

アプリケーション情報を基にした 仮想マシン配置手法の考案

東芝ソリューション株式会社

平原誠也

hirabara.masaya@toshiba-sol.co.jp

開発における問題点

従来のクラウド基盤に対する仮想マシン配置手法は、クラウド基盤のキャパシティ情報を利用したものが中心であった。これらはアプリケーションを構成する仮想マシン群の役割(ロードバランサ、Webサーバ、DBサーバなど)を考慮しないため、アプリケーション個別の非機能要件(可用性、性能など)を満たす仮想マシン配置を行わせることが困難であった。

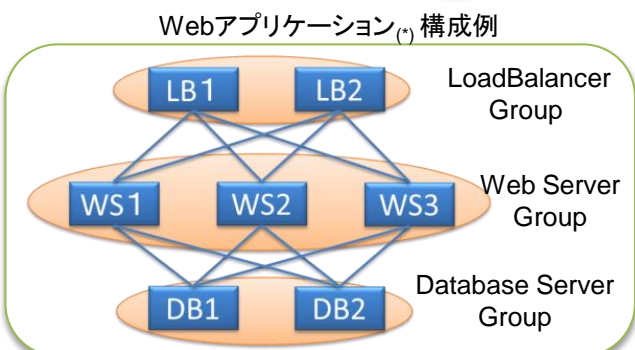
手法・ツールの提案による解決

クラウド基盤における仮想マシン配置問題を数学的制約充足問題(Constraint Satisfaction Problem: CSP)に置き換えて定式化し、アプリケーション個別に実現させたい仮想マシン配置をポリシーとして記述可能にした。
局所探索をベースとした反復修正アルゴリズムを提案し、手法の有効性を評価した。

定式化のアプローチ

① アプリケーションを構成する仮想マシンのうち、配置を制約するものごとにグルーピング

➡ 制約グループ:



(*) 冗長構成。LB, WS, DBが最低1台ずつ動作している必要がある。

② 同一の制約グループに属する仮想マシン同士の配置を制約するためのルールを定義

➡ アプリケーション情報に基づく仮想マシン配置

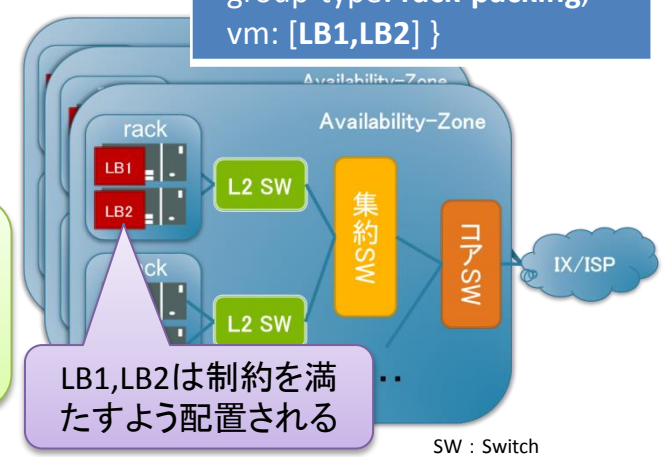
仮想マシンの配置制約

- ・ 同一ホストに集約: host-packing
- ・ 同一ラックに集約: rack-packing
- ・ ラックを分散: rack-slicing
- ・ AZ(*)を分散: az-slicing

(*) AZ (Availability-Zone): 物理的に独立したロケーション

制約グループ定義

```
{ id: LoadBalancer Group,
  group-type: rack-packing,
  vm: [LB1, LB2] }
```



LB1, LB2は制約を満たすよう配置される

SW: Switch

問題を解くアルゴリズム

CSPの局所探索アルゴリズムを応用

1. 既存の仮想マシン配置アルゴリズムで初期割り当て(Greedy, Round-Robin)
2. 制約グループの定義に違反している仮想マシンを選択
3. 選択した仮想マシンの制約違反数を少なくするようホスト割り当てを修正
4. 2.へ戻り、反復修正を行う

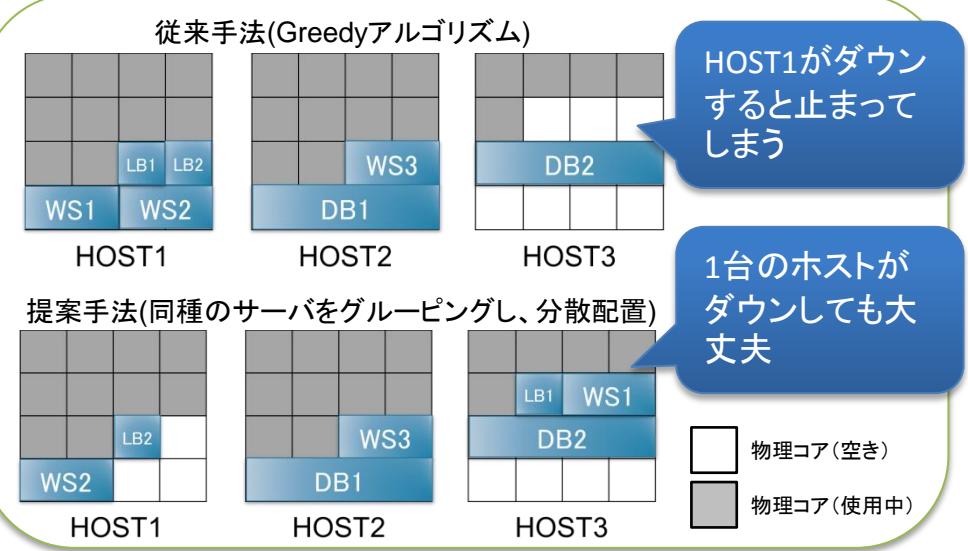


既存アルゴリズムの配置解の状態からスタートし、制約グループの配置制約を満たすように値の割り当てを反復改良

評価

目的に応じた制約グループを定義することによって、最適な仮想マシン配置を実現することが可能となる見込みを得た。

可用性に配慮したWebアプリケーション配置



HOST1がダウンすると止まってしまう

1台のホストがダウンしても大丈夫

□ 物理コア(空き)
■ 物理コア(使用中)