



COORDENAÇÃO ENTRE TÉCNICAS DE SIMULAÇÃO PARA INTEROPERABILIDADE DE COMPONENTES EM MODELOS HETEROGÊNEOS

ACÁCIA DOS CAMPOS DA TERRA^{1,2*}, BRÁULIO ADRIANO DE MELLO^{2,3}

1 Introdução/Justificativa

Uma das etapas da simulação é a construção de modelos de simulação, compostos por processos lógicos (LPs). Em simulações distribuídas e heterogêneas (que agregam variedade de recursos e tecnologias), a transmissão de mensagens fora de ordem pelos processos é comum. Para a solução desse problema, o uso de um relógio global de sincronização, o *Global Virtual Time* (GVT), é a prática mais utilizada.

A barreira no tempo (GVT) pode bloquear processos desnecessariamente, afetando negativamente a performance da simulação. Para solucionar o problema, é atribuída aos processos uma predição de comportamento, chamada de *lookahead*, que funciona através de promessas, informando até quando o processo estará em um “estado seguro”.

O estado seguro significa, para um processo, que pode avançar no tempo e evita situações como: a) caso um processo avance demais no tempo e receba eventos atrasados, gerando uma violação de tempo e causando o fim da simulação e b) quando um processo recebe um valor menor do que o ideal para o *lookahead*, a performance da simulação é afetada negativamente. Existem diversas estratégias encontradas na literatura com o intuito de obter um valor mais próximo do ideal para o *lookahead*. Foram estudadas algumas destas estratégias e feitas análises comparativas com a abordagem utilizada pelo *Distributed Co-Simulation Backbone* (DCB), uma arquitetura de suporte à execução distribuída de modelos heterogêneos de simulação.

2 Objetivos

O objetivo deste trabalho foi comparar diferentes estratégias para o cálculo da predição de comportamento (*lookahead*) em simulações conservativas, avaliando suas respectivas características e aptidões para uma possível implementação no ambiente do DCB.

1 Bacharela em Ciência da Computação, Universidade Federal da Fronteira Sul, *campus Chapecó*, **Bolsista** UFFS, contato: terra.acacia@gmail.com

2 Sistemas Embarcados

3 Doutor em Ciência da Computação, Universidade Federal do Rio Grande do Sul, **Orientador**.

3 Material e Métodos/Metodologia

Para a realização do trabalho proposto neste projeto, foi elaborado um cronograma exibido na Figura 1. O cronograma e as tarefas nele contidas foram ainda divididos em três etapas principais, a primeira uma etapa de ambientação em relação ao assunto tratado e estudo de referencial teórico, a segunda uma etapa de estudo e análise de trabalhos correlatos ao que estava sendo feito e a terceira uma etapa de análise para adaptação de soluções pensando no ambiente do DCB.

Atividades a serem desenvolvidas:	Meses											
	1	2	3	4	5	6	7	8	9	10	11	12
Estudo de referencial teórico em abrangência e profundidade no tema do projeto e de acordo com as metas do ano	█	█	█	█	█							
Estudo/revisão da arquitetura do DCB e ambientação com o protótipo vigente em operação			█	█	█							
Estudo de soluções semi-conservativas para o controle de lookahead em modelos híbridos no tempo				█	█	█	█					
Especificação de mecanismos para controle dinâmico do intervalo de lookahead em tempo de simulação						█	█	█	█			
Incorporar as soluções propostas para gerenciamento de lookahead no protótipo do DCB, planejar e executar operações de verificação e validação dos resultados										█	█	█
Elaboração de relatórios parciais e final e atender atribuições conforme termo de compromisso											█	█

Figura 1. Cronograma de atividades

4 Resultados e Discussão

Para a realização deste trabalho foram estudadas estratégias para obtenção do *lookahead* através da leitura de artigos. Alguns algoritmos foram escolhidos como os mais relevantes e para cada um deles foi feita uma análise e uma relação de itens que deveriam ser alterados (ou incluídos ou excluídos) no ambiente do DCB para que o algoritmo fosse implementado e testado.

Para o algoritmo proposto por (WENTONG; TURNER, 1995) há a necessidade da implementação das ‘mensagens nulas *carrier*’, então devem ser adicionadas estruturas condicionais ao código do embaixador de recepção de mensagem (*EDCB.java*), para identificação de recebimento e das mensagens. Também é preciso acrescentar um método que as crie no formato $M = (t, route.inf, la.inf, carrier.null)$ e para isto é necessário utilizar uma estrutura (por exemplo um *ArrayList*), com o intuito de armazenar as informações de rota, importante na identificação de possíveis ciclos no modelo.

Com a flexibilização do *lookahead* através do *la.inf*, deve ser retirado o valor fixo de 100, que é atribuído como *lookahead* no código *Fedgvt.java*. Também não será mais necessário somar o *lookahead* ao GVT, sempre que este for atualizado (no método *calcGVT*), pois como o *lookahead* não é mais fixo, não irá servir mais apenas como barreira de tempo, mas sim como predição de comportamento. Ao fazer as alterações citadas, os valores do GVT e do *lookahead* serão os mesmos (o menos dos LVTs), então pode-se passar a utilizar apenas uma das variáveis.

Já na proposta de (FU; BECKER; SZCZEBICKA, 2013) é trabalhado com um tipo de processo chamado semi-conservativo. O DCB trabalha apenas com processos síncronos (conservativos) e assíncronos (otimistas), então é preciso acrescentar o tratamento para este tipo híbrido de processos, dessa forma o DCB pode abranger diferentes modelos de simulação. A implementação do método para tratamento desses processos deve ser feito juntamente com os outros em *EDCB.java*, com implicações em outras classes também.

Os quatro subsistemas utilizados na abordagem (escalonador, estimador, reparador e otimizador) estão presentes em cada um dos LPs e, como o DCB se propõe a manter os elementos independentes, a implementação destes subsistemas deve ser feita no próprio escopo do DCB, podendo ser incluídos em *Fedgvt.java*. A mensagem de controle trocada entre o otimizador e o escalonador deve ser tratada pelos embaixadores de envio e recepção de mensagens (*EDCB.java* e *EF.java*).

Elaborar e implementar métricas para estimar a quantidade de ‘erros aceitáveis’ deste algoritmo é essencial. O cálculo é feito após cada envio de mensagem, no método de atualização de processos síncronos (*UpdateSync*), pertencente à classe *EDCB.java*.

Para as situações onde ocorram erros apesar do *lookahead*, pode ser feito um rollback, que no DCB já é feito apenas para os processos assíncronos.

Os mesmos autores propuseram outra abordagem (FU; BECKER; SZCZEBICKA, 2014), chamada de *lookahead* condicional. Nesta abordagem o GVT precisa ser removido do DCB. São utilizados dois tipos de mensagens: as mensagens que carregam o *lookahead* condicional e o anúncio de invalidação, emitido quando as condições para aplicar o *lookahead* não forem válidas. As implementações devem ser feitas nos embaixadores de comunicação *EDCB.java* e *EF.java*. A forma como o *lookahead* é atribuído aos processos em *Fedgvt.java* também deve ser alterada, deixando de ser um valor fixo.

5 Conclusão



As atividades propostas no plano de trabalho apresentado foram todas cumpridas em prazos muito próximos dos previstos. Apenas uma das atividades foi substituída durante a vigência do projeto, sendo esta a atividade onde seriam feitas algumas implementações e validações, sendo uma decisão tomada em conjunto com o orientador. Esta decisão não traz prejuízos ao projeto, pois a implementação foi feita em escopo de um trabalho de conclusão de curso. É visível a necessidade de efetuar uma alteração na que era a atual implementação do *lookahead* no DCB, que estava com valor fixo.

As comparações feitas durante este projeto, onde foram analisadas as mudanças necessárias em questão de código, são importantes para uma melhor visualização da situação do DCB antes de ser feita a implementação. Para estudos futuros, além de um estudo aprofundado sobre os processos semi-conservativos, há interesse em efetuar uma pesquisa de estratégias para flexibilização do *lookahead* que possibilitem a retirada do relógio global de sincronização durante a simulação, para evitar que processos sejam desnecessariamente bloqueados.

Referências

FU, Desheng; BECKER, Matthias; SZCZERBICKA, Helena. Accelerating Distributed Discrete Event Simulation through Exchange of Conditional Look-Ahead. **2014 Ieee/acm 18th International Symposium On Distributed Simulation And Real Time Applications**, [s.l.], p.183-189, out. 2014. IEEE. <http://dx.doi.org/10.1109/ds-rt.2014.30>.

FU, Desheng; BECKER, Matthias; SZCZERBICKA, Helena. On the potential of semi-conservative look-ahead estimation in approximative distributed discrete event simulation. **In Proceedings of the 2013 Summer Computer Simulation Conference (SCSC '13)**. Society for Modeling & Simulation International, Vista, CA, 2013, Article 28 , 8 pages.

WENTONG, Cai; TURNER, Stephen J. An Algorithm for Reducing Null Message of the CMB Approach in Parallel Discrete Event Simulation. Technical report, 1995. Research Report No 333.

Palavras-chave: *Lookahead*; Simulação; Distribuída; Heterogênea; Semi-conservativo.

Financiamento

UFFS