

# シミュレーションの高速化と精度向上技術の調査

シミュレーションの並列処理とデータ同化

和田 篤

October 27, 2016

# 動機と狙い

シミュレー  
ションの高速  
化と精度向上  
技術の調査

和田 篤

離散イベント  
シミュレー  
ション

並列処理

悲観的な方法

楽観的な方法

データ同化

三次元データ同化

四次元データ同化

## テーマ選択理由

- シミュレーションの精度を上げたい, 計算時間を短くしたいという声をよく聞く
- 得意分野の機械学習, アルゴリズムでシミュレーション技術に貢献したい

## 狙い

- 並列化とデータ同化技術のアイデアと基本を紹介し, 技術開発方針検討の一助に

# 目次

シミュレー  
ションの高速  
化と精度向上  
技術の調査

和田 篤

離散イベント  
シミュレー  
ション

並列処理

悲観的な方法  
楽観的な方法

データ同化

三次元データ同化  
四次元データ同化

## 1 離散イベントシミュレーション

## 2 並列処理

- 悲観的な方法
- 楽観的な方法

## 3 データ同化

- 三次元データ同化
- 四次元データ同化

# シミュレーションの分類

## 状態変化の数による分類

- 連続シミュレーション: 状態変化が無限回
- 離散イベントシミュレーション: 状態変化が有限回

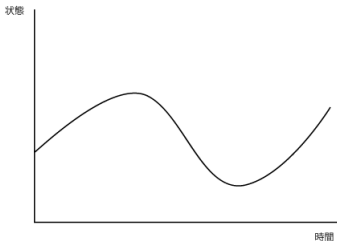


Figure: 連続シミュレーション

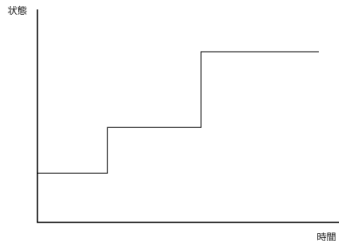


Figure: 離散シミュレーション

シミュレーションの高速化と精度向上技術の調査

和田 篤

離散イベントシミュレーション

並列処理

悲観的な方法

楽観的な方法

データ同化

三次元データ同化

四次元データ同化

# 離散イベントシミュレーション (DES)

シミュレーションの高速化と精度向上技術の調査

和田 篤

離散イベントシミュレーション

並列処理

悲観的な方法

楽観的な方法

データ同化

三次元データ同化

四次元データ同化

## DES の主要構成要素

**時刻** シミュレーション上の時刻

**状態** シミュレーションで計算したい値

**イベント** 状態を変化させる & 新たなイベントを生成

- ①処理開始@5:00
- ②処理終了@6:00
- ③処理開始@6:10
- ④処理終了@7:00



**Figure:** DES の例

**Table:** DES の例 (詳細)

時刻	イベント	状態
4:00	DES 開始	{ }
5:00		{ 装置 A: 処理中 }
6:00		{ 装置 A 工数:1 }
6:10		{ 装置 A 工数:1, 装置 B: 処理中 }
7:00		{ 装置 A 工数:1, 装置 B 工数: 1 }
8:00	DES 終了	{ 装置 A 工数:1, 装置 B 工数: 1 }

# DES に対するアルゴリズム: time-bucket 法

## time-bucket 法 (単位時間ごとに計算)

単位時間を定義し，単位時間ごとに処理すべきイベントがあるか確認する．

- 1: **while** 終了条件を満たさない **do**
- 2:     時刻を単位時間すすめる
- 3:     **if** 現在時刻に処理すべきイベントがある **then**
- 4:         イベントを処理する．

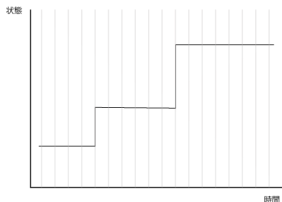


Figure: time-bucket 法のイメージ

# DES に対するアルゴリズム: event-driven 法

## event-driven 法 (状態変化時のみ計算)

イベントキューを利用して, イベントが発生する時刻のみ計算する.

- 1: **while** 終了条件を満たさない **do**
- 2:   イベントキューの先頭にあるイベントを処理
- 3:   時刻を取り出したイベントの発生時刻にする
- 4:   イベントを処理する

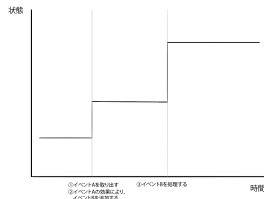


Figure: event-driven 法のイメージ

# Parallel DES の基本アイデア

シミュレーションの高速化と精度向上技術の調査

和田 篤

分散イベントシミュレーション

並列処理

悲観的な方法

楽観的な方法

データ同化

三次元データ同化

四次元データ同化

## アイデア

- 生産シミュレーションにおいて、別々の装置は並列に処理できるのではないかな？
- 待ち行列シミュレーションにおいて、別々のサービスは並列に処理できるのではないかな？



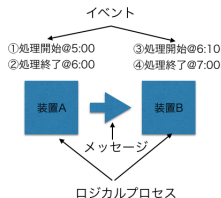
# 基本概念

## 基本概念

**LogicalProcess** イベントを処理する仮想エンティティ．それぞれの LogicalProcess が状態を持つ．

**イベント** 1つの LogicalProcess で発生し，その LogicalProcess の状態のみ変化させる．

**メッセージ** イベントが発生した際に，他の LogicalProcess に送信するイベントのこと．



シミュレーションの高速化と精度向上技術の調査

和田 篤

離散イベントシミュレーション

並列処理

悲観的な方法

楽観的な方法

データ同化

三次元データ同化

四次元データ同化

# 並列処理可能である十分条件

シミュレー  
ションの高速  
化と精度向上  
技術の調査

和田 篤

離散イベント  
シミュレー  
ション

並列処理

悲観的な方法

楽観的な方法

データ同化

三次元データ同化

四次元データ同化

## 並列処理可能である十分条件

Q: どんな時に並列に処理しても結果が変わらないか?

A: イベントを処理する順序が変わらない場合．ある

LogicalProcess は今後，送られてきうるメッセージの開始可能  
時間より小さい時刻では処理してもよい．

シミュレー  
ションの高速  
化と精度向上  
技術の調査

和田 篤

離散イベント  
シミュレー  
ション

並列処理

悲観的な方法

楽観的な方法

データ同化

三次元データ同化

四次元データ同化

## ネットワークベースの方法

# null-message

シミュレー  
ションの高速  
化と精度向上  
技術の調査

和田 篤

分散イベント  
シミュレー  
ション

並列処理

悲観的な方法  
楽観的な方法

データ同化

三次元データ同化  
四次元データ同化

## 課題

Queue が空の場合処理できない安全なメッセージのみ処理すればいいのか

## 解決方法

null-message: 状態を変化させないメッセージ arc の時間をすめるために、時刻のみのメッセージを追加する

# time-warp 法の動機

シミュレー  
ションの高速  
化と精度向上  
技術の調査

和田 篤

離散イベント  
シミュレー  
ション

並列処理

悲観的な方法  
楽観的な方法

データ同化

三次元データ同化  
四次元データ同化

## time-link 法の課題

time-link 法では，safe である可能性が高いが確実にない場合に，並列処理できない．また，null-message を多用した場合オーバーヘッドが大きい．

## time-warp 法のアイデア

time-warp 法では上記の状況でまず並列に処理し，safe でなかった場合 rollback することで並列性を高める．

# rollback

シミュレーションの高速化と精度向上技術の調査

和田 篤

分散イベント  
シミュレーション

並列処理  
悲観的な方法  
楽観的な方法

データ同化  
三次元データ同化  
四次元データ同化

ある LP で起きたイベントが safe でなかった場合，何をロールバックすればよいだろうか？

## イベントの作用

LP の状態を変化させる    他の LP にメッセージを送信する

## ロールバックの種類

そこで，以下 2 種類のロールバックが必要となる．  
restoration: LP の状態    antimessage: 他の LP に

シミュレー  
ションの高速  
化と精度向上  
技術の調査

和田 篤

離散イベント  
シミュレー  
ション

並列処理

悲観的な方法

楽観的な方法

データ同化

三次元データ同化

四次元データ同化

a

シミュレー  
ションの高速  
化と精度向上  
技術の調査

和田 篤

離散イベント  
シミュレー  
ション

並列処理

悲観的な方法

楽観的な方法

データ同化

三次元データ同化

四次元データ同化

a



# 動機

シミュレー  
ションの高速  
化と精度向上  
技術の調査

和田 篤

離散イベント  
シミュレー  
ション

並列処理

悲観的な方法

楽観的な方法

データ同化

三次元データ同化

四次元データ同化

a



Fujimoto, Richard M. "Parallel discrete event simulation." Communications of the ACM 33.10 (1990): 30-53.



Fujimoto, Richard M. Parallel and distributed simulation systems. Vol. 300. New York: Wiley, 2000.



Jefferson, David, and Henry A. Sowizral. "Fast concurrent simulation using the time warp mechanism." (1982).



Chandy, K. Mani, and Jayadev Misra. "Distributed simulation: A case study in design and verification of distributed programs." IEEE Transactions on software engineering 5 (1979): 440-452.