
スケジューリング・シミュレータ schesim ユーザーズマニュアル

名古屋大学 大学院情報科学研究科

松原豊 (yutaka@ertl.jp)

佐野泰正 (yasu@ertl.jp)

URL: <http://www.schesim.org>

最終更新日：2013年11月29日

はじめに

免責

- 本ソフトウェアは，無保証で提供されているものです．
- 著作権者は，本ソフトウェアに関して，特定の使用目的に対する適合性も含めて，いかなる保証も行いません．
- また，本ソフトウェアの利用により直接的または間接的に生じたいかなる損害に関しても，その責任を負いません．

お問い合わせ

- 本シミュレータをより良いものにするためのご意見，ご要望等を歓迎します．
- 本シミュレータに関する質問やバグレポート，ご意見，ご要望等は，開発者用メールアドレス（schesim@ertl.jp）にお送りください．

目次

1. シミュレータの概要
2. シミュレータの入出力ファイル
3. シミュレータの動作設定
4. 統計情報取得ツール
5. その他
 - シミュレーションの自動実行
 - シミュレータの内部構造

シミュレータの概要

スケジューリングシミュレータschesmの開発目的

複雑なアプリケーションのスケジューリング可能性確認

- リアルタイムスケジューリング理論では扱いにくいアプリケーションのスケジューリング可能性（デッドラインをミスすることなく実行できるか）を確認すること。
 - 例：起動周期や実行時間が変動するタスク
 - 例：タスク間の依存関係があるアプリケーション

多様なスケジューリングアルゴリズムの適用

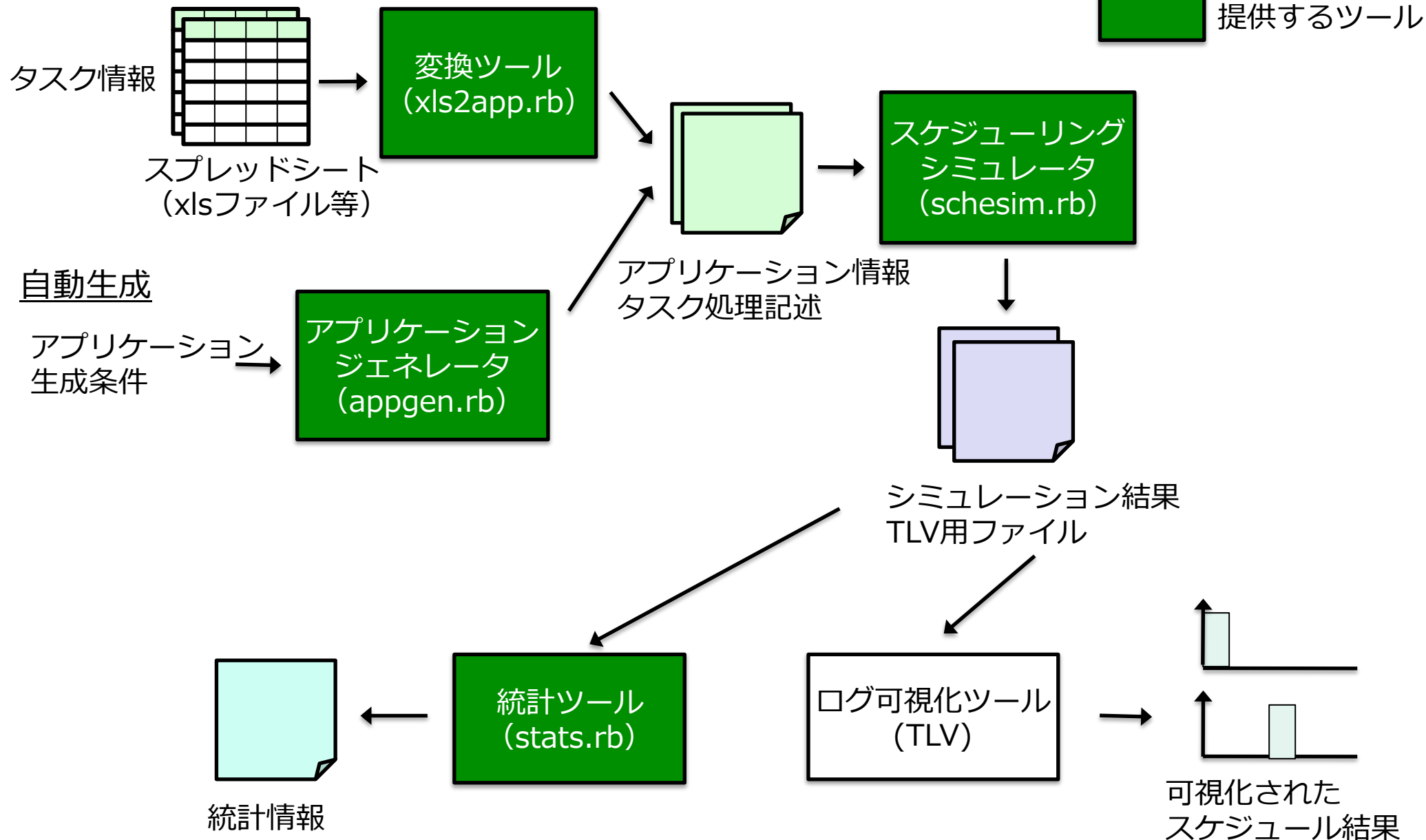
- 既存のアプリケーションに対して、多様なスケジューリングアルゴリズムを適用した場合の動作の違いを、OSのスケジューラを修正することなく、容易にスケジューリング可能性を確認すること。

新規スケジューリングアルゴリズムの開発支援

- 新しいスケジューリングアルゴリズムの動作を確認する場合に、OSのスケジューラを修正する前の段階として、アルゴリズムの動作を容易に確認すること。

シミュレーションの流れ

手動生成



パッケージの構成

説明ドキュメント

- README : 最初に読んでほしい文書

シミュレータスクリプトファイル

- schesim.rb : スケジューリングシミュレータ
- include/以下のファイル : schesim.rbから参照するファイル

ユーティリティツール (utils/)

- appgen.rb : アプリケーションジェネレータ
- xls2app.rb : xlsからアプリケーション情報ファイルへの変換ツール
- sample.xls : xls2app.rb用サンプルファイル
- test.sh : シミュレーション自動実行スクリプト
- appinfo.rb : アプリケーション情報の統計ツール
- stats.rb : 出力ログから統計情報をxlsに出力するツール

サンプルファイル (sample/)

- event, resource, scenario, offset, actque, nonpreemptive, measure

TLV用ファイル (tlv/) : TLVでログを表示するためのファイル

- ログコンバートルール, 可視化ルール, リソース定義

ユーザ向けドキュメント (doc/)

- manual.pdf : ユーザ向けマニュアル (本ファイル)

シミュレータの動作環境

Rubyが動作するコンピュータ

- OS : Windows (cygwin, mingw) , MacOS X, Linux
- Ruby : バージョン1.9.3以降 (1.8.x, 1.9.2を使用する場合には, schesim 0.7.3を使用して欲しい)

必要なライブラリ

- スクリプトを実行した際に, 次のエラーが表示される場合には, ライブラリが足りない可能性がある
 - Jsonライブラリがない場合のエラーメッセージ
 - ``require': no such file to load -- json (LoadError)`
 - インストール方法
 - `# gem install json`
 - gemコマンド自体がインストールされていない場合には, <http://rubyforge.org/projects/rubygems/>からダウンロードしてインストールする.

ログ可視化ツールTLV

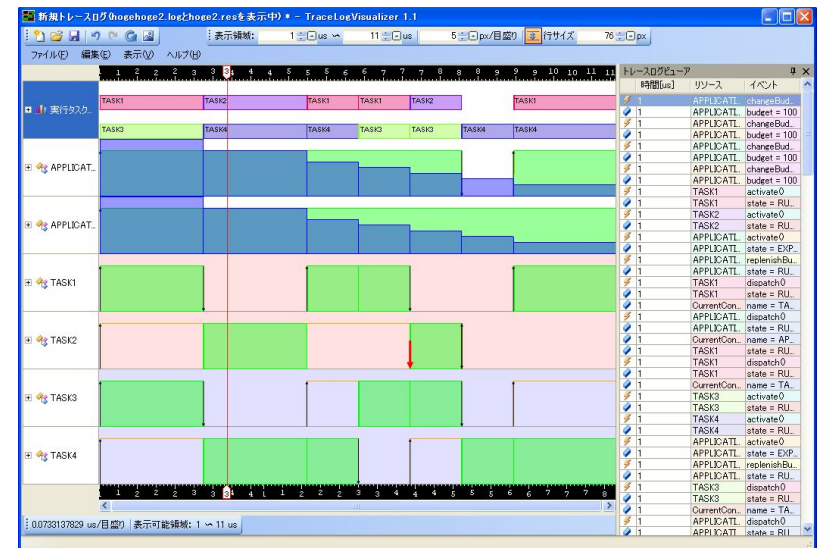
シミュレータの出力ログを可視化するために、TOPPERSプロジェクトから公開されているTLV (Trace Log Visualizer) を用いることを想定している。

動作環境

- Windows XP/Vista/7
- .NET Framework 3.5以降

ダウンロード方法

- TOPPERSプロジェクトのウェブサイト (<http://toppers.jp/tlv.html>) からダウンロード可能



とりあえずシミュレータを動かしてみる！

シングルスプロセッサで固定優先度ベーススケジューリングをシミュレーションする

(1) アプリケーションの生成

- 生成するアプリケーションを格納するディレクトリの作成
 - `mkdir obj`
- アプリケーションジェネレータの実行
 - `./utils/appgen.rb -t 1 -a 1 -c 1 -C -p rm -g fp -l fp -d obj/`

(2) シミュレーションの実行

- `./schesim.rb -t ./obj/000001.json -r ./obj/000001.res -d obj/000001.rb -e 500 > obj/000001.log`

(3) TLVでの表示

- objディレクトリ以下に生成された000001.logと000001.resをTLVに入力してログを可視化（詳細はTLVのドキュメントを参照のこと）

(4) アプリケーション情報の表示

- 000001.jsonの情報（タスク数, CPU利用率）を表示
- `./appinfo.rb -t obj/000001.json`

シミュレータの入出力ファイル

アプリケーションの自動生成： アプリケーションジェネレータ

機能

- 生成条件を満たすアプリケーションのアプリケーション情報ファイル（タスクのパラメータ, 適用するスケジューリングアルゴリズム等を含む, シミュレータへの入力ファイル）を生成
 - 複数のアプリケーションファイルを生成する場合に有効

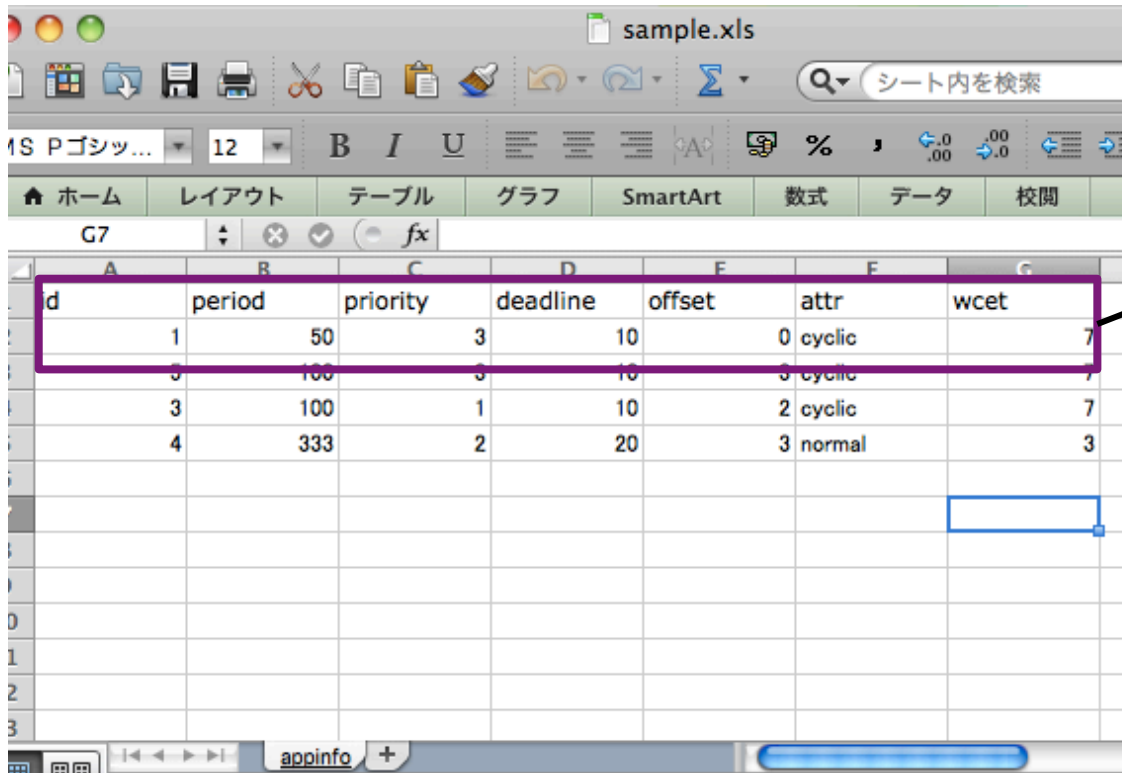
アプリケーション生成条件（指定できるパラメータ）

- 自動生成するアプリケーションの数
- 1プロセッサ内のコア数
- 1コア内のアプリケーション数
- タスクの起動属性（周期, 非周期, 初期起動しない）
- スケジューリングアルゴリズム
 - グローバルスケジューリングアルゴリズム
 - ローカルスケジューリングアルゴリズム

ツールの使用方法は, `appgen.rb`を参照のこと

アプリケーションの手動生成： スプレッドシートでの入力

- スプレッドシート（Excel等）にタスクのパラメータを入力し、変換ツールを使って、アプリケーション情報ファイルを生成
 - ツールの具体的な使用方法は、utils/xls2app.rbを参照のこと



d	period	priority	deadline	offset	attr	wcet
1	50	3	10	0	cyclic	7
2	100	3	10	3	cyclic	7
3	100	1	10	2	cyclic	7
4	333	2	20	3	normal	3

タスクのパラメータを入力
(詳細は次頁)

※現在は、シングプロセッサでの固定優先度スケジューリングを想定して
アプリケーション情報ファイルを生成
(変更したい場合は、アプリケーション情報ファイルを直接修正する)

アプリケーション情報ファイル（コアのパラメータ）

```
{
  "cpu": [
    {
      "id": 1,
      "core": [
        {
          "scheduling": "edf",
          "application": [
            {
              "scheduling": "fp",
              "share": 1,
              ...
              "task": [
                {
                  "period": 36,
                  "priority": 1,
                  "id": 1,
                  ...
                  "max_actcnt": 1
                }
              ],
              "pri": 1
            }
          ],
          "id": 2
        }
      ]
    }
  ]
}
```

JSON（JavaScript Object Notation）フォーマットを採用

- ・ グローバルスケジューリングアルゴリズム
以下のいずれかを指定
fp：固定優先度スケジューリング
edf：EDF
bss/tpa：階層型スケジューリング
- ・ アプリケーションの情報
次のページを参照のこと
- ・ コアID
1以上の整数値

アプリケーション情報ファイル（アプリのパラメータ）

```
{
  "cpu": [
    {
      "id": 1,
      "core": [
        {
          "scheduling": "edf",
          "application": [
            {
              "scheduling": "fp",
              "share": 1,
              "period": 1,
              "id": 1,
              "task": [
                {
                  "period": 36,
                  "priority": 1,
                  "id": 1,
                  "deadline": 36,
                  "wcet": 34,
                  "attr": "cyclic",
                  "offset": 5,
                  "max_actcnt": 1
                }
              ],
              "pri": 1
            }
          ]
        }
      ]
    }
  ]
}
```

- ・ローカルスケジューリングアルゴリズム
以下のいずれかを指定
fp：固定優先度スケジューリング
edf：EDF
- ・シェア
0より大きく、1以下の浮動小数点数
※1コア内のアプリケーションのシェアの合計は1以下であること
- ・起動周期
0より大きな浮動小数点数
- ・アプリケーションID
1以上の整数値
- ・タスクの情報
- ・優先度
1以上の整数値
※数値が小さいほど優先度が高い

アプリケーション情報ファイル（タスクのパラメータ）

```
{
  "cpu": [
    {
      "id": 1,
      "core": [
        {
          "scheduling": "edf",
          "application": [
            {
              "scheduling": "fp",
              "share": 1,
              "period": 1,
              "id": 1,
              "task": [
                {
                  "period": 36,
                  "priority": 1,
                  "id": 1,
                  "deadline": 36,
                  "wcet": 34,
                  "attr": "cyclic",
                  "schedule": "full"
                },
                {
                  "offset": 5,
                  "max_actcnt": 1
                }
              ],
              "pri": 1
            }
          ]
        }
      ]
    }
  ]
}
```

- 周期（最小到着間隔）
0より大きな浮動小数点数
- 優先度
1以上の整数値，数値が小さいほど優先度が高い
- タスクID
1以上の整数値
- 相対デッドライン
0より大きな浮動小数点数
- 最悪実行時間
0より大きな浮動小数点数
※シミュレーションでは使用しないため必須ではない
統計情報ツール（appinfo.rb）でのみ使用する
- 起動属性 次のいずれかを指定
cyclic: 指定した周期で起動する
sporadic: 指定した周期を最小到着間隔として起動する
normal: 起動要求が発生するまで起動しない
- スケジューリング属性 次のいずれかを指定
full: プリエンプティブなタスク（デフォルト）
non: ノンプリエンプティブなタスク
- 初期起動時刻のオフセット
0以上の浮動小数点数: 指定した時刻にタスクを起動する
null: タスクは自動的に起動しない（デフォルト）
- 起動要求キューイング数の最大値
0以上の整数値
null: キューイング数を制限しない（デフォルト）

タスク処理記述ファイル

タスクの処理内容を, rubyを用いて記述したもの

```
class TASK
  @@sem1 = SEMAPHORE.new(1) # セマフォの生成
  @@mode = 1                # 動作モード
  def task10                  # タスクIDが10のタスク処理記述
    pretask_hook
    case @@mode
    when 1
      # モード1のときの動作
      wai_sem(@@sem1)
      exc(1)
      sig_sem(@@sem1)
    when 2
      # モード2のときの動作
      exc(2)
    end
    posttask_hook
  end

  def task20
    pretask_hook
    exc(1)
    wai_sem(@@sem1)
    exc(2)
    sig_sem(@@sem1)
    posttask_hook
  end
end
end
```

指定可能なスケジューリングアルゴリズム

コアごとに、アプリケーションの実行順序を決定する
グローバルスケジューリングアルゴリズムを指定

- EDFS (Earliest Deadline First Scheduling)
 - FPS (Fixed Priority Scheduling)
 - BSS (Bandwidth Sharing Server) Algorithm
 - TPA (Temporal Protection Algorithm)
 - ※コアを跨いでスケジューリングするSMP型アルゴリズムには未対応
- 階層型
スケジューリング

アプリケーションごとに、タスクの実行順序を決定する
ローカルスケジューリングアルゴリズムを指定

- FPS
 - 優先度の割り当て方法をRM (Rate Monotonic) ,
DM (Deadline Monotonic) , ランダムから選択可能
- EDFS

シミュレーション結果（ログ情報）

システム時刻

TOPPERSカーネルのログフォーマットを採用

コアID

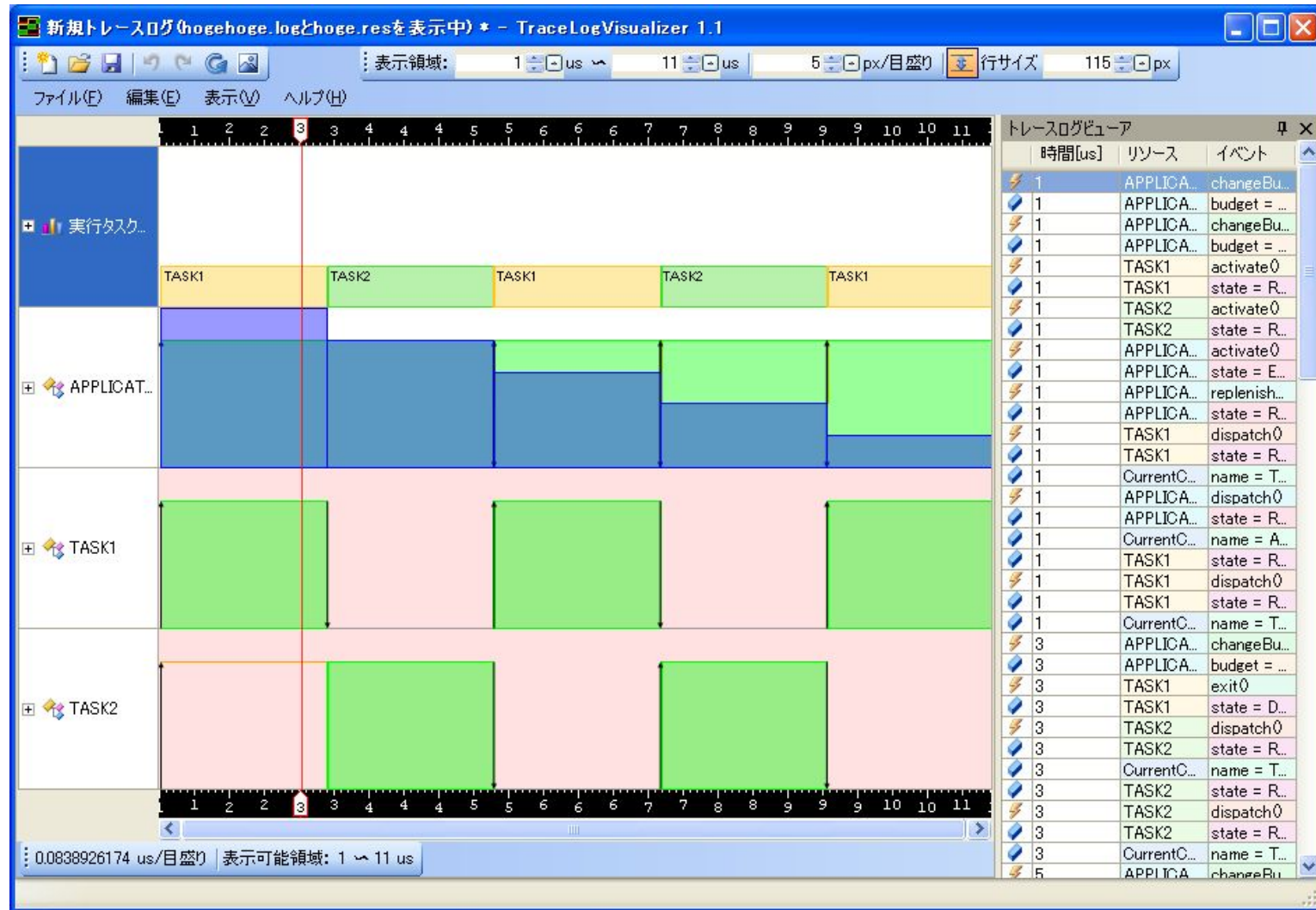
```
[1]:[1]: task 1 becomes RUNNABLE.  
[1]:[1]: task 2 becomes RUNNABLE.  
[1]:[1]: task 3 becomes RUNNABLE.  
[1]:[1]: application 1 becomes EXPIRED.  
[1]:[1]: application 1 becomes RUNNABLE.  
[1]:[1]: dispatch to task 1.  
[1]:[1]: task 4 becomes RUNNABLE.  
[1]:[1]: application 2 becomes EXPIRED.  
...  
[2]:[1]: budget of application 1 is 0.  
[2]:[1]: application 1 becomes EXPIRED.  
[2]:[2]: budget of application 3 is 0.  
[2]:[2]: application 3 becomes EXPIRED.  
[2]:[1]: budget of application 1 is 10.
```

タスクの状態遷移ログ
(TOPPERSカーネルと同じ)

アプリケーション関連のログ
(独自に追加)

TLVによる可視化（シングルコア）

1コア1アプリ2タスク



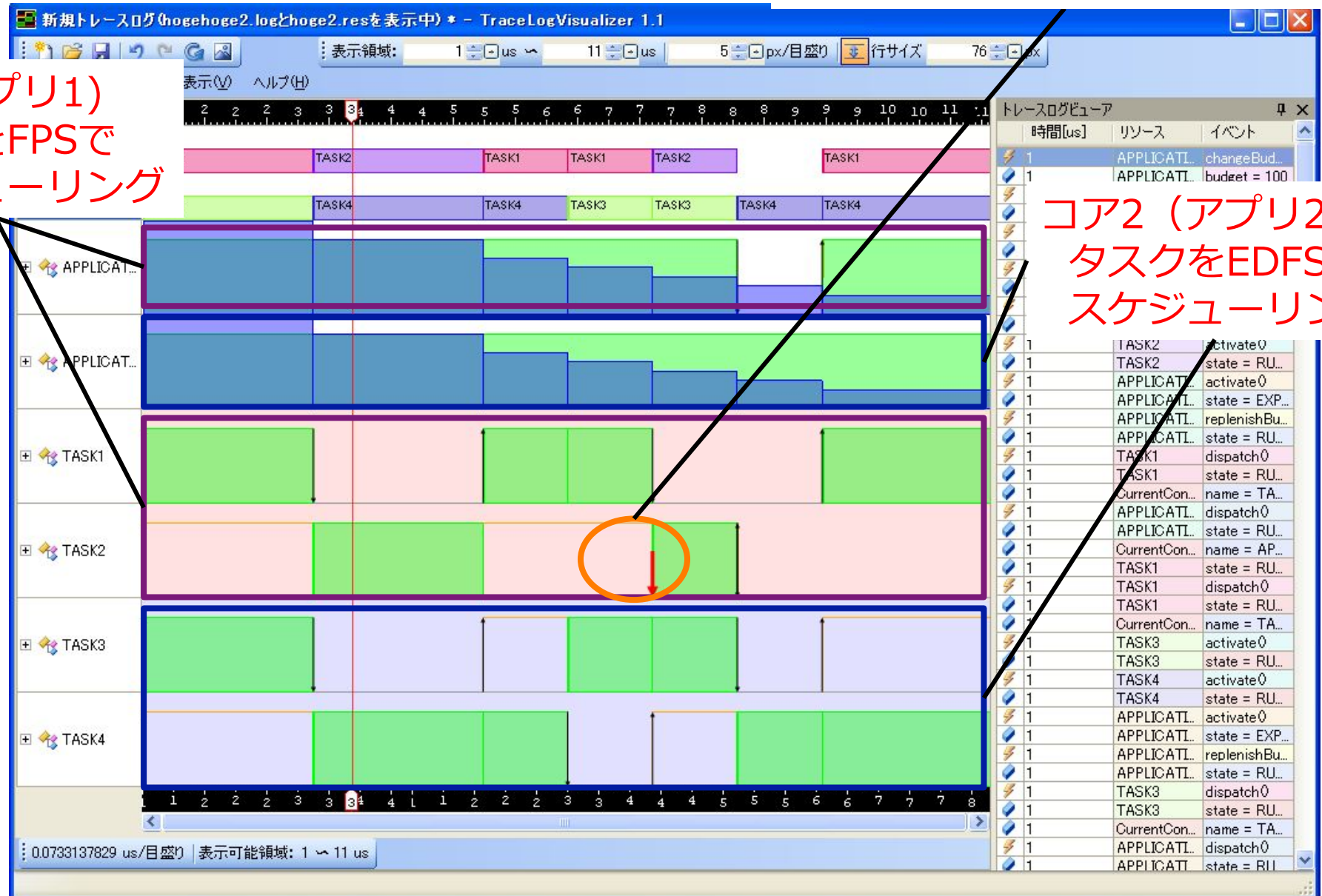
TLVによる可視化 (2コア)

2コア各1アプリ各2タスク

デッドライン・ミス発生時の表示

コア1(アプリ1)
タスクをFPSで
スケジューリング

コア2 (アプリ2)
タスクをEDFSで
スケジューリング



シミュレータの動作設定

シミュレータの動作設定を変更する

- シミュレータ設定ファイル
(schesim.conf) を変更することで、シミュレータの動作を変更できる.
- 現時点で設定可能な項目は、以下の通りである.
 - シミュレーション停止条件
 - 発生したデッドラインミス回数
 - 出力されるログ
 - システム時刻の小数点以下の桁数
 - タスクの動作に関連するログ
 - アプリケーションの動作に関連するログ
 - API呼出しのログ

動作設定ファイル (schesim.conf)

```
{
  "stop_condition": {
    "deadline_miss": 2
  },
  "log": {
    "system_time": 0
    "task": {
      "stat": true,
      "dispatch_from": true,
      "dispatch_to": true,
      ...
    },
    "api": {
      "act_tsk": true,
      "wai_sem": true,
      "sig_sem": true,
      ...
    },
    "application": {
      "stat": true,
      "dispatch_from": true,
      "dispatch_to": true,
      ...
    }
  }
}
```

一部抜粋

← シミュレーションを停止するデッドラインミス回数
例：2回デッドラインミスが発生した時点で停止する

← システム時刻表示における小数点以下の桁数
例：2を設定すると、1.234を123（2桁に丸めて100倍）と表示する

← タスクの状態変化に関するログの設定
false にすると次のようなログが出力されなくなる
例：[10]:[1]: task 1 becomes RUNNABLE.

← act_tsk呼出し時に出力されるログの設定
false にすると次のようなログが出力されなくなる
例：[10]:[2]: applog strtask : TASK 4 : act_tsk(5).

← アプリケーション切換え時に出力されるログの設定
false にすると次のようなログが出力されなくなる
例：[50]:[2]: dispatch from application 2.

統計情報取得ツール

統計情報取得ツールstat

統計情報取得ツール（utils/stats.rb）を使うと、シミュレーション結果（*.log）から統計情報を簡単に計算できる

- 統計情報は、xlsファイルにまとめて生成

タスクごとの統計情報

- 起動回数
- 平均, 最大, 最小応答時間
- 平均, 最大, 最小機能応答時間
- 合計, 平均, 最大, 最小実行時間
- CPU利用率

システム全体の統計情報

- タスク数
- CPU利用率

ユーザが指定した計測区間の情報

- 計測回数
- 平均, 最大, 最小応答時間
- 平均, 最大, 最小実行時間

統計情報取得ツールの動作に必要なログの種類

- 統計情報取得ツールを動作させるためには、少なくとも以下のログが必要である
 - taskに関するログ
 - stat
 - dispatch to
 - dispatch from
 - actque
 - APIに関するログ
 - act_tsk
 - SetEvent
 - WaitEvent
 - ClearEvent

※./schesim.conf 内で対応するログ出力設定を“true”にすること

統計情報の定義

起動回数

- 対象タスクが休止状態から実行可能状態に遷移した回数
- 待ち状態から実行可能状態に遷移した回数も、起動回数に加える場合には、-Fもしくは、-Sのいずれかのオプションを付けて実行する

応答時間

- 対象タスクに対する起動要求から対象タスクが終了するまでの時間

機能応答時間

- 対象タスクの起動元タスクの起動時刻から、対象タスクが終了するまでの時間

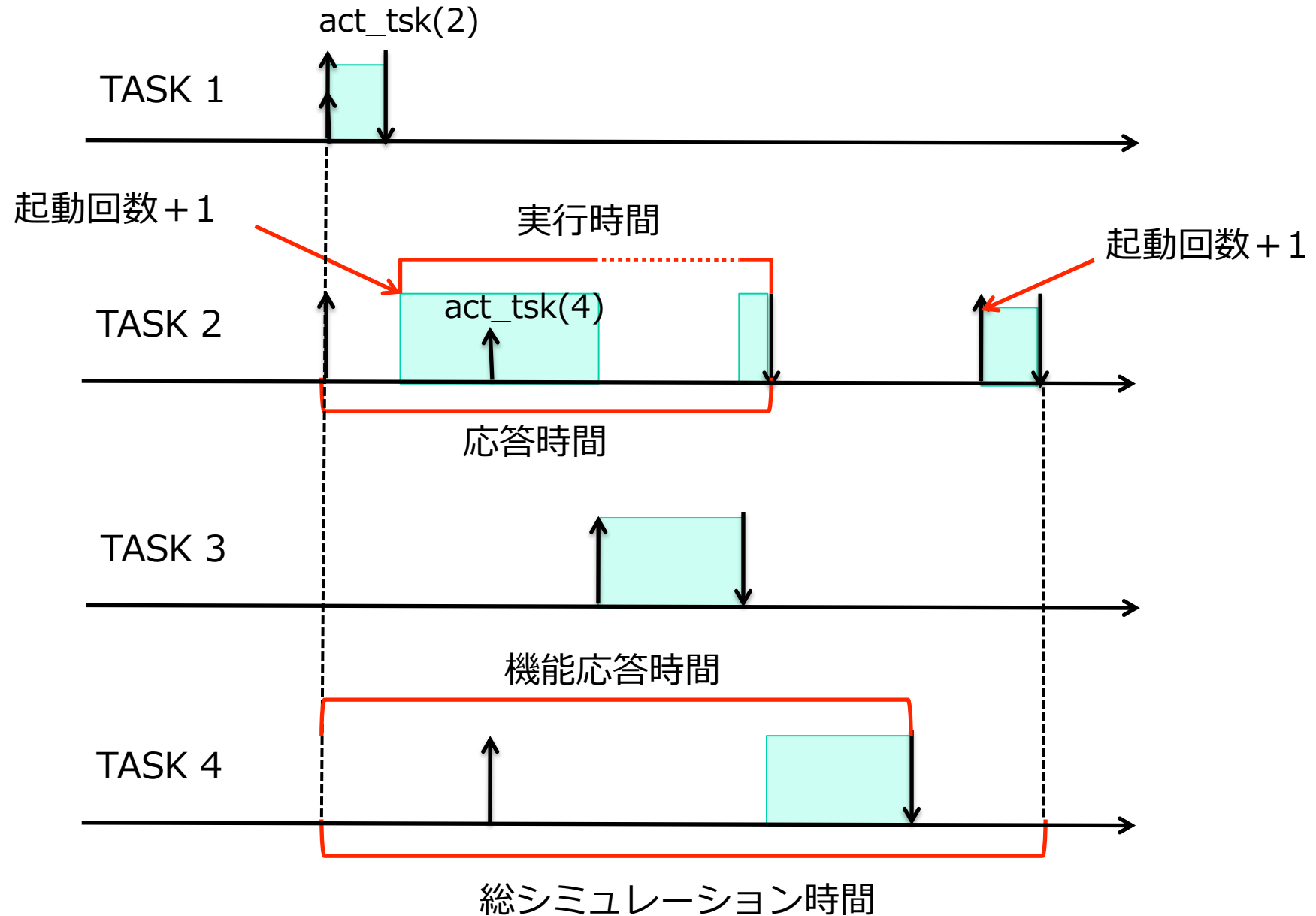
実行時間

- 対象タスクの実行状態の時間（プリエンプトされている時間は除く）

CPU利用率

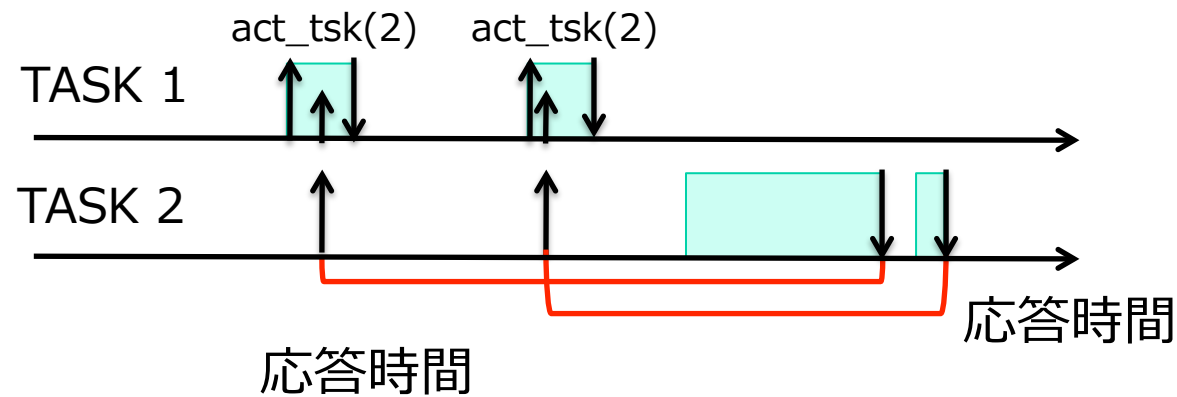
- 対象タスクの合計実行時間 / 総シミュレーション時間
- 総シミュレーション時間 = (最後のログの時刻) - (最初のログの時刻)

統計情報の定義の図示

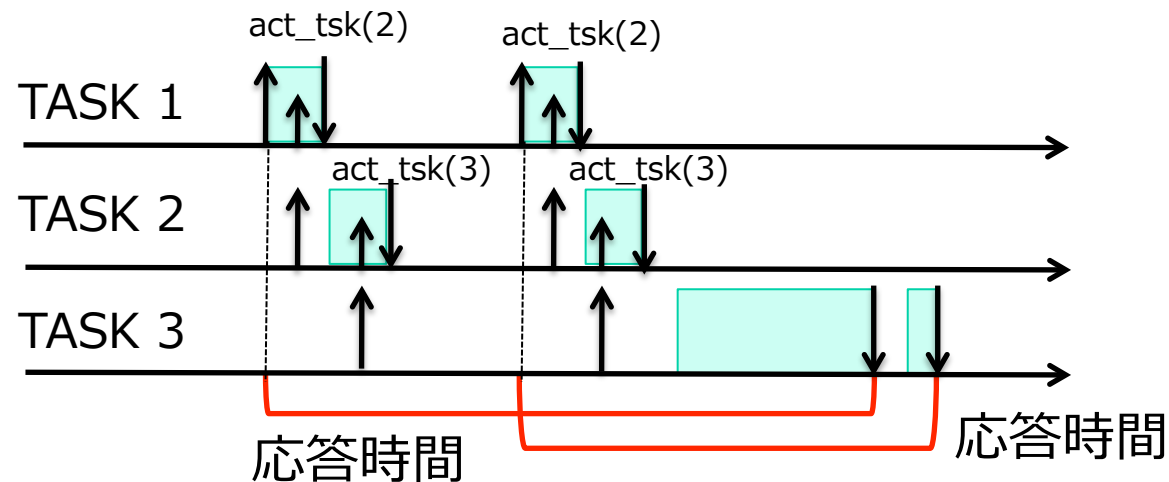


起動要求キューイング時の定義の図示

応答時間



機能応答時間



区間 (begin_measure, end_measure) を指定した計測

計測内容

- 計測回数：対象IDの計測区間の数
- 実行時間：対象区間内における, begin_measureを発行したタスクの実行時間
- 応答時間：対象区間の時間

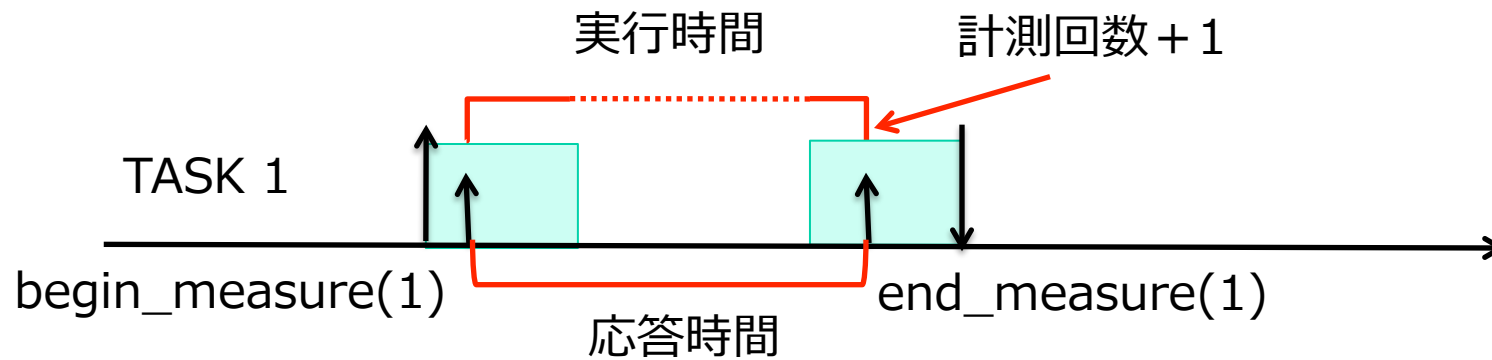
前提条件

- 計測に用いるID (begin_measure, end_measureの引数) は0以上の整数値を用いる

制限事項 (エラーになること)

- 例えば, begin_measure(1)を発行後, end_measure(1)を発行する前に, 再度begin_measure(1)を発行した場合
- →異なるタスク間で同一のIDを使用しないこと
- 例えば, TASK1でbegin_measure(1)を発行後, TASK2でend_measure(1)を発行して, タスクをまたいだ区間を計測したい場合
- →実行時間の計測区間が明確でなくなるため, 許可しない

区間指定時の統計情報の定義の図示



統計情報取得ツールの使用方法

- シミュレーション結果(*.log)が./obj/result.logにある場合に、./statistics.xlsというxlsファイルを生成する
 - ./utils/stats.rb -i ./obj/result.log -o ./statistics.xls
- タスクが終了することなく、イベントフラグを用いて、待ち状態、待ち解除状態を繰り返すタスクの統計情報も計測する場合
 - 最初の起動から最初のWaitEvent()までを計上する場合
 - ./utils/stats.rb -i ./obj/result.log -o ./statistics.xls -F
 - 最初の起動から最初のWaitEvent()までを無視する場合
 - ./utils/stats.rb -i ./obj/result.log -o ./statistics.xls -S
 - ※このオプションを付けると上記以外の目的でイベントフラグを使っているタスクセットでは、正しい計測結果が得られない場合があるので注意

機能応答時間を計測する場合

- ./utils/stats.rb -i ./obj/result.log -o ./statistics.xls -R

出力例

タスク ID	起動回数	平均応答時間	最大応答時間	最小応答時間
TASK1	10	15	20	12
TASK2	23	14	23	10
TASK3	5	20	25	3

その他

シミュレータの実行（自動実行）

大量のアプリケーションを対象にシミュレーションを実行する場合には、シミュレーションの自動実行ツールを使用できる。

- 例) シングルプロセッサで、10個のアプリケーションに対して、それぞれ固定優先度ベーススケジューリング、階層型スケジューリング（BSS, TPA）の3つの手法でのスケジューリングをシミュレーションする。

(1) 各種ディレクトリの作成

- mkdir fp_ok
- mkdir bss_ok
- mkdir tpa_ok

(2) アプリケーション生成 + シミュレーションの実行

- 10個のアプリケーションを生成してシミュレーションを実行する。
 - ./utils/test.sh 10 fp_ok/ bss_ok/ tpa_ok/
- fp_ok, bss_ok, tpa_okに各シミュレーション結果 (*.log) が生成される

(3) アプリケーション統計情報の表示

- fp_okディレクトリにあるアプリケーションの統計情報（平均タスク数, 平均CPU利用率）を表示
- ./utils/appinfo.rb -d ./fp_ok/

シミュレーションの流れ

