# Checkpoint Selection for Temporal Verification in Grid Workflows


Dr. Jinjun Chen
CITR - Centre for Information Technology Research
Swinburne University of Technology, Australia
http://www.ict.swin.edu.au/personal/jchen


June 2007

## Content

- Grid Workflow and Verification & Validation (V&V)
- Temporal Verification and Checkpoint Selection
- Problem Analysis
- A New Checkpoint Selection Strategy
- Simulation Evaluation
- Conclusions
- References

## Grid Workflow and V&V

- A grid workflow management system
  - A type of workflow management system aiming at supporting large-scale sophisticated scientific and business processes in complex e-science and e-business applications, by facilitating the resource sharing and computing power of underlying grid infrastructure.
- A scientific workflow management system
  - A type of workflow management system aiming at supporting complex scientific processes in many e-science applications such as climate modelling, astronomy data processing. It may or may not be built upon grid infrastructure. Can be cluster or P2P.
- A business workflow management system
  - A type of workflow management system aiming at supporting business processes in many business applications such as student registration or insurance claims. It may typically be built on Internet, Intranet or other internal enterprise information infrastructure.

## Grid Workflow and V&V

- Similarities (typically rather than exactly)
  - Motivation
    - Some applications often require the creation of a collaborative workflow.
    - Many e-scientists and e-business people lack the necessary low-level expertise to utilise the underlying information infrastructure (Grid, Internet protocol suites and etc.).
    - Documented workflow specifications are beneficial for modelling and managing processes; and can be reused, modified and shared.

## Grid Workflow and V&V

- Similarities (typically rather than exactly)
  - Common requirements/functions
    - Build-time, run-time
    - Control flow modelling
    - Even-based analysis
    - Collaboration support
    - Exception handling
    - Overall management
    - QoS issue
    - Security issue
    - ……

## Grid Workflow and V&V

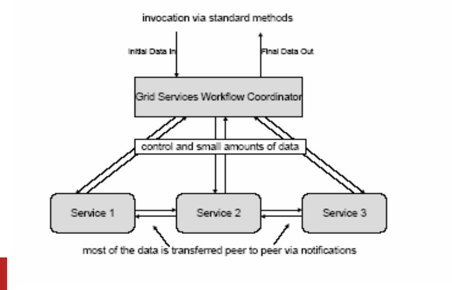
- Differences (typically rather than exactly)
  - Grid/Scientific Workflows
    - Computation or data intensive
    - Less human interaction and transient service management
    - A large number of activities
    - High dynamic execution environments
    - Dynamic resource allocation, scheduling and mapping to underlying distributed infrastructure
    - Scientific workflows may or may not be supported by grid infrastructure

## Grid Workflow and V&V

- Differences (typically rather than exactly)
  - Business Workflows
    - More Logic rather than computation or data intensive
    - Much more human interaction (an office workflow)
    - A reasonable number of activities
    - Execution environments is often much more certain.
    - Not much resource allocation, scheduling and mapping
    - Internet (Web Services), Intranet or other internal enterprise information infrastructure

## Grid Workflow and V&V

- Differences (examples from [3])
  - Dealing with large amount of data (for grid workflows)

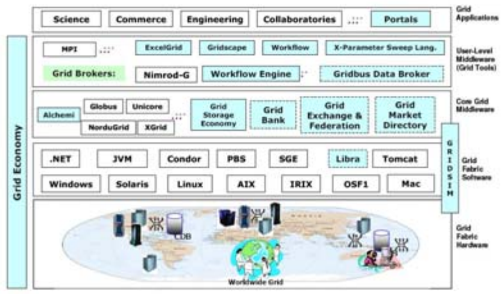



## Grid Workflow and V&V

- Differences (examples from [3])
  - Lifecycle management (for grid workflows)
    - Grid service instances are created by the Factories,
    - And destroyed explicitly or via soft-state
    - Certain grid services in the workflow will not be executing while others are

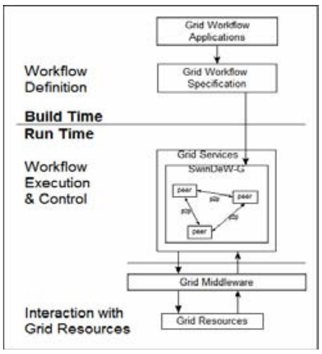
## Grid Workflow and V&V

- General architecture (from [4])




## Grid Workflow and V&V

- A sample grid workflow management system – SwinDeW-G: Swinburne Decentralised Workflow for Grid [7].

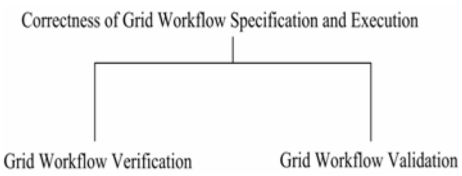



## Grid Workflow and V&V

- Grid workflow verification and validation is concerned about the correctness of grid workflow specification and execution [1].
- According to IEEE Standard Glossary of Software Engineering Terminology, the correctness is defined as freedom from faults, meeting of specified requirements, and meeting of user needs and expectations [5].
- Based on IEEE Software Verification and Validation Standard, the correctness can be more commonly described as grid workflow verification and validation [6].

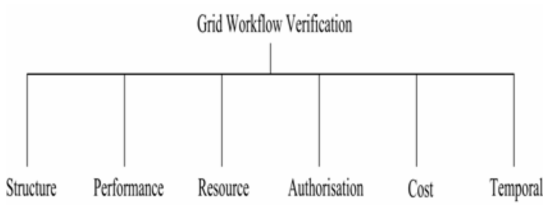
## Grid Workflow and V&V

- Verification failure results in the grid workflow specification and execution containing faults or flaws.
- Validation failure constitutes a breach of contract between the complex scientific and business process developer and the client.



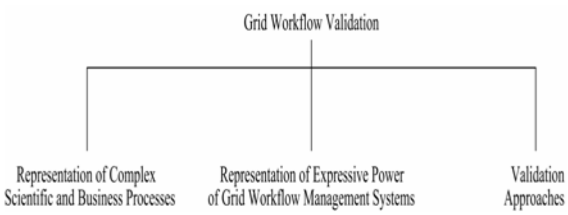

## Grid Workflow and V&V

- Grid workflow verification




## Grid Workflow and V&V

- Grid workflow validation




## Temporal Verification and Checkpoint Selection

- In reality, grid workflows should be time constrained.
- Three types of temporal constraints
  - Upper Bound
  - Lower Bound
  - Fixed-time
- Temporal verification to identify any temporal violations in grid workflow specification and execution

## Temporal Verification and Checkpoint Selection

- At build-time stage, temporal verification is static. No need of finding out where to verify.
- At run-time execution stage, along grid workflow execution we may need to verify at some activities while at some other activities we may not.
- So, at run-time execution stage, where to verify?
- Not efficient if at all activities
- Not effective if some necessary checkpoints are omitted.

## Temporal Verification and Checkpoint Selection

- So, how to select necessary yet sufficient checkpoints?
- Research topic: *CSS: Checkpoint Selection Strategy*
- Seven typical strategies proposed, $CSS_1$~$CSS_7$
  - *CSS1: all activities*
  - *CSS2: start and end points of each activity*
  - *CSS3: user-defined static activity points*
  - *CSS4: start activity and decision activities*
  - *CSS5: each activity – completion duration > estimated maximum duration*
  - *CSS6: each activity – completion duration > mean duration*
  - *CSS7: each activity – completion duration > minimum proportional time redundancy + mean duration*

## Problem Analysis

- Existing *CSSs* often
  - select some unnecessary checkpoints
  - omit some necessary checkpoints
- Unnecessary checkpoints -> Unnecessary verification
- Omitted checkpoints -> Omitted verification
- Clearly, neither is desirable.
- The question: can we develop a *CSS* which can select necessary yet sufficient checkpoints ONLY?
- We answer the question positively by presenting one.

SWIN BUR NE · SWINBURNE UNIVERSITY OF TECHNOLOGY

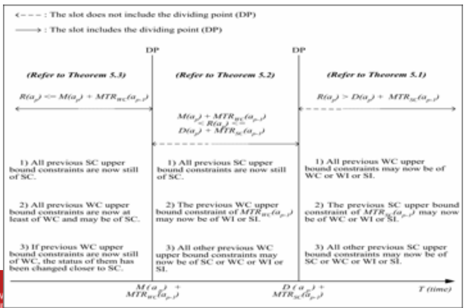
## A New Checkpoint Selection Strategy

- Take upper bound constraints as the example…
- Four types of temporal consistency:
  - SC – Strong Consistency
  - WC – Weak Consistency
  - WI – Weak Inconsistency
  - SI – Strong Inconsistency
- Basic idea: an activity is a checkpoint if at least one upper bound constraint is violated.
- A new concept: Minimum Time Redundancy
  - Minimum one of all time redundancies
  - SC time redundancy
  - WC time redundancy

SWIN BUR NE · SWINBURNE UNIVERSITY OF TECHNOLOGY

## A New Checkpoint Selection Strategy

- Relationships between SC & WC time redundancy and SC, WC, WI & SI [2].




SWIN BUR NE · SWINBURNE UNIVERSITY OF TECHNOLOGY

## A New Checkpoint Selection Strategy

- Three conclusions from the above figure
- Conclusion 1 - $R(a_p) > D(a_p) + MTR_{SC}(a_{p-1})$:
  - Verify all previous SC and WC upper bound constraints.
  - At least one previous SC upper bound constraint which is violated and now is not of SC.
  - It is exactly the one whose SC time redundancy at $a_{p-1}$ is $MTR_{SC}(a_{p-1})$.

SWIN BUR NE · SWINBURNE UNIVERSITY OF TECHNOLOGY

## A New Checkpoint Selection Strategy

- Conclusion 2 - $M(a_p) + MTR_{WC}(a_{p-1}) < R(a_p) \leq D(a_p) + MTR_{SC}(a_{p-1})$:
  - Verify all previous WC upper bound constraints only
  - At least one previous WC upper bound constraint which is violated and now is not of SC and WC.
  - It is exactly the one whose WC time redundancy at $a_{p-1}$ is $MTR_{WC}(a_{p-1})$.

SWIN BUR NE · SWINBURNE UNIVERSITY OF TECHNOLOGY

## A New Checkpoint Selection Strategy

- Conclusion 3 - $R(a_p) \leq M(a_p) + MTR_{WC}(a_{p-1})$:
  - No need to verify all previous SC upper bound constraints.
  - The status of previous WC ones has been changed closer to SC (can even be changed to SC sometimes). No need to verify.
    - *If a previous WC upper bound constraint is still of WC after execution of $a_p$, the previous handling or adjustment on it can be carried forward. Hence, we need not do anything further to it. That is to say, we need not verify it.*

SWIN BUR NE · SWINBURNE UNIVERSITY OF TECHNOLOGY

## A New Checkpoint Selection Strategy

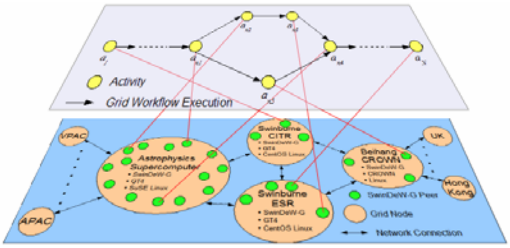
- Our strategy: $CSS_{MTR}$
- *At activity* $a_p$ :
  - *If* $R(a_p) > D(a_p) + MTR_{SC}(a_{p-1})$, *take* $a_p$ *as a checkpoint.*
    - *For verifying all previous SC and WC upper bound constraints*;
  - *If* $M(a_p) + MTR_{WC}(a_{p-1}) < R(a_p) \leq D(a_p) + MTR_{SC}(a_{p-1})$, *take* $a_p$ *as a checkpoint*
    - *For verifying all previous WC upper bound constraints only*;
  - *If* $R(a_p) \leq M(a_p) + MTR_{WC}(a_{p-1})$, *do not take* $a_p$ *as a checkpoint*.

## A New Checkpoint Selection Strategy

- Necessity
  - Theorem (*Necessity*). Along grid workflow execution, all checkpoints selected by $CSS_{MTR}$ are necessary, i.e. there are no any unnecessary checkpoints.
- Sufficiency
  - Theorem (*Sufficiency*). Along grid workflow execution, the checkpoints selected by $CSS_{MTR}$ are sufficient, i.e. there are no any omitted checkpoints.
- Proofs can be referred to [2].

## Simulation Evaluation

- Performed on SwinDeW-G [7]
- Overall picture --- now part of SwinGrid

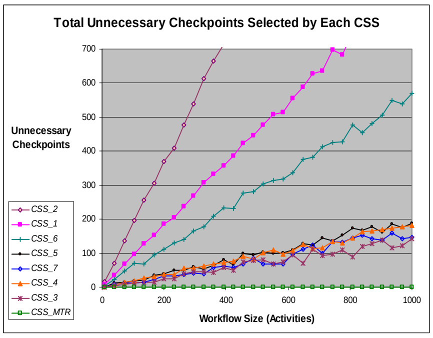



## Simulation Evaluation

- Simulation process along grid workflow execution
  - Sub-process 1: Execute $CSS_1$~$CSS_7$ and $CSS_{MTR}$ to output their checkpoint selection results at each activity.
  - Sub-process 2: Carry out temporal verification to check out whether an activity should be selected as a checkpoint.
- The checkpoints selected by sub-process 2 are necessary and sufficient.
- Compare the results of two sub-processes
  - Identify unnecessary and omitted checkpoints of $CSS_1$~$CSS_7$ and $CSS_{MTR}$

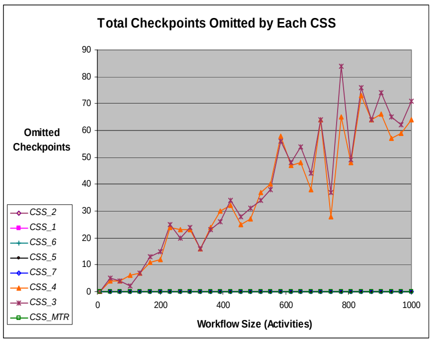
## Simulation Evaluation

- Simulation results – unnecessary checkpoints by each *CSS*




## Simulation Evaluation

- Simulation results – checkpoints omitted by each *CSS*



## Simulation Evaluation

- Conclusions from the simulation
  - Necessity and sufficiency of $CSS_{MTR}$
  - Significant improvement on checkpoint selection over $CSS_1$~$CSS_7$

## Conclusions

- Unnecessary checkpoints – unnecessary temporal verification → not efficient
- Omitted checkpoints – omitted temporal verification → not effective
- Existing CSS →unnecessary checkpoints and/or omitted checkpoints
- A novel strategy called $CSS_{MTR}$ is developed which can only select necessary yet sufficient checkpoints.
- Rigorous proof and experimental simulation demonstrate:
  - Necessity and sufficiency
  - Significant improvement over other existing representative strategies

## Questions

- Thanks for your patience and attention!
- Questions?

## References


- [1] J. Chen and Y. Yang. *A Taxonomy of Grid Workflow Verification and Validation*. Concurrency and Computation: Practice and Experience, ISSN: 1532-0626, Wiley, 2007, to appear, http://www.ict.swin.edu.au/personal/jchen/papers/VVTaxonomy.pdf.
- [2] J. Chen and Y. Yang, *Adaptive Selection of Necessary and Sufficient Checkpoints for Dynamic Verification of Temporal Constraints in Grid Workflow Systems*. ACM Transactions on Autonomous and Adaptive Systems (TAAS), accepted, 2007, to appear, http://www.ict.swin.edu.au/personal/jchen/papers/ACMTAASCheckpoint.pdf.


## References


- [3] Cybok, D. 2006. A Grid Workflow Infrastructure. Concurrency and Computation: Practice and Experience, Special Issue on Workflow in Grid Systems 2006; 18(10): 1243-1254.
- [4] The Gridbus Middleware (The Gridbus Project): http://www.gridbus.org/middleware/.
- [5] IEEE. Standard Glossary of Software Engineering Terminology. *In IEEE Software Engineering Standards Collection*, IEEE, Std 610.12-190, 1994.


## References


- [6] IEEE-SA Standards Board. *IEEE Standard for Software Verification and Validation*, IEEE Std 1012, 1998.
- [7] SWINDEW-G Team. 2007. System Architecture of SwinDeW-G. http://www.ict.swin.edu.au/personal/jchen/SwinDeW-G/System_Architecture.pdf, accessed on June 9, 2007.
- [8] Yu, J., Buyya, R. 2005. A Taxonomy of Workflow Management Systems for Grid Computing. *Journal of Grid Computing* 2005; 3(3-4): 171-200.