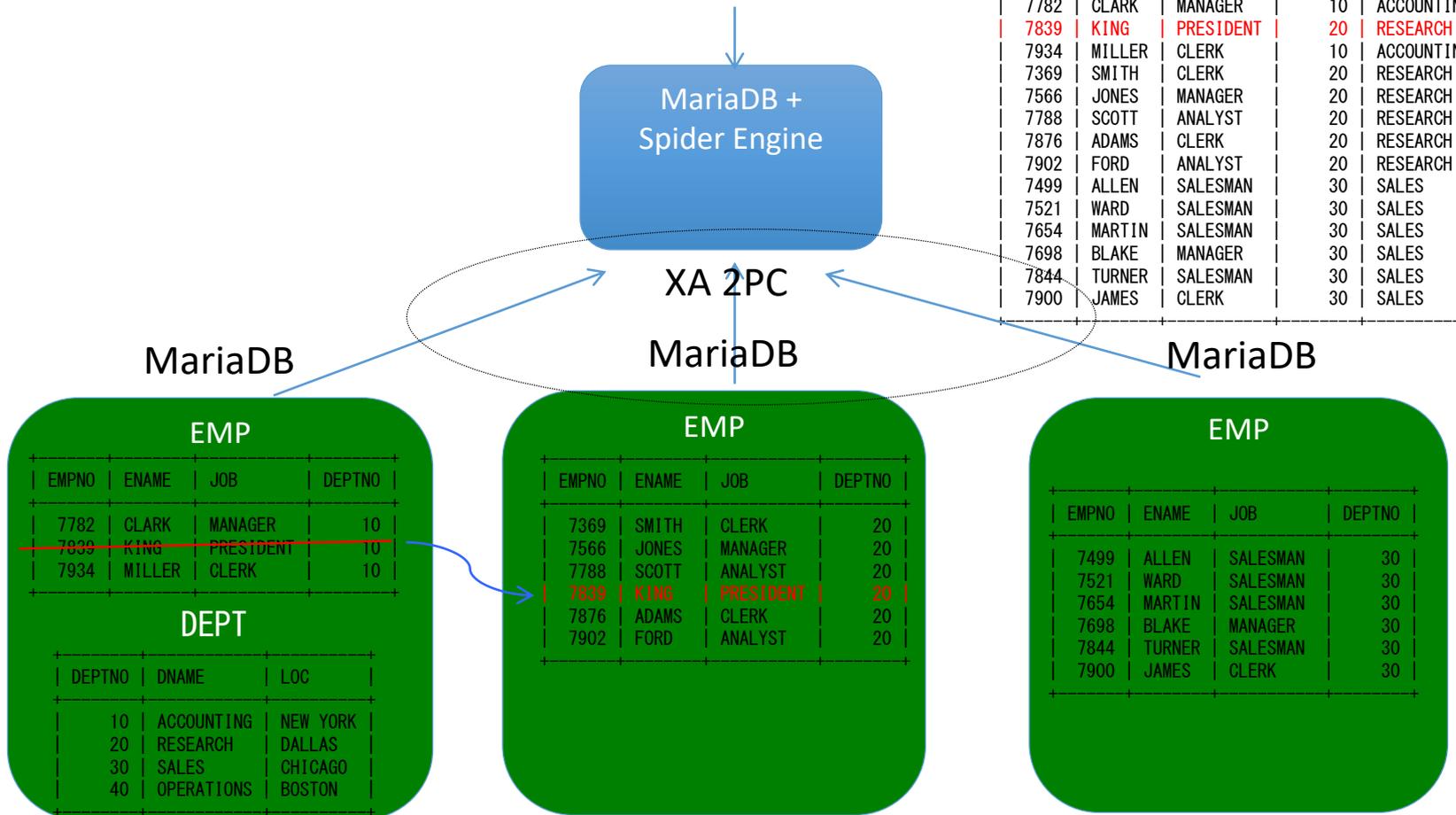
EMP			
EMPNO	ENAME	JOB	DEPTNO
7369	SMITH	CLERK	20
7566	JONES	MANAGER	20
7788	SCOTT	ANALYST	20
7876	ADAMS	CLERK	20
7902	FORD	ANALYST	20

EMP			
EMPNO	ENAME	JOB	DEPTNO
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7654	MARTIN	SALESMAN	30
7698	BLAKE	MANAGER	30
7844	TURNER	SALESMAN	30
7900	JAMES	CLERK	30

データの更新

```
update EMP set DEPTNO = 20 where EMPNO=7839;
commit;
select e.empno,e.ename,e.job,d.deptno,d.loc from emp e,dept d
where e.deptno=d.deptno;
```

empno	ename	job	deptno	dname
7782	CLARK	MANAGER	10	ACCOUNTING
7839	KING	PRESIDENT	20	RESEARCH
7934	MILLER	CLERK	10	ACCOUNTING
7369	SMITH	CLERK	20	RESEARCH
7566	JONES	MANAGER	20	RESEARCH
7788	SCOTT	ANALYST	20	RESEARCH
7876	ADAMS	CLERK	20	RESEARCH
7902	FORD	ANALYST	20	RESEARCH
7499	ALLEN	SALESMAN	30	SALES
7521	WARD	SALESMAN	30	SALES
7654	MARTIN	SALESMAN	30	SALES
7698	BLAKE	MANAGER	30	SALES
7844	TURNER	SALESMAN	30	SALES
7900	JAMES	CLERK	30	SALES

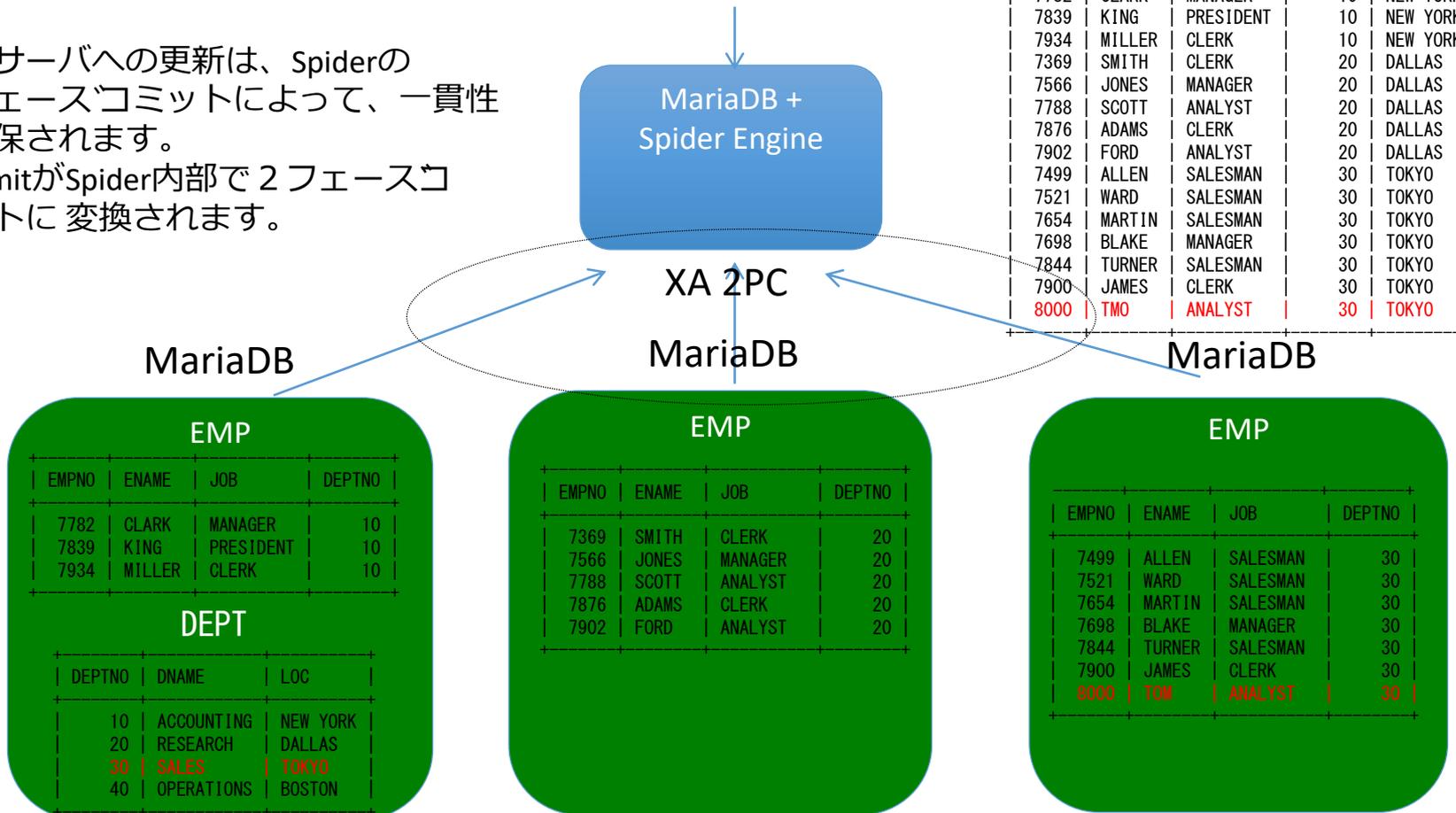


複数サーバへの更新

```
update dept set LOC = 'TOKYO' where deptno=30;
insert into emp values(8000,'TOM','ANALYST',7566,'2017-05-12',1000,100,30);
commit;
select e.empno,e.ename,e.job,d.deptno,d.loc from emp e,dept d
where e.deptno=d.deptno;
```

empno	ename	job	deptno	loc
7782	CLARK	MANAGER	10	NEW YORK
7839	KING	PRESIDENT	10	NEW YORK
7934	MILLER	CLERK	10	NEW YORK
7369	SMITH	CLERK	20	DALLAS
7566	JONES	MANAGER	20	DALLAS
7788	SCOTT	ANALYST	20	DALLAS
7876	ADAMS	CLERK	20	DALLAS
7902	FORD	ANALYST	20	DALLAS
7499	ALLEN	SALESMAN	30	TOKYO
7521	WARD	SALESMAN	30	TOKYO
7654	MARTIN	SALESMAN	30	TOKYO
7698	BLAKE	MANAGER	30	TOKYO
7844	TURNER	SALESMAN	30	TOKYO
7900	JAMES	CLERK	30	TOKYO
8000	TOM	ANALYST	30	TOKYO

複数サーバへの更新は、Spiderの
2フェースコミットによって、一貫性
が担保されます。
CommitがSpider内部で2フェースコ
ミットに変換されます。



Spider の設定例

【サーバ情報の定義】

```
CREATE SERVER spidata1 FOREIGN DATA WRAPPER mysql OPTIONS (USER 'spider', HOST '192.168.XX.100', PORT 3306);  
CREATE SERVER spidata2 FOREIGN DATA WRAPPER mysql OPTIONS (USER 'spider', HOST '192.168.XX.101', PORT 3306);
```

```
MariaDB [mysql]> select * from servers;
```

Server_name	Host	Db	Username	Password	Port	Socket	Wrapper	Owner
spidata1	192.168.XX.100		spider		3306		mysql	
spidata2	192.168.XX.101		spider		3306		mysql	

```
2 rows in set (0.00 sec)
```

【1対1 Spider テーブルの作成】

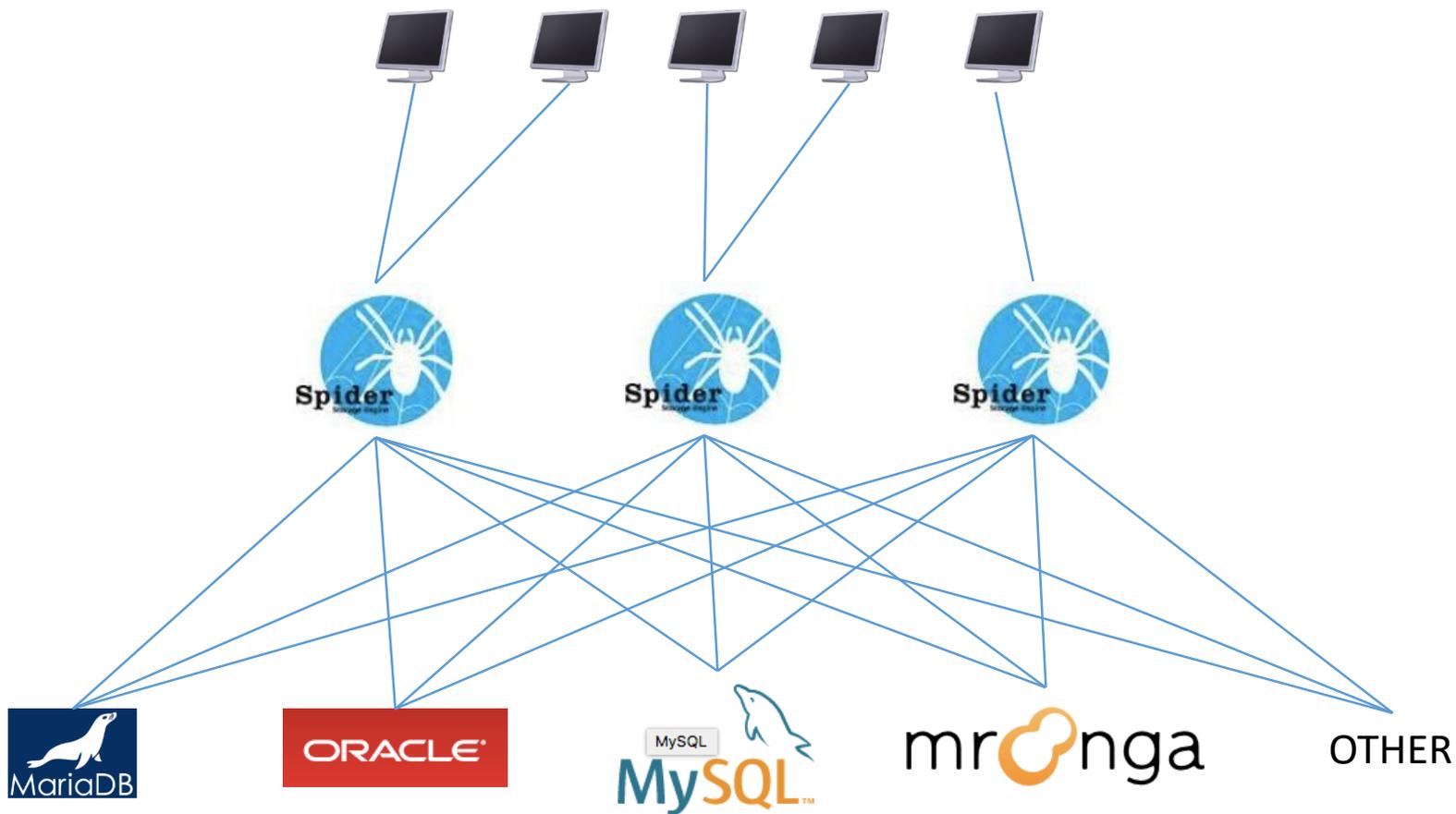
```
CREATE TABLE nation ( N_NATIONKEY INTEGER NOCREATE TABLE nation ( N_NATIONKEY INTEGER NOT NULL,  
N_NAME CHAR(25) NOT NULL,  
N_REGIONKEY INTEGER NOT NULL,  
N_COMMENT VARCHAR(152))  
ENGINE = SPIDER DEFAULT CHARSET=utf8  
comment 'server "spidata1", table "nation" ;
```

Spider の設定例

【1対多（シャーディング） Spider テーブルの作成】

```
CREATE TABLE orders ( O_ORDERKEY      INTEGER NOT NULL,  
                      O_CUSTKEY       INTEGER NOT NULL,  
                      O_ORDERSTATUS   CHAR(1) NOT NULL,  
                      O_TOTALPRICE    DECIMAL(15,2) NOT NULL,  
                      O_ORDERDATE     DATE NOT NULL,  
                      O_ORDERPRIORITY CHAR(15) NOT NULL,  
                      O_CLERK         CHAR(15) NOT NULL,  
                      O_SHIPPRIORITY  INTEGER NOT NULL,  
                      O_COMMENT       VARCHAR(79) NOT NULL)  
ENGINE = SPIDER DEFAULT CHARSET=utf8  
partition by range (O_ORDERKEY)  
(  
  PARTITION P1 VALUES LESS THAN (30000000) comment 'server "spidata1", table "orders",  
  PARTITION P2 VALUES LESS THAN (60000000) comment 'server "spidata2", table "orders" ' );
```

Spiderを使えばこんな事も出来るはず.....



Spiderを使えばこんな事も出来るはず.....



to be continued



OTHER

db tech showcase OSS 2017

JUN 16th - 17th Akihabara



Insight Technology

no challenge, no life

記載されている会社名、サービス名、製品名は、株式会社インサイトテクノロジーおよび各社の商標または登録商標です。

Copyright 2016 Insight Technology, Inc. All Rights Reserved.