Do We Need to Handle Every Temporal Violation in Scientific Workflow Systems?

Yun Yang (& Xiao Liu) – 杨耘 (yyang@swin.edu.au)
http://www.ict.swin.edu.au/personal/yyang
SUCCESS
Swinburne University of Technology
Melbourne, Australia
SUCCESS – A Brief Introduction

Swinburne University Centre for Computing and Engineering Software Systems

- Swinburne is one of top 400 universities in the world
  - 2nd smallest one (only some 400 academic staff)
  - No 1 technological university in Australia
- SUCCESS has the strongest SE group in Australia
- SUCCESS 2011 figures on two top SE journals:
  - TSE – IEEE Trans. on Software Engineering
    4 (2+2) (world total: 48)
  - ToSEM – ACM Trans. on Software Engineering and Methodology
    2 (1+1) (world total: 18)
Melbourne – Capital City of Victoria

- a very dynamic city
- population over 4 million
- Australia’s cultural capital
- famous for parks and gardens
- “The Most Liveable City in the World”
- Welcome for (joint) PhD program etc.
Outline

- Related Publications (and Acknowledgement)
- Temporal Violation Handling Point Selection
- Evaluation
- Conclusions
Related Publications for This Talk

Acknowledgement: Assoc. Prof. Jinjun Chen; Dr Xiao Liu (two former PhD graduates)


Based on TOSEM Submission (under revision)
Outline

- Related Publications (and Acknowledgement)
- Temporal Violation Handling Point Selection
- Evaluation
- Conclusions
**Functional Exceptions vs Non-Functional Temporal Violation**

- **Question**: “Do we need to handle every temporal violation?”
  - Equivalent: “is every necessary and sufficient checkpoint a violation handling point?”
- **Answer 1**: “Yes”, we need to handle every temporal violation when it is detected in the first place
- **Answer 2**: “No”, we can further select a subset from the necessary and sufficient checkpoints
  - Cost effectiveness: the overall violation handling cost is huge
  - Self-recovery: auto recovery with future time redundancy
Motivating Example

- Astrophysics: pulsar searching

- Pulsars: the collapsed cores of stars that were once more massive than 6-10 times the mass of the Sun

- [http://astronomy.swin.edu.au/cosmos/P/Pulsar](http://astronomy.swin.edu.au/cosmos/P/Pulsar)


- Typical scientific workflow which involves a large number of data and computation intensive activities. For a single searching process, the average data volume (not including the raw stream data from the telescope) is over 4 terabytes and the average execution time is about 23 hours on Swinburne high performance supercomputing facility ([http://astronomy.swinburne.edu.au/supercomputing/](http://astronomy.swinburne.edu.au/supercomputing/)).
Problem Analysis

- Fundamental requirements:
  - Temporal conformance: the lower the violation rate, the better the temporal conformance is.
  - Cost effectiveness: the smaller the number of selected handling points, the better the cost effectiveness is.

- **Problem 1)** How to measure temporal violations in a quantitative fashion
  - Solution: Probability based temporal consistency model

- **Problem 2)** How to decide whether a checkpoint needs to be selected as a handling point or not
  - Solution: Adaptive temporal violation handling point selection strategy
Temporal QoS

- System performance
  - Response time
  - Throughput

- Temporal constraints
  - Global constraints: deadlines
  - Local constraints: milestones, individual activity durations

- Satisfactory temporal QoS
  - High performance: fast response, high throughput
  - On-time completion: low temporal violation rate
Temporal Checkpoint Selection

- Our Strategy

Necessary and Sufficient Checkpoint Selection Strategy

- Probability Time Redundancy
- Minimum Probability Time Redundancy
- DOMTR: Dynamically Obtaining Minimum Time Redundancy
- Theorem of Checkpoint Selection
- Proof of Necessity and Sufficiency
Results (1)
Results (2)

![Graph showing total checkpoints omitted by each CSS vs. workflow size (activities).](image)
Results (3)

Verification Computation Units

Number of Verification Computation Units

Number of Upper Bound Constraints
Basic Idea (1)

- **Strategy overview**

  - **Input:**
    - A necessary and sufficient checkpoint $a_p$ selected by $CSS_{TD}$;
    - The maximum probability time deficit $MPTD(a_p)$;
    - The minimum probability time redundancy $MPTR(a_p)$;
    - The probability threshold for self-recovery $PT$;
    - The result of last temporal violation handling $Success$ in $[true, false]$.

  - **Output:**
    - True or False $a_p$ as a temporal violation handling point

  - **Steps:**
    - Step 1: Adaptive modification of PT
    - Step 2: Temporal violation handling point selection
Basic Idea (2)

- Temporal Violation Handling Point Selection Rule
  - At activity $a_p$, with the probability of self-recovery $P(T)$ and the probability threshold $PT$, the rule for temporal violation handling point selection is as follows: if $P(T) > PT$, then the current checkpoint is not selected as a handling point; otherwise, the checkpoint is selected as a handling point.

- Quantitatively measure the probability of self-recovery $P(T)$
  - The model for “Probability of Self-Recovery”: a probability distribution model, the maximum probability time deficit, the minimum probability time redundancy.
Basic Idea (3)

- Adaptive Modification Process for Probability Threshold $PT$

  - Given current probability threshold $PT$ and checkpoint $a_p$, i.e. a temporal violation is detected at $a_p$, $PT$ is updated as $PT*(1+r)$. Afterwards, based on our handling point selection rule, if $a_p$ is not selected as a handling point, then $PT$ is updated as $PT*(1-r)$. Otherwise, $PT$ remains unchanged. Here, $r$ stands for the update rate.

  - The adaptive modification process is to increase the probability threshold, i.e. the probability of violation handling, where violation handling is triggered; or to decrease where violation handling is skipped if self-recovery applies.
Outline

- Related Publications (and Acknowledgement)
- Temporal Violation Handling Point Selection
- Evaluation
- Conclusions
Simulation Environment

- SwinCloud

Cloud Simulation Environment

Data Centres with Hadoop

VMware

Swinburne Computing Facilities

Astrophysics Supercomputer
- GT4
- SuSE Linux

Swinburne CS3
- GT4
- CentOS Linux

Swinburne ESR
- GT4
- CentOS Linux
Experimental Settings

**Scientific Workflows**
- Scientific workflow size: from 2000 to 50,000 workflow activities;
- Activity durations: all activity durations follow the normal distribution model. The mean duration is randomly selected from 30 to 3000 time units, and the standard deviation is set as 10% of its mean;
- Temporal constraints: the initial build time probability for deadlines are set as 90% according to [Liu et al. 2011a];
- Workflow segments: the average length of the workflow segments for subsequent activities is set as 20;
- Random noises: the duration of one selected activity in each workflow segment is increased by 5%, 15% or 25% of its mean in different rounds.

**Temporal Violation Handling**
- Temporal violation handling strategy: a pseudo strategy with 50% time compensation rate;
- Size of subsequent workflow segment: randomly selected as 3 to 5;
- Success rate: the success rate for violation handling is set as 80%.

**CSS\textsubscript{TD}**
- Default values as defined in [Chen and Yang 2011].

**RT**
- The fixed confidence threshold: $FT$ is set as 0.9, i.e. select 10% from the total checkpoints as adjustment points in a pure random fashion.

**AD**
- The update rate: $\gamma$ is initially set as 0.5 and gradually decreased to 0.05.

**NIL**
- Without temporal violation handling.

* Note that we have also applied our strategy with different distribution models, and the conclusions are consistent
Experimental Results

Experimental results (Noise=0%).

Experimental results (Noise=5%).
Experimental Results

Experimental results (Noise=15%).

Experimental results (Noise=25%).
Experimental Results

Cost reduction rate vs. violation increase rate

- Cost Reduction Rate
- Violation Increase Rate

Round1(0%)  Round2(5%)  Round3(15%)  Round4(25%)
Experimental Results

- Yearly cost reduction for the pulsar searching workflow.

- Yearly time reduction for the pulsar searching workflow.
Outline

- Related Publications (and Acknowledgement)
- Temporal Violation Handling Point Selection
- Evaluation
- Conclusions
Conclusions

- Temporal conformance vs. cost effectiveness

- Not every necessary and sufficient checkpoint needs to be selected as a violation handling point

- Saving a lot of time and cost (e.g. over 98% under normal circumstance) while maintaining satisfactory temporal conformance (close to 0 violations)
Future Work

Move from computation-intensive scientific workflows to instance-intensive business workflows

- Fast response time vs. high system throughput
- Different temporal consistency model
- Different monitoring strategies
- Different violation handling strategies
End - Q&A

- Thanks for your attention!