

Slerのためのビッグデータ実用セミナーⅢ

～システム開発におけるCassandra適用ケース・導入から運用まで～

セッション4

Cassandraトラブル事案と対処方法

konekto株式会社 / 株式会社INTHEFOREST

Cassandraプロフェッショナルサポート

コネク
株式会社

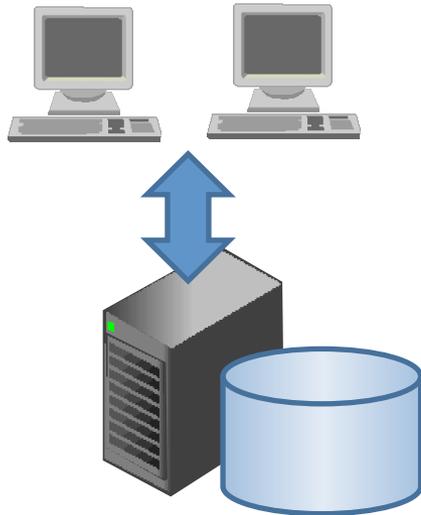
- 営業/契約窓口
- 問合せ窓口(一時対応)
- 他のOSSサポート

株式会社
INTHEFOREST

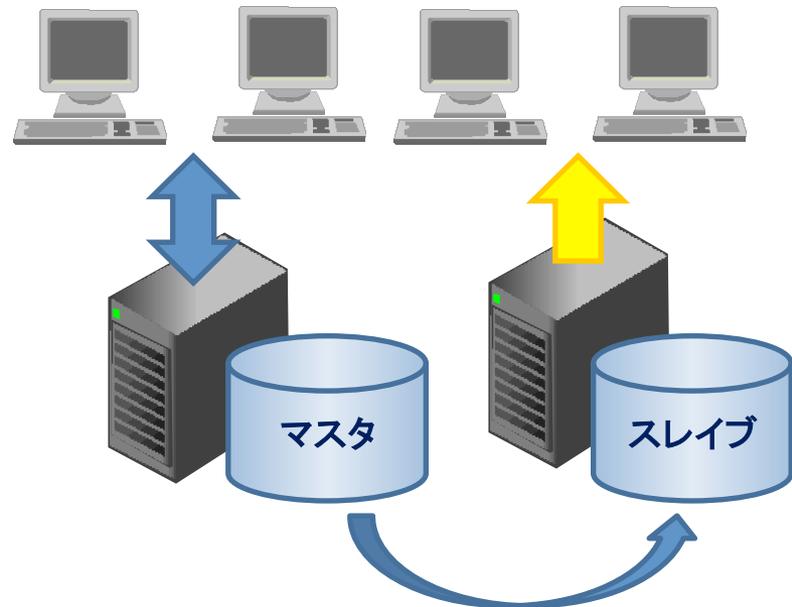
- 技術コンサルティング
- バグ対応・パッチ制作
(オプション)

トポロジーの違い: RDBMS

スタンドアロン



レプリケーション

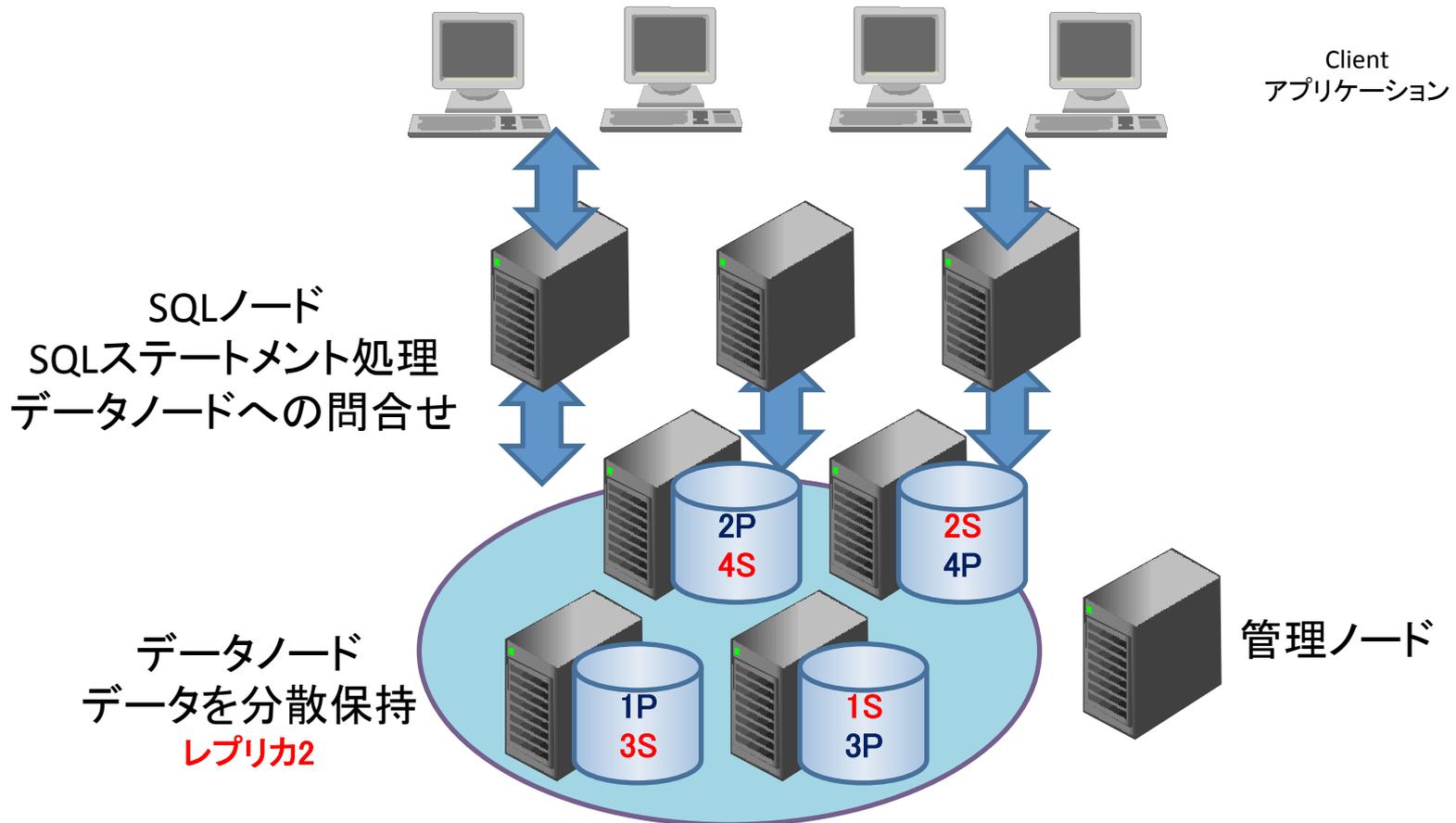


レプリケーション
データの同期

トポロジーの違い: RDBMS Cluster

- 分散クラスタ

 MySQL Cluster



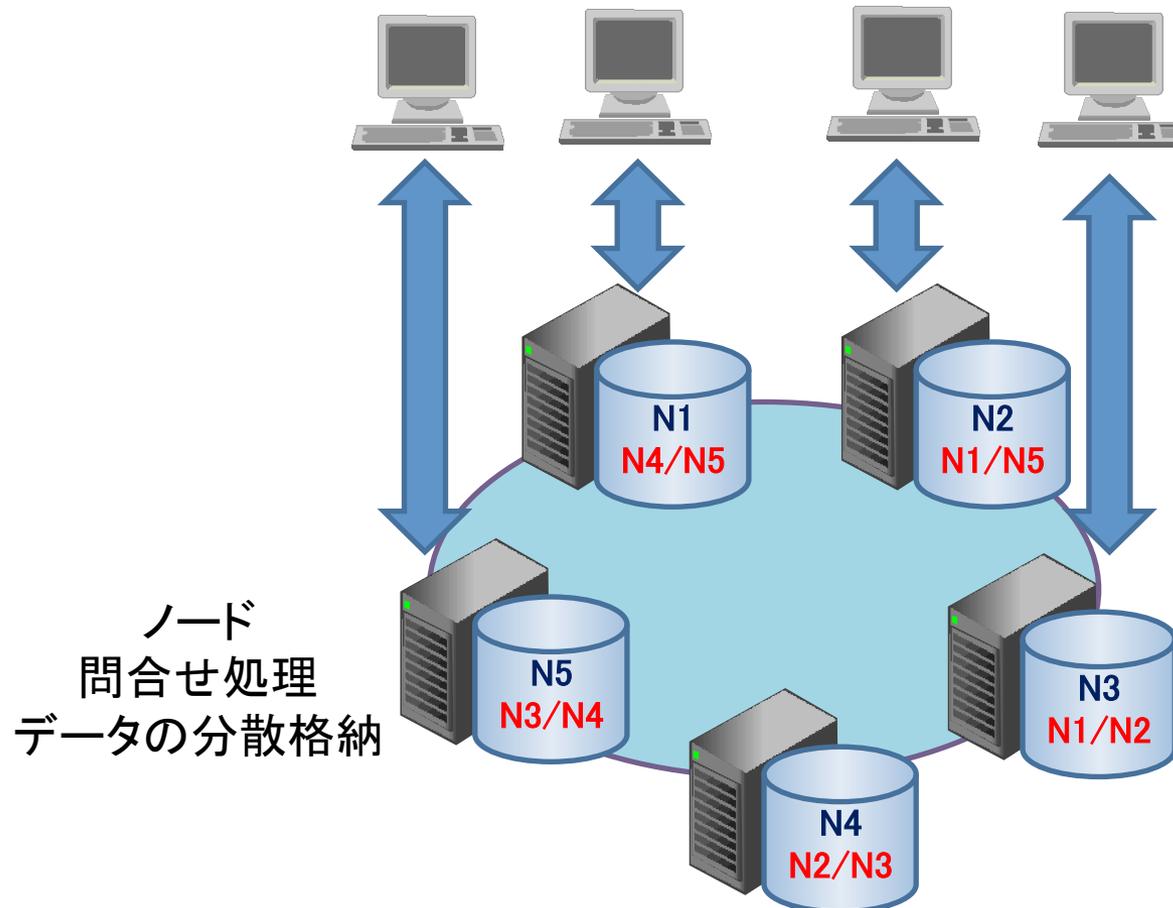
トポロジーの違い: 分散Cluster

- 分散クラスタ



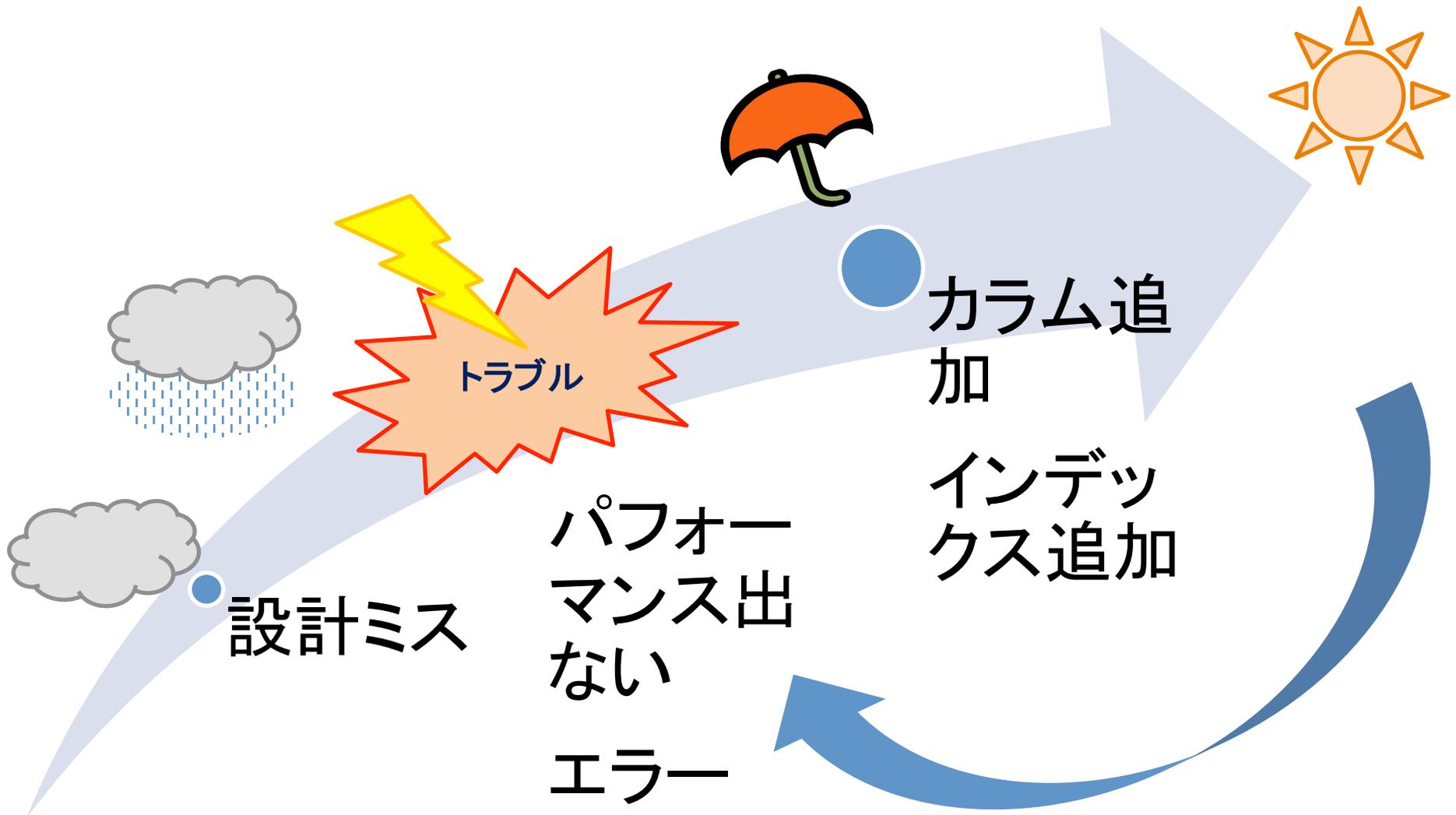
Client
アプリケーション

いずれのノードでも処理可能

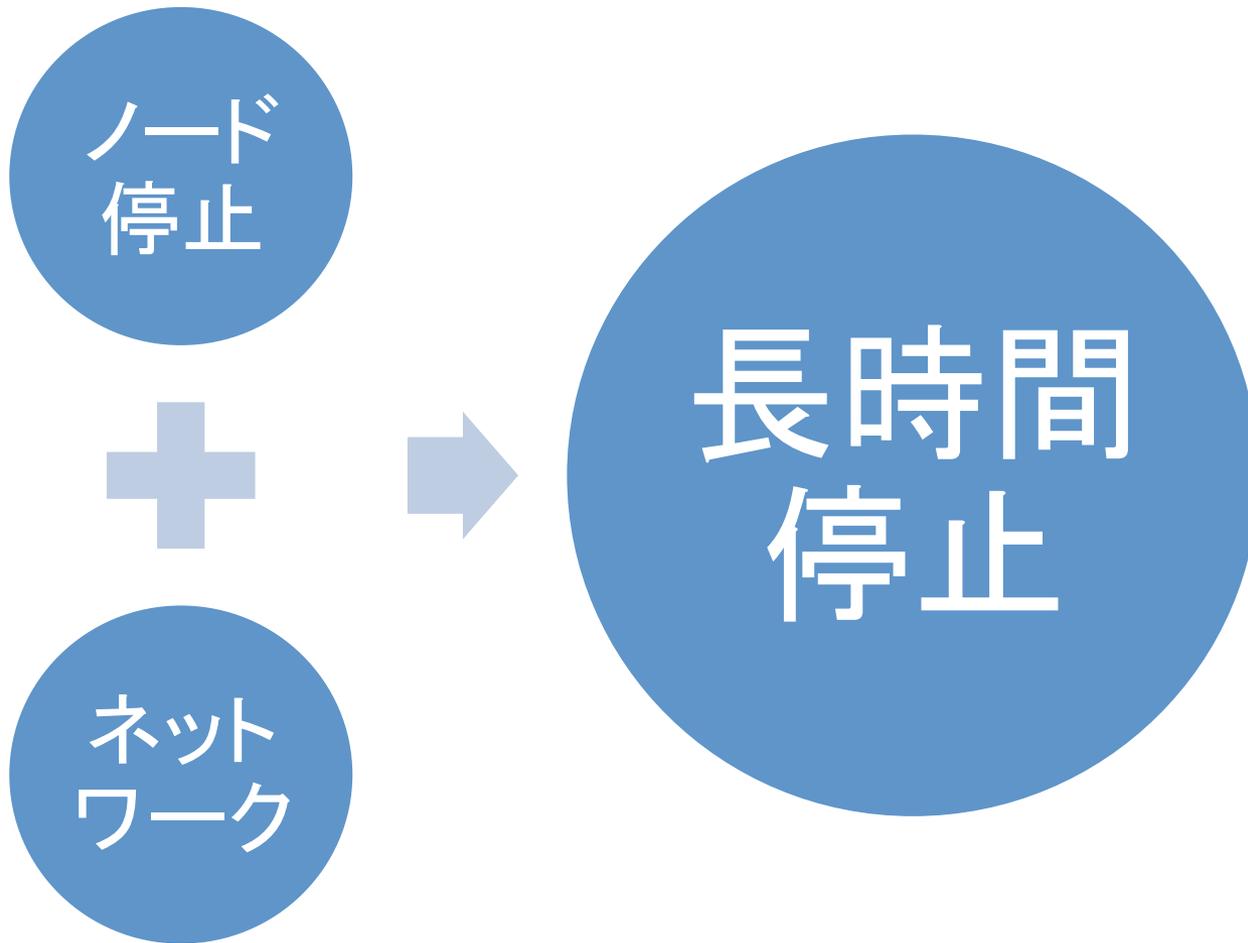


Cassandraのよくあるトラブル

Cassandraあるある



運用トラブル



株式会社INTHEFOREST

トラブルと対処の詳細は

よく起きるトラブル

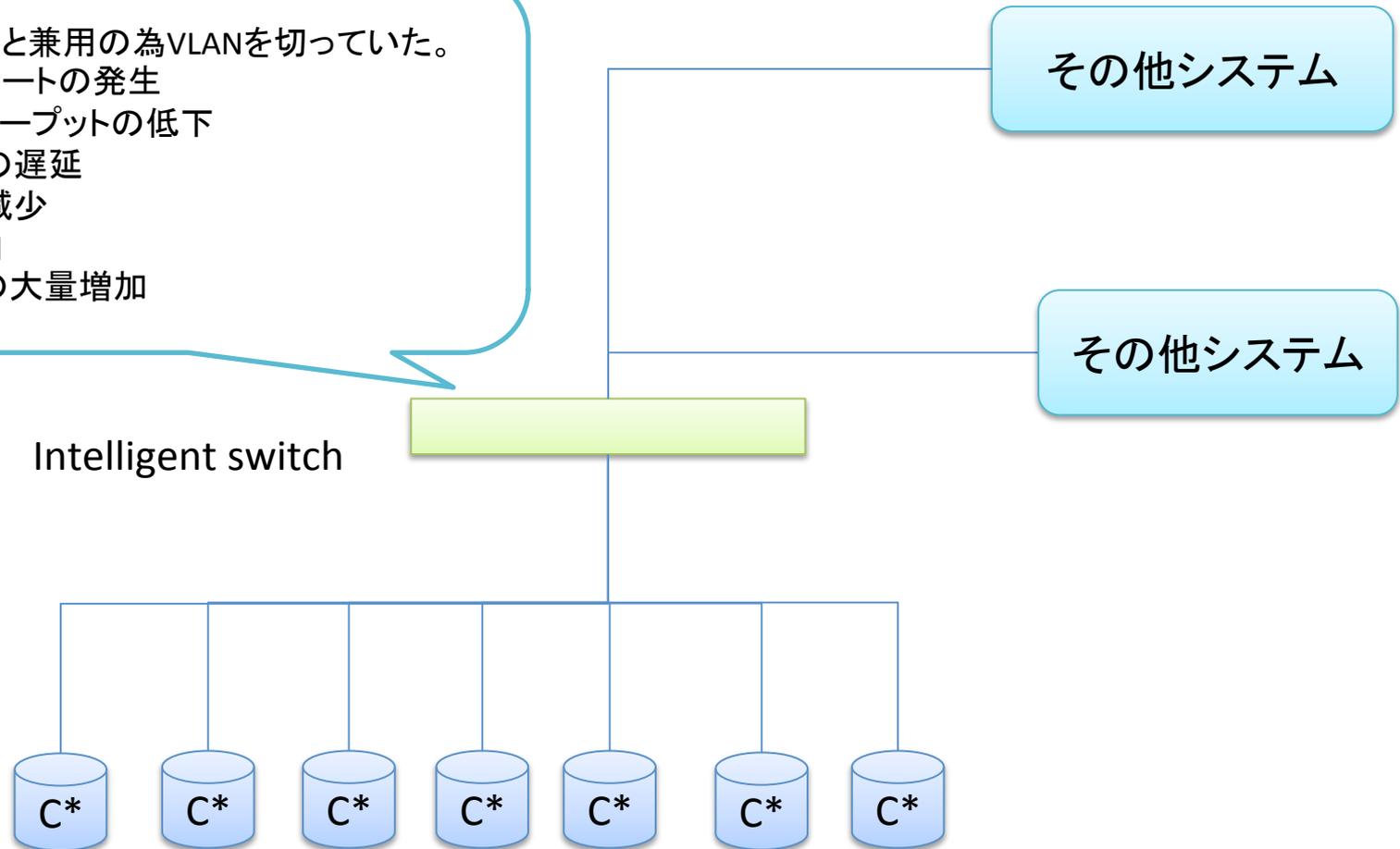
1. ネットワーク障害
2. データ設計障害
 1. 複合キー
 2. セカンダリインデックス

ネットワーク障害

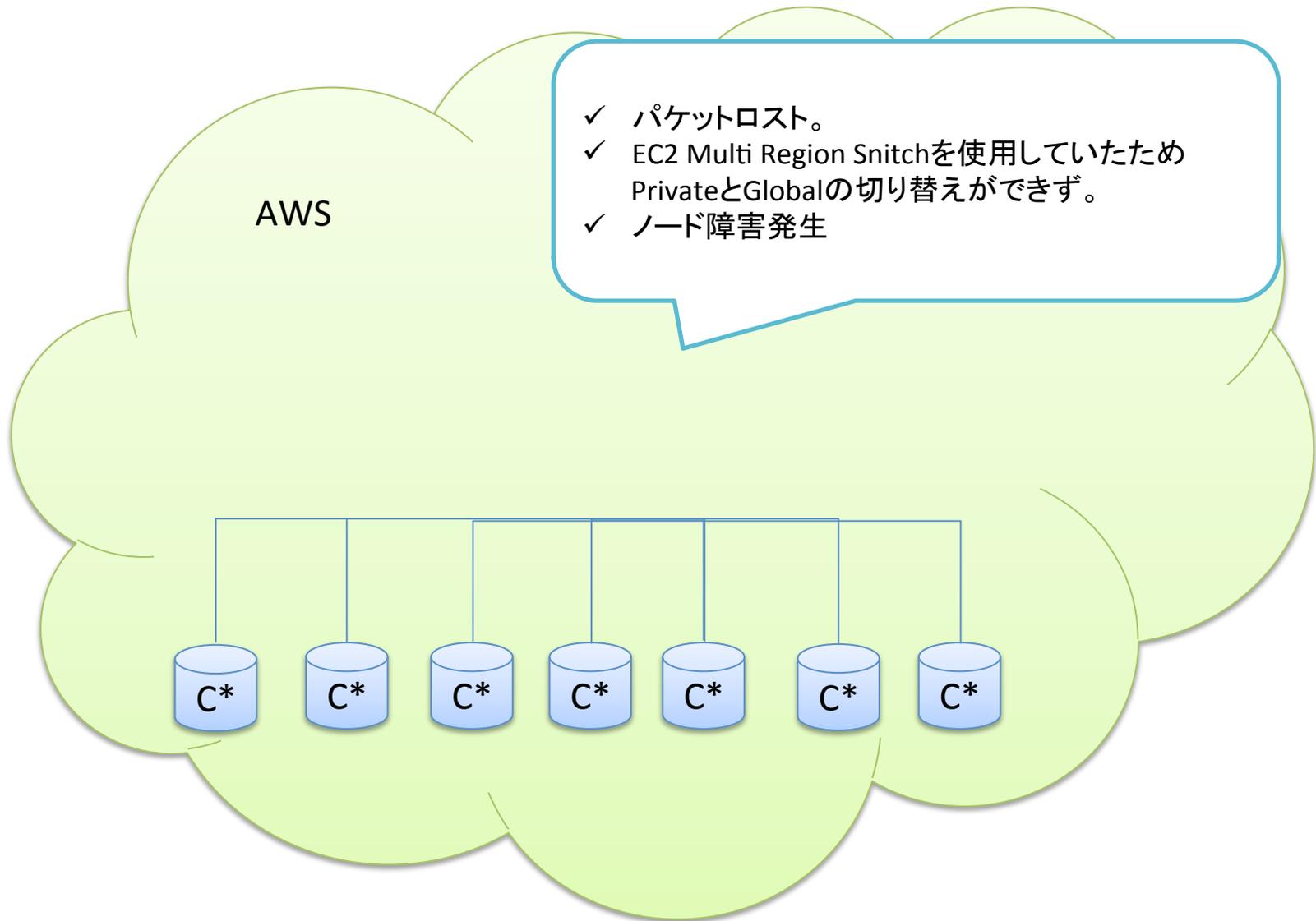
1. Cassandraのネットワーク通信は一定時間応答がないとTimeOutをする。
(1.1迄は一律10秒、1.2以降は処理毎に異なる。)
2. Timeoutした処理はそのプロセスでは対応を行わない。(別のスレッドが対応を行う。)
3. TimeOutが発生してもCLのレベルに達していなければエラーを返さない。

事例①

- ✓ 他のシステムと兼用の為VLANを切っていた。
- ✓ 大量のインサートの発生
- ✓ Switchのスループットの低下
- ✓ ネットワークの遅延
- ✓ Live Nodeの減少
- ✓ Hintedの増加
- ✓ Read Repairの大量増加



事例②



データ設計障害

- 内部実装としてはKVS
- Secondary indexはローカルに配置されたRowKeyレンジ分の転置のCF
- 複合キーはデフォルトではRowKey
+Secondary index
- くくった場合はくくった分の複合RowKey

データ設計障害(複合キー)

テーブル定義

```
CREATE TABLE test (  
  column1 text,  
  column2 text,  
  key1 text,  
  key2 text,  
  key3 text,  
  key4 text,  
  PRIMARY KEY (key1,key2,key3,key4)  
);
```

CQL3でも入れ物は結局
Key Column Value型

```
insert into test (column1, column2, key1, key2, key3, key4) values ('c', 'a', '11', '22', '33', '44');  
insert into test (column1, column2, key1, key2, key3, key4) values ('d', 'f', '11', '22', '33', '55');
```

実態

```
test [key1,11][key2:22,key3:33,key4:44,column1] = c  
test [key1,11][key2:22,key3:33,key4:44,column2] = a  
test [key1,11][key2:22,key3:33,key4:55,column1] = d  
test [key1,11][key2:22,key3:33,key4:55,column2] = f
```

事例①

```
Test[id,1][sequence_number:10001,create_date] = '2013-11-14 09:30:36+0900'  
Test[id,1][sequence_number:10002,create_date] = '2013-11-14 09:30:36+0900'  
test[id,1][sequence_number:10003,create_date] = '2013-11-14 09:30:36+0900'
```

「test」配下のすべてのデータがRowkey「id,1」に存在している。

及びCassandraの分離レベルはRowKeyレベル

データ設計障害 (secondary Index)

事例②

テーブル定義

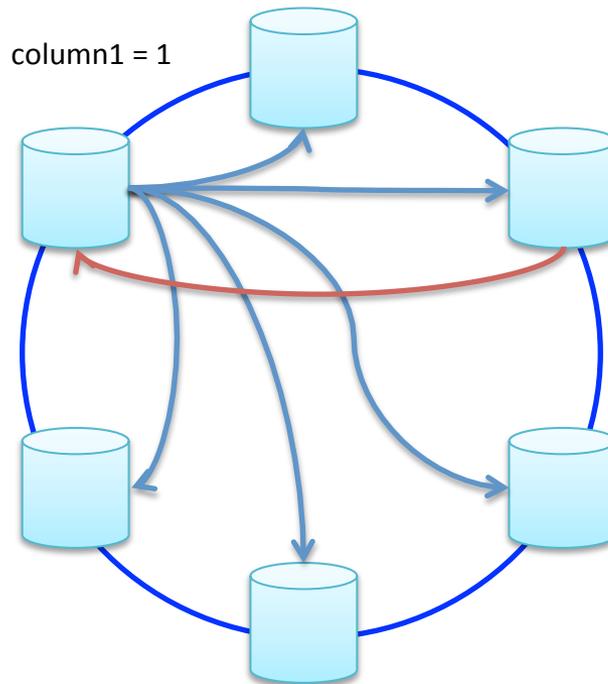
```
CREATE TABLE test2 (  
  column1 int,  
  column2 int,  
  key1 text,  
  PRIMARY KEY (key1)  
);  
create index clm_idx1 on test2 (column1);  
create index clm_idx2 on test2 (column2);
```

```
insert into test2 (column1, column2, key1) values (1,1,'a');  
insert into test2 (column1, column2, key1) values (0,1,'f');
```

データ設計障害 (secondary Index)

Query

```
Select * from test2 where column1 = 1 ;
```

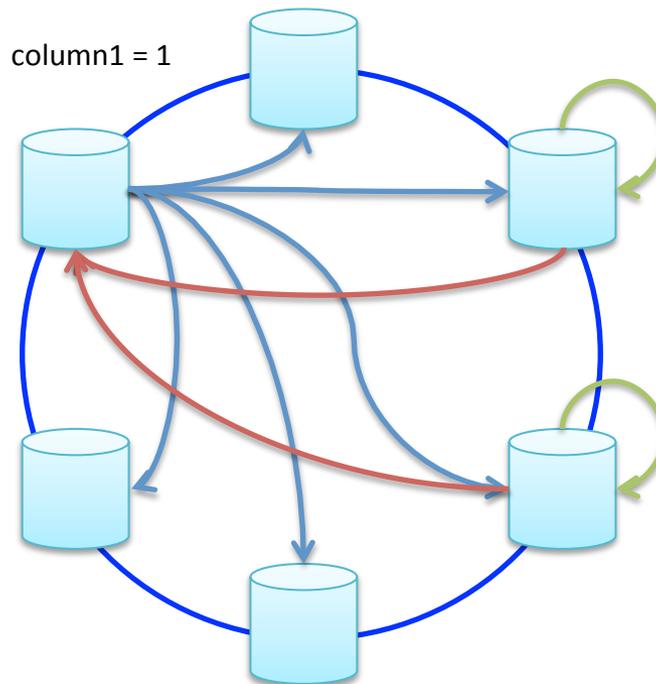


1. すべてのノードに条件句を投げる
2. 各ノードで条件句に適合するデータの抽出
3. データを保持しているノードがデータを返答

データ設計障害 (secondary Index)

Query

```
Select * from test2 where column1 = 1 and column2 > 0;
```



1. すべてのノードに条件句を投げる
2. 各ノードで条件句に適合するデータの抽出
3. 2つ目の条件句で範囲指定
4. データを保持しているノードがデータを返答