B-010

# リアルオブジェクトを対象とするシミュレータの時刻同期
# Synchronization of the simulator which communicates with real objects

加藤　靖之†　　平中　幸雄†　　武田　利浩†
Yasuyuki Kato　　Yukio Hiranaka　　Toshihiro Taketa

## 1. はじめに

近年ネットワークは急速に発展し、SNS やクラウドコンピューティングなどのネットワークを使用した大規模なサービスが普及している。それに伴いネットワークシステムは大規模化、複雑化が進んでいる。ネットワークシステムの開発は、テスト段階においてネットワークシミュレータが使用される。シミュレーター用にプログラムを作成し、実装用プログラムも作成する必要があり、時間が掛かり実際の動作との違いが出る。また、実際にテスト用ネットーワークを構築し、実装用プログラムのテストを行う場合、コストが多大になりやすい。

そこで、シミュレータ上でシステム開発ができ、そのまま実装できる仕組みを開発したい。そのシミュレータは実装するソフトウェア(本稿では、このようなソフトウェアを開発中のソフトウェア(RS)と呼ぶ)を組み込みシミュレーションできる必要がある。また、実際にあるサーバー(本稿ではリアルオブジェクト(RO)と呼ぶ)と通信を行えるシミュレータにしたい。ここで問題点として実時計とシミュレーション時計の同期がある。シミュレーション時計は、実際の時間より速く進んだり、遅く進んだりする。そこで、時刻同期を仮想マシン上で行う方法を提案する。

## 2. リアルオブジェクトを対象とするシミュレータ

リアルオブジェクトを対象とするシミュレータの概略を図 1 に示す。

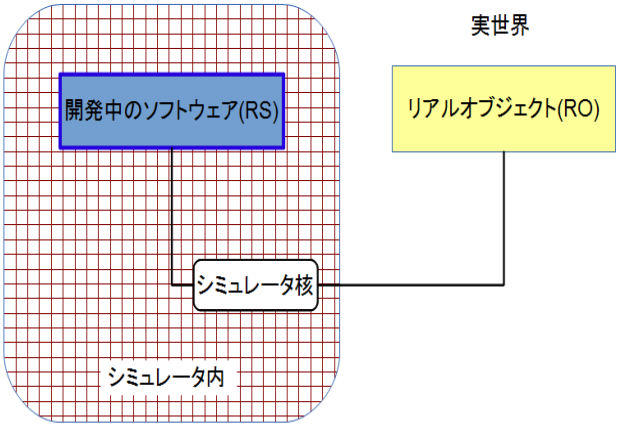


図 1: リアルオブジェクトを対象とするシミュレータ

開発中のソフトウェアはシミュレータ核を通してリアルオブジェクトと通信を行う。シミュレータ核はシナリオスクリプトからシミュレーションシナリオを読み込み通信の順序制御、シミュレーションログの取得の機能をもつ。また,シミュレータは実時計とは進み方が異なる独立したシミュレーション時計をもっている。ここで問題となるのは、リアルオブジェクトが従う実時計とシミュレーション時計の進む速度が違う場合である。シミュレーション時計が実時計より速い場合、RS が送信したメッセージのタイムスタンプが RO が受け取る時刻より、進んだ時刻になってしまい、通信の順序が入れ替わったり、通信自体が破綻する可能性がある。

## 3. 時刻管理

時刻を管理し時刻を同期させる方法として、進んでいる方の時計をロールバックし遅い方の時計の時刻まで戻す方法、遅い方の時計が速い方の時計に追いつくまで、速い方の時計を止める方法の 2 つがある。本研究では、メッセージのやり取りを行うため、ロールバックではメッセージのキャンセルが必要になるので、速い方の時計を停止させる方法を使用する。

### 3.1 VMware Workstation

本研究では、計算機の時計を停止させるので、開発中のソフトウェア、リアルオブジェクトをそれぞれ仮想マシン(VMwareWorkstation)上で動作させる。特徴として、ホスト OS と仮想 OS のクロック同期を完全非同期化することができ、VIXAPI を使用することで仮想マシンをホスト OS から制御することができることが挙げられる。また、クロックを完全非同期化することで、仮想マシンの時計を一時停止、再開を行う際、仮想 OS の時計がホスト OS の時計と同期されない。VIXAPI は仮想マシンの起動、終了、一時停止などが行える。

#### 3.1.1VIX　API

VIX API では以下のような関数が用意されている。

start：仮想マシンを起動、再開させる
stop：仮想マシンをパワーオフにする
suspend:仮想マシンを一時停止させる
pause:仮想マシンをスリープさせる
unpause:仮想マシンをスリープから復帰させる

仮想マシンを一時停止するものは、pause コマンドと suspend コマンドがあるが、pause コマンドは、再開した際に一時停止した時刻ではなく、停止中に経過していた時間分進んだ時刻になってしまう。そこで、今回は suspend コマンドを使用する。

† 山形大学　Yamagata University

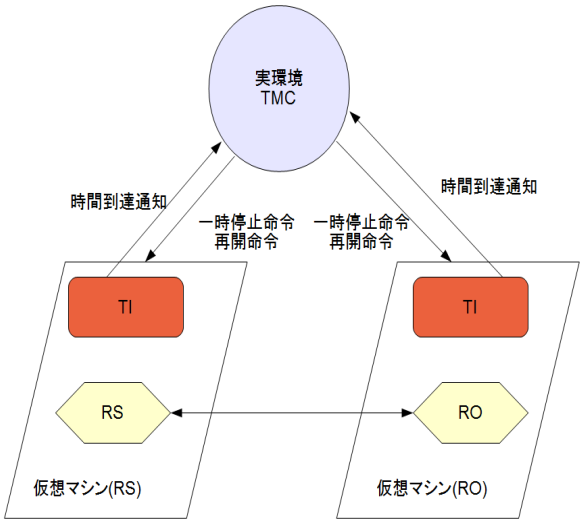


図 2: 時刻管理オブジェクトの構成と流れ

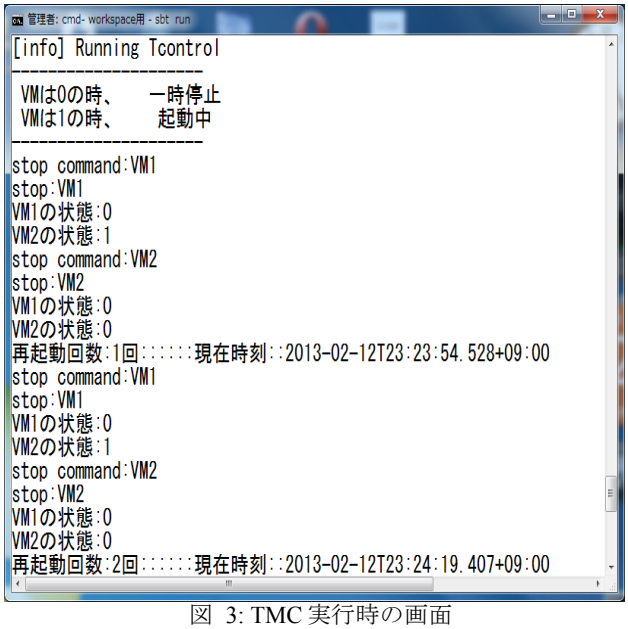


図 3: TMC 実行時の画面

## 4. 時刻管理オブジェクト

### 4.1 構成

ホスト OS 上では、両仮想マシンを制御する TMC(Time Management Controller)、仮想 OS 上では、仮想 OS の時刻通知をする TI(Time Informer)からなる。

#### 4.1.1TMC(Time manegement Controller)

仮想マシンから一定時間経過したことを通知するメッセージを受け取る。受け取った後、VIXAPI の suspend コマンドを使用し送信元の仮想マシンを一時停止させる。両仮想マシンを一時停止させた後、start コマンドで再開させる。

#### 4.1.2 TI(Time Informer)

仮想 OS の時刻が 2 秒経過するごとに TMC に時刻到達通知を送信する。

### 4.2 開発中のソフトウェア(RS)

送信時の仮想マシンの時刻をメッセージの内容とし、scala の Actor 間メッセージ通信を使用しリアルオブジェクトに送信する。リアルオブジェクトからの返信を受け取ると、またリアルオブジェクトに送信する。

### 4.3 リアルオブジェクト(RO)

RS からメッセージを受け取ると、RS に送信時の時刻を送信する。

## 5. 時刻管理の流れ

1)～3)を繰り返すことで時刻を同期させる。

**1)**RS-RO 間の通信を開始する。

**2)**TI は、一定時間経過後、TMC にメッセージを送信する。

**3)**TMC は TI からメッセージを受け取ると送信元の仮想マシンを一時停止させる。2 つの仮想マシンを一時停止させた後、両方に再開命令を出す。

## 6. 動作実験

### 6.1 動作環境

一台の計算機上(windows7,32bit)で、２つの仮想マシン(VMware Workstation7.1.6, 仮想 OS:ecolinux11.04)を起動させ、片方を開発中のソフトウェアとシミュレータとし、もう一方をリアルオブジェクトとする。また、今回の実験では RS-RO 間の通信は、シミュレータ核を通さない。

### 6.2 実験結果

図 3 は TMC 実行時の画面である。今回は一時停止、再開を 30 回行い、TI から時刻到達メッセージを受け取り、一時停止、再開命令を繰り返し実行できていることを確認できた。RS-RO 間の通信では、送信時の時刻から受信時を受信時の時刻を引いた値が正の値となり、正常に通信できていることが確認できた。

## 7. おわりに

今回の実験により仮想 OS 間上の RS-RO 間通信においての時刻を管理することができた。しかし、設定した 2000[ms]間隔ではなく、5000[ms]間隔に時刻到達通知が送信されていた。原因として仮想マシンの一時停止、再開の動作に時間が掛かり長くなっていると考えられる。今後の課題として、それぞれの動作時間を考慮した設計を考える必要がある。

### 参考文献

[1]竹田勇人,”リアルソフトウェアのネットワークシミュレータへの組み込み方法”,情報処理学会東北支部平成 22 年度第 6 回研究会,資料番号 10-6-A3-2, 2011.8.24

[2]VMWARE INC,Workstation ユーザーマニュアル VMware Workstation7.1,http://www.vmware.com/files/jp/pdf/WS71_userguid.pdf