

# Em Direção à Programação Distribuída na Seleção de Planos em Sistemas Multi-Agentes

Renato B. Hoffmann, Dalvan Griebler

<sup>1</sup> Escola Politécnica, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)  
Porto Alegre – RS – Brasil

renato.hoffmann@edu.pucrs.br, dalvan.griebler@pucrs.br

***Resumo.** Sistemas multi-agentes são compostos por múltiplos agentes autônomos que interagem com um ambiente e entre si para atingir objetivos específicos. Jason, uma linguagem multi-agentes popular modela esses sistemas através de crenças, metas e planos, que é atualmente realizada através de uma varredura linear. Sendo assim, essa pesquisa propõe investigar a seleção dos planos de maneira paralela e em um sistema distribuído.*

## 1. Introdução

Sistemas reativos são especificados e descritos de acordo com seu comportamento, ou seja, como eles continuamente interagem com seu ambiente. Exemplos são servidores *web*, sistemas bancários, e até mesmo sistemas operacionais. Outro exemplo mais complexo são os sistemas multi-agentes [Cardoso and Ferrando 2021]. Agentes recebem uma tarefa e devem determinar eles mesmos como realizá-la. Esse processo de decisão é realizado individualmente ou cooperando, competindo com outros agentes.

A linguagem multiagente *AgentSpeak* [Bordini and Hübner 2006] utiliza um modelo teórico para representar o sistema de ação de um agente baseado em crenças, metas e planos. Crenças são informações que o agente possui sobre o mundo, que podem ser erradas ou desatualizadas. Metas representam um estado do ambiente que o agente gostaria de alcançar. Planos são uma sequência de ações que o agente decidiu tomar para tentar realizar pelo menos uma de suas metas.

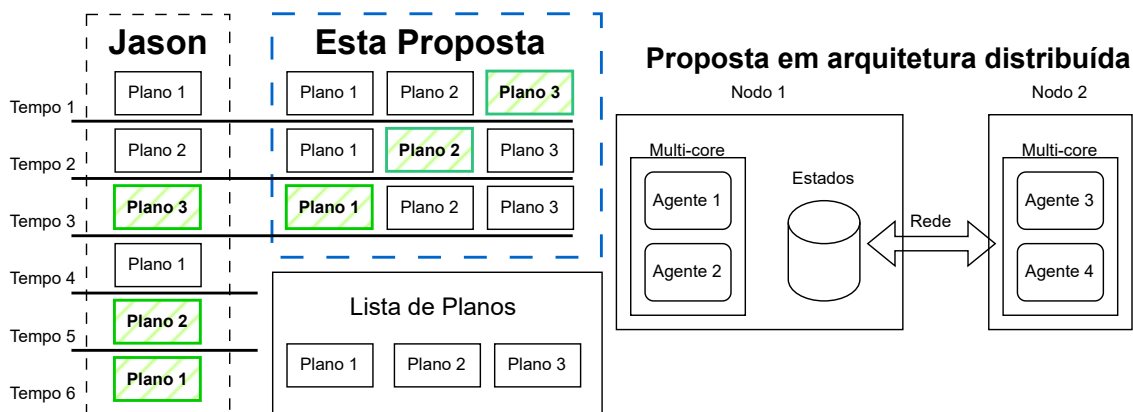
Múltiplos planos representam opções de ações que os agentes podem selecionar. Além do mais, múltiplos planos podem ser executados ao mesmo tempo, de maneira que existem diferentes ordens de execução dos planos que podem resultar no mesmo resultado final. Entretanto, quando um plano é selecionado, ele deve ser executado até o final, mesmo que uma mudança no ambiente não o torne mais a melhor opção. Portanto, a seleção dos planos não só implica na eficiência do sistema mas como também influencia na qualidade do resultado final [Marir et al. 2016].

Na atual versão de Jason, que implementa *AgentSpeak* através de um interpretador descrito em Java, os planos podem ser descritos em uma lista. A cada ciclo de decisão, um agente percorre sua lista de planos linearmente, avalia critérios de execução, e executa o primeiro plano que atende todos os critérios. Dessa forma, a execução segue um padrão linear e a ordem de descrição dos planos impacta diretamente no resultado final.

## 2. Proposta de Pesquisa

A proposta de pesquisa deste trabalho consiste na utilização de computação distribuída na seleção dos planos. A esquerda da Figura 1 demonstra, de uma maneira alto-nível,

o processo de seleção de planos na metodologia atual Jason e também a metodologia proposta neste trabalho. Neste exemplo, a lista de planos contém 3 planos, onde só existe uma ordem de execução: Plano 3, seguido do 2 e 1. Na execução Jason, a cada tempo, um plano é avaliado e, quando possível, executado (representado na imagem pela rasura e cor verde). Portanto, um ciclo de decisão pode levar de 1 até 3 tempos. Já na versão proposta nesta pesquisa, cada ciclo de decisão ocorre em apenas 1 tempo. Todos os planos são avaliados em paralelo, e somente o plano possível é executado.



**Figura 1. Demonstração dos processos de seleção de planos e exemplo de uma arquitetura distribuída para avaliação.**

Existem dois benefícios para a metodologia de seleção de planos paralelos: menor tempo de busca de planos menor e maior espaço de busca nos planos, o que implica em potencialmente um resultado com maior qualidade. A direita da Figura 1 demonstra uma possível implementação do sistema multi-agentes, onde cada agente possui um processo de seleção de planos. Neste possível exemplo, existem 4 agentes, que são distribuídos em 2 nodos computacionais. Em um dos nodos, são armazenados os estados globais do sistema ou metadados sobre os estados dos agentes. Múltiplos agentes com muitos planos tornam cada vez mais viável a execução em sistemas distribuídos. Entretanto, a identificação do limiar da carga de trabalho que torna viável a execução distribuída também é uma contribuição prevista para este trabalho. Por fim, a estratégia de implementação desta proposta de pesquisa envolve o desenvolvimento de um sistema OpenMPI que executa processos de seleção de proposta e comunique somente os resultados para um estado centralizado a cada ciclo de decisão.

## Referências

- Bordini, R. H. and Hübner, J. F. (2006). Bdi agent programming in agentspeak using jason. In Toni, F. and Torroni, P., editors, *Computational Logic in Multi-Agent Systems*, pages 143–164, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Cardoso, R. C. and Ferrando, A. (2021). A review of agent-based programming for multi-agent systems. *Computers*, 10(2).
- Marir, T., Mokhati, F., Seridi-Bouchelaghem, H., Acid, Y., and Bouzid, M. (2016). Qm4mas: A quality model for multi-agent systems. *International Journal of Computer Applications in Technology*, 54:297.