



るアルゴリズムの設定を行う。

また本システムで設計最適化問題を解くのにあたっては、解候補として最適な設計パラメータを設定する「最適化」と、設計パラメータをもとに計算シミュレーションを実行する「制御」を非同期的に行うことを想定しているの、それらの役割をそれぞれ「最適化エンジン」と「制御システム」が担う。これらはインタークラウドで生成された仮想マシンによって動作する。「最適化エンジン」では、解情報データベースから過去にシミュレーションを行った設計情報を取得し、新たな解候補となる設計パラメータの集合を生成し、生成した設計パラメータを評価値と付随情報が未定義の状態としてデータベースに登録する。また新たな設計パラメータの生成に際してその設計パラメータが解空間においてどれだけ最適解に近いかの目安となる値を算出し、「優先度」として制御システムが計算シミュレーションを行う順番を決定するための基準として用いる。「制御システム」では、評価値が未定義となっている設計パラメータを取得し、優先度情報を基に計算シミュレーションの実行をシミュレータにリクエストする。

## 2.2 解情報データベースの webAPI の設計

各システム要素が連携を行うために、解情報データベースに REST 形式の web サービスでアクセスするための webAPI サーバーを構築した。この webAPI は PHP によって動作しており、HTML プロトコルでアクセスすることによって指定した情報を XML 形式で取得することができる。アクセスする際には POST メソッドのパラメータによってユーザーのログイン情報、処理を行う問題の選択、各 API 毎の必要な情報を指定する必要がある。

また現在、スケーラブルな解情報データベースを実現するための分散データベースとして Apache Cassandra[2] を利用しているが、これはシンプルな書き込みや読み込みリクエストだけに対応している。最適化エンジンや制御システムとの連携をするために豊富な書き込み、読み込み機能が必要となるため以下の様な web サービス API を開発した。

### 過去の解候補の取得：getSolutinCandidates

設計パラメータを POST メソッドのパラメータで指定してアクセスすると、その設計パラメータに一致する解候補の評価値などの設計情報を返す。主に最適化エンジンが過去にシミュレーションを行った設計情報を取得して、それらを基に新たなパラメータの生成をするために利用する。オプションとして POST メソッドのパラメータで取得する評価値の個数の指定と、設計パラメータの値を指定しないランダムな取得を行うことを指定する。

### 新たな解候補の保存：postSolutionCandidates

設計パラメータを POST メソッドのパラメータで指定し

てアクセスすると、解情報データベースに新たな解候補を書き込む。分散データベースへ設計パラメータの書き込みと同時に、シミュレーションを行う順番の基準となる優先度情報や解候補の総数などのインデックスの更新を行う。

### 優先度の高い解候補を取得：getTopPriority

この webAPI にアクセスすると、解情報データベースにある優先度情報が参照され、計算シミュレーションの優先度が最も高い設計パラメータを返す。制御システムがシミュレータにシミュレーションをリクエストする際に、リクエストする設計パラメータの取得に利用する。

### シミュレーションの結果を保存：postEvaluationValue

設計パラメータとその解候補の評価値を POST メソッドのパラメータで指定してアクセスすると、解情報データベースに存在するその解候補の評価値を更新する。制御システムがシミュレータから結果を取得した後に、解情報データベースにその結果を保存するために利用する。

これらの webAPI は最適化エンジンもしくは制御システムと、Cassandra データベース間でのデータのやり取りの間で、一度 webAPI サーバー上においてデータを処理、整形することによって実現している。例えば「過去の解候補の取得」でランダムに解候補を取得する場合は、解候補となるデータをまとめて Cassandra データベースから読み出し、指定された個数をランダムに抽出して XML 形式に整形してクライアントに返す。こういった処理によって解情報データベースとその他のシステム要素での通信におけるデータの総量を削減し、結果的にインタークラウドのネットワーク上において通信されるデータの総量の削減を目指している。

## 3. 今後の課題

現在、分散データベースとして Apache Cassandra[2]、クラウド管理ミドルウェアとして Apache CloudStack[3] を用いて、プロトタイプシステムの構築を行っている。その後、新たなパラメータの生成に複数のアルゴリズムを実装してサンプル問題を適用し、実用上の問題点や性能を検証する予定である。

今後の課題として、実際の設計問題でこのシステムを用いる場合を想定した設計の改善や、ユーザーインターフェースに関する評価検討等が挙げられる。

### 参考文献

- [1] 棟朝雅晴：インタークラウド環境における大規模分散設計最適化フレームワークに関する検討，情報処理学会研究報告，Vol.2014-MPS-98，No.28，pp.1-2（2014）
- [2] Apache Cassandra，入手先<<http://cassandra.apache.org>>
- [3] Apache CloudStack，入手先<<http://cloudstack.apache.org>>