

Além da Escalabilidade: Inteligência de Enxames Afetada por Campos Magnéticos em Espaços de Tuplas Distribuídos

Henrique Duarte Lima¹, Luiz A. de P. Lima Jr.¹, Alcides Calsavara¹,
Henri F. Eberspächer¹, Ricardo C. Nabhen¹, Elias P. Duarte Jr²

¹ Programa de Pós-Graduação em Informática (PPGIA)
Pontifícia Universidade Católica do Paraná (PUCPR)

²Departamento de Informática - Universidade Federal do Paraná (UFPR)

henrique@engduarte.com, {laplima,alcides}@ppgia.pucpr.br,
{henri.eberspacher,ricardo.nabhen}@pucpr.br, elias@inf.ufpr.br

Abstract. Tuple Spaces simplify the process of creating distributed systems providing a model for component communication and synchronization that is loosely coupled in terms of both space and time. However, traditional implementations in high-performance environments present low scalability. Although “anti-over-clustering” techniques allow the construction of more scalable solutions, the excessive dispersion of elements throughout the network compromises the overall performance. In this work we propose a strategy to cope with this problem: Magnetic SwarmLinda, an approach based on swarm intelligence and on the concept of virtual magnetic fields. Simulation results show that the proposed strategy significantly improves the system’s performance particularly the tuple retrieval latency.

Resumo. Espaços de Tuplas simplificam a construção de sistemas distribuídos provendo um modelo de comunicação e sincronização de componentes com desacoplamento espacial e temporal. Porém, implementações tradicionais em ambientes com demandas de alto desempenho exibem baixa escalabilidade. Embora técnicas de “anti-over-clustering” permitam a construção de soluções mais escaláveis, mostramos que o espalhamento excessivo resultante reduz o desempenho do sistema. Para lidar com o problema, o Magnetic SwarmLinda, um método baseado em inteligência de enxames e no conceito de campos magnéticos virtuais, é apresentado. Resultados de simulação confirmam que a estratégia proposta melhora efetivamente o desempenho do sistema em termos da latência da recuperação de tuplas.

1. Introdução

Os Espaços de Tuplas Distribuídos – ou simplesmente Espaços de Tuplas [Atkinson 2008] – são um modelo baseado em memória associativa compartilhada para comunicação e coordenação de processos em sistemas paralelos e distribuídos. Os dados são armazenados no espaço compartilhado na forma de *tuplas*, que são listas ordenadas de dados de diferentes tipos. A tupla $\langle \text{“SBRC”}, 2017 \rangle$, por exemplo, contém valores para dois elementos, o primeiro do tipo *string* e o segundo, do tipo número inteiro. Processos clientes de um Espaço de Tuplas Distribuído são *produtores* e/ou *consumidores* de tuplas.

A diferença fundamental entre os Espaços de Tuplas e outros modelos consiste no fato das tuplas nunca serem explicitamente endereçadas. A busca de tuplas no Espaço de Tuplas é feita de forma associativa, através de *templates*, que especificam um “formato” da tupla, bem como restrições de valores (por exemplo, o *template* $\langle [string], 2017 \rangle$ representa todas as tuplas com 2 elementos, sendo o primeiro uma *string*, e o segundo o valor 2017). Outra característica importante é o fato de que processos que utilizam um Espaço de Tuplas não interagem diretamente entre si: toda comunicação acontece através da inserção e busca/remoção de tuplas no Espaço de Tuplas. Por causa desta característica, há um baixo acoplamento de processos no espaço e no tempo. No espaço, porque processos podem estar espalhados de forma arbitrária sobre uma rede de computadores. No tempo, porque processos não precisam estar executando no mesmo instante de tempo para poderem se comunicar – um processo pode criar e inserir uma tupla no Espaço de Tuplas para, logo em seguida, encerrar sua execução. Outro processo mais tarde pode iniciar a sua execução e consumir aquela tupla armazenada.

O modelo de Espaço de Tuplas foi originalmente proposto e implementado no contexto da linguagem Linda [Gelernter and Bernstein 1982] e é, portanto, frequentemente associado à própria linguagem. As duas operações básicas para inserir e recuperar tuplas no Espaço de Tuplas são *out* e *in*, respectivamente. A operação *in* para um determinado *template* remove do Espaço de Tuplas a primeira tupla encontrada (em uma escolha não determinística) que possua o formato especificado pelo *template*. Caso não haja no Espaço de Tuplas nenhuma tupla do formato solicitado, o processo solicitante fica bloqueado até que uma tupla apropriada seja posteriormente publicada por meio da operação *out*.

Apesar de terem seus méritos reconhecidos, os Espaços de Tuplas originais apresentam problemas de escalabilidade em ambientes de alto desempenho. Nos sistemas originais, e inclusive na implementação JavaSpaces [Waldo et al. 1997], um componente central é responsável pelo processamento que antecede tanto a inserção como a recuperação de tuplas. Diversos esforços (descritos na Seção 2 deste trabalho) foram realizados para eliminar este componente centralizado e melhorar o desempenho sobretudo da operação *in*. Estas abordagens distribuídas tornaram possível a aplicação dos Espaços de Tuplas em novas áreas como [Maia et al. 2016] a *Internet of Coisas* (IoT) (na comunicação com baixo nível de acoplamento entre sensores, atuadores e aplicações) e aplicações que executam em nuvens [Hari 2012].

Uma das abordagens descentralizadas com potencial para tais utilizações é o *SwarmLinda* [Menezes and Tolksdorf 2003] que é uma implementação baseada em inteligência de enxames. No *SwarmLinda*, sempre que uma operação *out* é executada, uma formiga é instanciada. Essa formiga, denominada de *tuple-ant*, é responsável por percorrer os nós que compõem o Espaço de Tuplas para encontrar um nó adequado para depositar a tupla. O critério utilizado para depositar uma tupla é a quantidade de tuplas “semelhantes” à tupla carregada pela formiga que já estão depositadas no nó atual. Isto permite a formação de *clusters* de tuplas similares com o objetivo de acelerar buscas posteriores. A estratégia, entretanto, não resolve o problema da escalabilidade, pois conforme a quantidade de tuplas semelhantes depositadas aumenta, ocorre uma degradação importante do desempenho devido a concentração de processamento em certos nós. Além disso, nós sobrecarregados acabam tornando-se mais atrativos por terem uma alta concentração

de tuplas similares – e a situação tende a piorar com o passar do tempo.

Em [Casadei et al. 2007] é proposta a abordagem *Anti-Over-Clustering* para evitar a sobrecarga de nós devido à concentração excessiva de tuplas. Nesta abordagem, a probabilidade de uma formiga depositar uma tupla também aumenta de acordo com a quantidade de tuplas similares. No entanto, quando a quantidade de tuplas aproxima-se de um limiar, esta probabilidade diminui ainda que a quantidade de tuplas aumente. Desta forma, as *tuple-ants* evitam depositar suas tuplas em nós sobrecarregados. Infelizmente, esta estratégia obriga as formigas a continuarem a exploração do Espaço de Tuplas, produzindo um maior consumo de recursos de processamento e comunicação do sistema. Além disso, a estratégia leva à formação de *clusters* dispersos no Espaço de Tuplas, o que, no balanço final, reduz sensivelmente o desempenho da recuperação de tuplas.

Este trabalho descreve o *Magnetic SwarmLinda*: uma estratégia baseada em inteligência de enxames afetada por “campos magnéticos virtuais” para resolver o problema da dispersão de *clusters* distribuídos de forma a manter a escalabilidade e eficiência na recuperação de tuplas. Campos Magnéticos Virtuais implementam um modelo distribuído autônomo de relações de auxílio entre nós formando redes *overlay* para o encaminhamento de mensagens de aplicação aos nós mais aptos (i.e., aqueles com maior força de atração magnética). A intensidade do campo magnético é parâmetro dinâmico dependente da aplicação. O modelo foi originalmente proposto para o encaminhamento de mensagens a destinatários móveis e para implementar mecanismos distribuídos de balanceamento de carga [de Paula Lima Jr. and Calsavara 2010] (neste caso, quanto maior a ociosidade de um nó, maior a sua força de atração). Na abordagem proposta no presente trabalho, campos magnéticos atuam como “campos de força” que produzem um comportamento “anormal” nas formigas, protegendo cada nó de uma sobrecarga resultante de uma concentração excessiva de tuplas similares. O *Magnetic SwarmLinda* foi implementado e os resultados obtidos confirmam que a formação de *clusters* de *clusters* (grupos de nós vizinhos contendo tuplas similares) resultantes do método proposto produz um aumento considerável do desempenho global do sistema em comparação com todas as abordagens concorrentes.

O restante do trabalho está organizado da seguinte maneira. Na Seção 2, são apresentados trabalhos relacionados, inclusive os campos magnéticos virtuais. Na Seção 3, o *Magnetic SwarmLinda* é descrito. Resultados experimentais são apresentados na Seção 4, comparando os benefícios da estratégia proposta com o concorrente mais direto, o *Anti-Over-Clustering*. As conclusões seguem na Seção 5.

2. Trabalhos Relacionados

Nesta Seção, serão apresentadas as principais abordagens *distribuídas* de Espaços de Tuplas. Iniciamos pelo WCL [Rowstron 1998], que é uma implementação de múltiplos Espaços de Tuplas distribuídos geograficamente. Cada Espaço de Tuplas é executado sobre um único nó, sendo, portanto, centralizado. No entanto, o WCL é capaz de realizar a migração de Espaços de Tuplas de forma transparente entre os nós que compõem o sistema. A migração pode ocorrer com o intuito de reduzir a latência de comunicação com um determinado Espaço de Tuplas, migrando este para um nó de onde provém a maior parte das requisições. Além disso, a migração pode ocorrer quando a carga em um determinado nó está elevada, permitindo o balanceamento de carga. No entanto, esta

estratégia é eficaz apenas quando a sobrecarga do nó é resultado da soma das cargas de vários Espaços de Tuplas. Quando um determinado Espaço de Tuplas é capaz de sobrecarregar sozinho um nó, o balanceamento de carga torna-se inócuo.

Linda in Mobile Environment (LIME) [Picco et al. 1999] é uma implementação de Espaço de Tuplas que permite que agentes móveis transitem entre dispositivos móveis heterogêneos. Cada agente tem uma visão do sistema em termos do conjunto de tuplas disponíveis e esta modifica-se, de maneira transparente, conforme ocorrem mudanças de conectividade entre os *hosts* dos demais agentes. É possível também que o volume de tuplas se altere em função da movimentação de agentes entre os *hosts*. A distribuição das tuplas no LIME ocorre de maneira diferenciada. Cada tupla adicionada no Espaço de Tuplas é armazenada no *host* executando o agente responsável pela inserção desta tupla. Quando um agente migra de *host*, este carrega consigo todas as suas tuplas que ainda não foram recuperadas. Devido a este estilo de distribuição das tuplas, quando um agente não consegue encontrar uma tupla qualquer na sua porção de um Espaço de Tuplas compartilhado com outros agentes, este agente precisa verificar os demais nós do Espaço de Tuplas até encontrar a tupla desejada, o que envolve atrasos sem limite conhecido.

O *Dtuples* [Jiang et al. 2006] é uma implementação de Espaço de Tuplas construída sobre uma tabela de *hash* distribuída com o objetivo de simplificar a comunicação entre agentes em ambientes distribuídos. Há uma restrição imposta pelo *Dtuples*, na qual o elemento da posição inicial da tupla deve ser uma *string* com o nome da tupla, sendo que este valor não necessita ser único. Esta restrição deve-se ao fato da implementação utilizar o *hash* deste primeiro elemento para determinar o local de armazenamento da tupla. Na recuperação de uma tupla também é necessário informar este nome no primeiro campo do *template*, permitindo que o Espaço de Tuplas saiba em qual local deve ser executado o processo de busca. Infelizmente, esta abordagem permite uma concentração indesejável de carga em poucos nós quando há muitas colisões de *hash*. Além disso, essa restrição impõe mudanças na interface tradicional dos Espaços de Tuplas.

O *Tupleware* [Atkinson 2008] é um *middleware* inspirado em Linda que implementa um Espaço de Tuplas que dispõe de um algoritmo descentralizado para recuperação de tuplas. As tuplas inseridas por um processo são armazenadas em uma instância local que compõe o Espaço de Tuplas, permitindo assim uma concentração excessiva de tuplas em apenas alguns nós, dependendo das características de comunicação da aplicação. Quando um processo requisita uma determinada tupla que não está disponível localmente, é efetuada uma busca - de forma transparente para a aplicação - nos demais nós que compõem o Espaço de Tuplas. A busca nos demais nós ocorre de maneira sequencial, sendo que os primeiros nós a serem verificados são aqueles que apresentam um elevado *fator de sucesso*. Este valor indica a taxa histórica de sucesso produzida pelas buscas em determinado nó, sendo incrementado em caso de sucesso e decrementado em cenário oposto. Desta forma, nós que eventualmente possuam uma grande porção das tuplas do Espaço de Tuplas serão, provavelmente, muito consultados pelos demais nós, provocando uma distribuição inadequada da carga do Espaço de Tuplas.

O *SwarmLinda* [Menezes and Tolksdorf 2003] é uma implementação de Espaço de Tuplas que usa técnicas de inteligência de enxames. Nessa abordagem, as características do Espaço de Tuplas emergem das interações entre indivíduos simples, inspirados pelas formigas, que cooperam de forma descentralizada. As decisões tomadas por

estes indivíduos baseiam-se unicamente em informações locais, tornando desnecessário o conhecimento do estado global do sistema, evitando assim uma troca intensa de mensagens. A estratégia de movimentação das formigas utiliza como critérios a intensidade do feromônio e a concentração local de tuplas semelhantes. Quando uma formiga não pode atingir seu objetivo no nó atual, ela obtém informações sobre o número de tuplas semelhantes e a intensidade do feromônio em nós vizinhos para determinar a probabilidade de se mover para cada um desses nós.

Uma formiga é instanciada sempre que uma operação `out` é executada. Essa formiga, denominada de *tuple-ant*, é responsável por percorrer os nós do Espaço de Tuplas para encontrar um nó adequado para depositar a tupla. O critério utilizado para depositar uma tupla é o número de tuplas depositadas no nó que são similares à tupla carregada pela formiga, permitindo a formação de “*clusters*” de tuplas similares. Durante seu percurso, a *tuple-ant* deposita um feromônio, que é específico para o tipo de tupla, em cada nó que é visitado. Além disso, o feromônio também é espalhado nos vizinhos do nó onde a tupla é depositada. Similar à operação `out`, sempre que uma operação `in` uma formiga é instanciada. Esta formiga, que é denominada de *template-ant*, é responsável por percorrer os nós, a fim de encontrar uma tupla compatível com seu *template*. Quando a *template-ant* encontra uma tupla compatível, ela deve retornar ao seu “local de nascimento” carregando a tupla enquanto sinaliza o percurso com feromônio relativo à tupla em questão.

A formação dos *clusters* de tuplas similares permite que as formigas não necessitem percorrer todo o Espaço de Tuplas em busca de seus objetivos. Ao invés disto, cada formiga deve concentrar seus esforços em localizar um caminho, através das trilhas formadas por feromônio, que leve até um *cluster* de tuplas de seu interesse. Neste local, a probabilidade da formiga obter sucesso é maior que nos demais nós. No entanto, conforme a quantidade de tuplas depositadas nos *clusters* aumenta, ocorre uma degradação do desempenho devido ao excesso de formigas que precisam ser processadas nesses nós. Além disso, nós sobrecarregados acabam tornando-se mais atrativos – na perspectiva das formigas – devido a quantidades elevadas de tuplas e feromônio, provocando a formação de um “círculo vicioso” que provoca um aumento crescente da quantidade de tuplas.

[Casadei et al. 2007] propõe a abordagem Anti-Over-Clustering para evitar a sobrecarga de nós devido à concentração excessiva de tuplas. Neste modelo, a probabilidade de depositar uma tupla aumenta de acordo com a quantidade de tuplas similares. No entanto, a partir de um momento em que a quantidade de tuplas aproxima-se de um *threshold*, a probabilidade diminui ainda que a quantidade de tuplas aumente. Desta forma, as *tuple-ants* evitam de depositar suas tuplas em nós sobrecarregados. Infelizmente, esta estratégia obriga que essas formigas continuem a explorar o Espaço de Tuplas, produzindo uma maior carga de processamento sobre o sistema. Além disso, é propiciada a formação de *clusters* similares excessivamente dispersos no Espaço de Tuplas, dificultando a recuperação de tuplas.

3. Proposta: *Magnetic SwarmLinda*

Neste trabalho é apresentada uma estratégia distribuída para a implementação de Espaços de Tuplas Distribuídos denominada *Magnetic SwarmLinda*. A estratégia propõe que o uso da inteligência de enxames seja afetado por Campos Magnéticos Virtuais. O objetivo é prevenir a concentração excessiva de tuplas em um único nó, que pode degradar conside-

ravelmente o desempenho do processo de recuperação de tuplas e, ao mesmo tempo, evitar a dispersão excessiva de tuplas semelhantes, prejudicando o desempenho da recuperação.

A rede é representada por um grafo $G = (V, E)$, sendo V o conjunto de nós e E é o conjunto de arestas bidirecionais entre um par de nós. $N = |V|$ é o número de nós e $M = |E|$ é o número de arestas de G . $NH(i)$ é definido como o conjunto de vizinhos diretos do nó $i \in V$ (i.e., $NH(i) = \{j : (i, j) \in E \vee (j, i) \in E\}$).

As próximas seções detalham o modelo proposto especificando o comportamento das *tuple-ants*, bem como o suporte ao processo de tomada de decisão para depositar/recuperar tuplas.

3.1. Comportamento das *tuple-ants*

Quando um nó recebe uma requisição para executar a operação *out*, os seguintes passos são executados:

1. O nó atribui a tupla informada a um novo agente *tuple-ant* que é responsável por depositar a tupla em algum nó do Espaço de Tuplas. Neste momento, o seu tempo de vida máximo (*Time to Live - TTL*) lhe é atribuído em número de saltos.
2. Em seguida, a formiga verifica se há muitas tuplas depositadas no nó corrente que são similares à tupla carregada. Esta informação é utilizada para decidir se a tupla deve ser depositada. A probabilidade de depositar a tupla aumenta à medida que o número de tuplas similares cresce (conforme será descrito na Seção 3.2).
3. Se a formiga tiver depositado a tupla, sua última tarefa antes de morrer é sinalizar o local através da dispersão de feromônio no nó corrente e nos nós vizinhos. Este processo reforça o nó como um local adequado para tuplas similares, permitindo que outras formigas interessadas nesse tipo de tupla possam encontrar este nó.
4. Se a formiga decidir não depositar a tupla no nó corrente, esta deve escolher um nó adjacente para visitar. Esta escolha é feita estocasticamente com base em informações sobre os nós vizinhos (detalhado na Seção 3.3). No entanto, é possível que esta etapa não ocorra devido à ocorrência de um comportamento “estranho” (que será apresentado na Seção 3.5).
5. Como a formiga se move para um novo nó, esta se torna mais velha, i.e., seu *TTL* é decrementado. Quando *TTL* atinge zero, a formiga deposita a sua tupla no nó corrente e dispersa feromônio referente ao tipo da tupla (como descrito na etapa 3) antes de morrer. Caso contrário, a formiga continua a partir da etapa 2.

Este comportamento das *tuple-ants* pode ser simplificado dependendo da equação que define a probabilidade de depositar uma tupla. A verificação do *TTL* pode ser omitida (conforme o fluxograma da Figura 1) caso a equação que define a probabilidade de depositar uma tupla garanta que a tupla será depositada quando o *TTL* atingir zero. No modelo proposto, a equação que define a probabilidade de depositar uma tupla (apresentada na Seção 3.2) garante que toda tupla será depositada quando o *TTL* da formiga responsável pela tupla atingir o valor zero, permitindo tal simplificação.

A decisão de depositar uma tupla baseada na concentração de tuplas ocorre de forma probabilística. A partir da concentração de tuplas similares, a probabilidade $P_{drop}(i, \tau_c)$ de depositar a tupla τ_c no nó atual i é calculada (ver Seção 3.2). Na sequência, a formiga sorteia um número r ($0 \leq r < 1$) e o compara com a probabilidade calculada. Se r for inferior à $P_{drop}(i, \tau_c)$, a formiga deposita a tupla e espalha feromônio referente ao tipo da tupla. Caso contrário, a formiga continua a exploração do Espaço de Tuplas

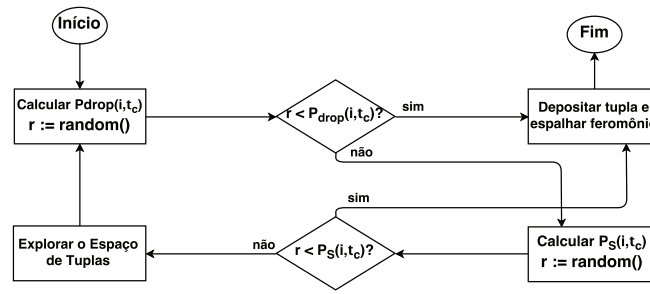


Figura 1. Comportamento simplificado das *tuple-ants*.

(descrito na Seção 3.3) desde que não adote um comportamento estranho (descrito na Seção 3.5). Assim como ocorre na avaliação do $P_{drop}(i, \tau_c)$, a análise da probabilidade de comportamento estranho $P_S(i, \tau_c)$ é realizada comparando o valor da probabilidade com um número aleatório r ($0 \leq r < 1$).

3.2. Probabilidade de Depositar uma Tupla

A probabilidade de depositar uma tupla no nó corrente depende da concentração de tuplas similares no referido nó. A *função de similaridade* – designada por $sim(\tau_A, \tau_B) \in [0, 1]$ – deve ser definida de modo que o grau de similaridade entre duas tuplas τ_A e τ_B possa ser avaliado. Para os experimentos descritos na Seção 4, tuplas são consideradas similares se suas quantidades de campos e os respectivos tipos de cada campo são idênticos.

A concentração de tuplas em determinado nó i de tuplas que são similares à τ_c (a tupla carregada) – designada por $C(i, \tau_c)$ – é dada pela Equação (1). $C(i, \tau_c)$ é determinada através da comparação de τ_c com cada tupla τ_s armazenada em $i \in V$. Caso não exista nenhuma tupla similar à τ_c no nó i , $C(i, \tau_c)$ assume um valor mínimo para permitir uma pequena probabilidade da tupla τ_c ser depositada em um nó i . Nos experimentos descritos na Seção 4, o valor mínimo de $C(i, \tau_c)$ é 0,0001.

$$C(i, \tau_c) = \sum_{\forall \tau_s \in i} sim(\tau_c, \tau_s) \quad (1)$$

A probabilidade de uma formiga depositar sua tupla τ_c em algum nó i – designada por $P_{drop}(i, \tau_c)$ – é dada pela Equação (2). Observe que $P_{drop}(i, \tau_c)$ depende de TTL .

$$P_{drop}(i, \tau_c) = \left(\frac{C(i, \tau_c)}{C(i, \tau_c) + TTL} \right)^2 \quad (2)$$

Claramente, a probabilidade de depositar a tupla aumenta conforme o TTL diminui. Consequentemente, o TTL representa o limite superior de saltos que uma formiga percorre com o objetivo de depositar sua tupla. Além disso, a probabilidade de depositar uma tupla é maior quando a concentração de tuplas similares é elevada, pois a influência de TTL em $P_{drop}(i, \tau_c)$ diminui, permitindo a formação de *clusters* de tuplas similares.

3.3. Movimentação das Formigas

Se uma formiga não atingir seu objetivo no nó atual, ela precisa escolher um nó vizinho para visitar com o intuito de continuar tentando atingir seu objetivo. A fim de aumentar

suas chances, ela deve mover-se em direção ao local que muitos outros indivíduos portadores de tuplas similares foram. Além do mais, a formiga deve considerar o número de tuplas similares no potencial nó de destino. A Equação (3) define a probabilidade de uma formiga em um nó i carregando uma tupla τ_c de mover-se para um outro nó j , onde $Ph(i, \tau_c)$ representa a quantidade atual de feromônio de tipo τ_c presente no nó $i \in V$.

$$P(\tau_c)_{i,j} = \frac{C(j, \tau_c) + Ph(j, \tau_c)}{\sum_{\forall n \in NH(i)} (C(n, \tau_c) + Ph(n, \tau_c))} \quad (3)$$

Conforme o diagrama da Figura 2, a escolha do próximo nó a ser visitado depende de um valor aleatório r ($0 \leq r < 1$). A formiga percorre o vetor $NH(i)$, que contém os nós adjacentes do nó corrente i , comparando o valor de r com a soma P da probabilidade $P(\tau_c)_{i,j}$ do nó que está sendo avaliado (*index* atual) e das probabilidades que já foram avaliadas. O nó escolhido é determinado pela iteração no qual o valor de P atinge um valor igual ou superior ao valor de r . Se *index*, que é utilizado como índice para avaliar os vizinhos de i , atingir o valor máximo ($|NH(i)| - 1$), o valor de P atingirá 1. Desta forma, não há a possibilidade da formiga não escolher o próximo nó a ser visitado.

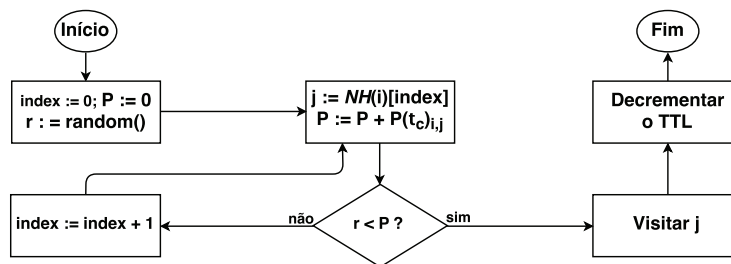


Figura 2. Procedimento de exploração do Espaço de Tuplas.

Quando a formiga encontra o nó j que satisfaz a condição $r < P$, esta desloca-se para o nó j . Além disso, a formiga decrementa o seu *TTL*, ou seja, a formiga envelhece.

3.4. Evaporação de Feromônio

A evaporação de feromônio é uma mecanismo essencial para tornar o sistema adaptável, uma vez que trilhas referentes a regiões que já não tem uma concentração significativa de tuplas devem desaparecer. Um mecanismo de “evaporação” previne que agentes tenham um comportamento caótico que comprometeria o funcionamento adequado do sistema. Além disso, o mecanismo de evaporação torna os caminhos curtos mais atraentes, otimizando assim a quantidade de saltos necessários para que um agente alcance as regiões desejadas do grafo. A Equação (4) define como os feromônios de cada nó i desaparecem conforme o tempo t avança.

$$Ph_t(i, \tau_c) = Ph_{(t-1)}(i, \tau_c)(1 - \rho) \quad (4)$$

Todos os nós do sistema decrementam as suas quantidades de feromônio de acordo com a taxa de evaporação ρ ($\rho \in [0, 1]$). É importante destacar que ρ não deve ser elevado ao ponto de provocar que novas trilhas nunca sejam exploradas. Se ρ é muito baixo, o deslocamento dos indivíduos pode ser afetado negativamente por trilhas que levam à regiões que deixaram de possuir grandes quantidades de tuplas similares.

Apesar de que o uso da inteligência de enxames traz diversos benefícios, um efeito negativo é a sobrecarga dos nós. Embora as trilhas de feromônio constituam um importante mecanismo de orientação das formigas na natureza, este não é o único mecanismo usado por elas. De acordo com [Wajnberg et al. 2010], há também pontos de referência, vibrações, gravidade, bússola solar e luz polarizada que são usadas como critérios de orientação. Em particular, há a influência de campos magnéticos, que, como afirmado por [Banks and Srygley 2003], são capazes de causar uma mudança na orientação de certas espécies de formigas.

[de Paula Lima Jr. and Calsavara 2010] introduz o conceito de campos magnéticos virtuais inspirados da física, para implementar um mecanismo de balanceamento de carga. Neste caso, a intensidade de um campo magnético virtual representa a potência computacional ociosa. Ela é responsável por atrair para os nós mais disponíveis as tarefas para serem processadas. Neste trabalho, a interferência magnética no comportamento das formigas será usada para proteger os nós de forma inteligente de uma sobrecarga que comprometeria o desempenho do sistema.

3.5. Interferência Magnética

Interferência Magnética é um mecanismo cujo objetivo é evitar a concentração de tuplas em poucos (possivelmente sobrecarregados) nós. A inteligência de enxames por si só determina que a probabilidade de um nó receber uma nova tupla é proporcional ao número de tuplas semelhantes que este detém. Esta propriedade melhora o desempenho das subseqüentes operações de recuperação de tupla, pois permite que o processo de busca ocorra orientado a apenas uma determinada região do Espaço de Tuplas. No entanto, quando há uma excessiva quantidade de tuplas em um único nó, é altamente provável que a capacidade de processamento do nó torne-se um gargalo.

A sobrecarga de processamento está relacionada ao fato que um nó contendo muitas tuplas provavelmente receberá uma elevada quantidade de formigas tentando recuperar e/ou depositar tuplas. Isto é particularmente grave para *template-ants* que naturalmente exigem mais processamento que as *tuple-ants*, já que uma *template-ant* precisa executar um grande volume de comparações para encontrar uma tupla que satisfaça as restrições específicas em seu *template*. E isso é computacionalmente dispendioso.

O *nível magnético* a respeito de uma determinada tupla τ_c que algum nó i está exposto é definido pela Equação (5).

$$M_L(i, \tau_c) = \text{Max}\{C(n, \tau_c) : n \in NH(i)\} \quad (5)$$

O nível magnético para um nó i corresponde à máxima concentração de tuplas que são similares à tupla τ_c nos nós vizinhos de i . M_L é utilizado para determinar a *força magnética* sofrida pelas formigas presentes no nó i , designada por $F_M(i)$ (Equação (6)). Esta força magnética é responsável por produzir um “comportamento estranho” em uma *tuple-ant*, obrigando ela a depositar sua tupla no nó atual, ou seja, antes do esperado.

$$F_M(i, \tau_c) = M_c(i) * \frac{M_L(i, \tau_c)}{M_c(i) + M_L(i, \tau_c)} \quad (6)$$

A força magnética que afeta as *tuple-ants* no nó i tende à zero quando não existe nenhum campo magnético de nível significativo produzido pelos vizinhos do nó

i . $F_M(i, \tau_c)$ depende da constante de *restrição magnética* $M_c(i)$ que representa o número máximo de tuplas que um nó i pode armazenar sem ser considerado “sobrecarregado”.

A probabilidade de um “comportamento estranho” durante a fase de movimentação de uma formiga que está atualmente no nó i carregando uma tupla τ_c é definida pela Equação (7).

$$P_S(i, \tau_c) = \frac{F_M(i, \tau_c) + C(i, \tau_c)}{\sum_{\forall n \in NH(i)} (C(n, \tau_c) + Ph(n, \tau_c)) + F_M(i, \tau_c) + C(i, \tau_c)} \quad (7)$$

O “comportamento estranho” de uma *tuple-ant* consiste na decisão de depositar a tupla carregada τ_c no nó corrente ao invés mover-se para outro nó como é esperado. Um nó i que possui um nó adjacente sobrecarregado, por exemplo, está sobre uma forte interferência magnética (calculada a partir de $F_M(i, \tau_c)$) e, conseqüentemente, uma formiga neste nó possui uma elevada probabilidade de depositar a tupla em i , i.e., antes de chegar ao nó sobrecarregado.

Considerando uma $F_M(i, \tau_c)$ elevada, a influência magnética é potencializada pelo aumento de $C(i, \tau_c)$, que é provocado pela exposição contínua do nó ao campo magnético, aumentando a probabilidade de uma formiga carregando uma tupla τ_c apresentar um comportamento estranho em um nó i . Devido a este comportamento, há uma tendência de formação de *clusters* de nós que contém *clusters* de tuplas similares, ou seja, a sobrecarga de nós é evitada através da formação de *clusters* de *clusters*. Observe, no entanto, que a interferência magnética não restringe completamente o comportamento dos indivíduos, pois a adaptabilidade do sistema é alcançada exatamente pela tomada de decisões estocásticas. Portanto, ainda que exista uma intensa interferência magnética, uma formiga pode apresentar um comportamento normal, embora com uma pequena probabilidade.

4. Avaliação Experimental

Com o intuito de avaliar o desempenho do modelo proposto, o *Magnetic SwarmLinda* foi implementado através de simulação. Nos diversos cenários avaliados, que são apresentados na Tabela 1, o desempenho do modelo proposto é comparado aos desempenhos do *SwarmLinda* Tradicional e do *SwarmLinda* com *Anti-Over-Clustering*. A métrica de interesse para a avaliação de desempenho é o tempo médio para a recuperação de uma tupla através da operação *in*. Esta escolha deve-se ao fato da operação de inserção ser instantânea na perspectiva da aplicação, pois não é necessário que o processo cliente aguarde que a tupla seja depositada.

Uma *unidade de tempo ideal* (UTI), representa o tempo necessário para a entrega de uma mensagem entre dois nós diretamente conectados. Foram utilizadas topologias *Small-World* [Watts and Strogatz 1998], no qual inicialmente é criada uma rede em anel com $N = |V|$ nós, sendo que cada nó é conectado com seus k vizinhos mais próximos (se k é par). Em seguida, cada aresta $(u, v) \in E$ é substituída com probabilidade σ por uma nova aresta (u, w) , onde $w \in V$ é escolhido aleatoriamente. Foram usados os seguintes parâmetros para a geração de gráficos: a probabilidade $\sigma = 30\%$ de reescrever uma aresta do grafo *Small-World*, o número $N = 16$ de nós do sistema e o grau médio $k = 4$.

Em cada nó que compõe o Espaço de Tuplas é conectado um processo cliente que é responsável por produzir carga sobre o Espaço de Tuplas. Cada cliente executa uma

operação a cada I UTI's, alternando entre operações *out* e *in*. Entretanto, no início da simulação, os processos clientes executam apenas operações *out* até que T tuplas tenham sido depositadas, permitindo que uma carga inicial de tuplas seja estabelecida. Cada operação tem um probabilidade uniforme de ser relativa a qualquer um dos τ^t tipos de tuplas e aos seus τ^v valores possíveis. As operações de recuperação são relativas exclusivamente a tuplas disponíveis no Espaço de Tuplas. Outros parâmetros *default*, que foram variados nos cenários dos experimentos realizados incluem: o número de operações $OP = 1000$ que cada nó é capaz de executar durante uma UTI, a duração $D = 1000$ da simulação em termos de número de operações *in*, o número $S = 50$ de simulações executadas em cada cenário e a taxa de evaporação $\rho = 20\%$.

Além disso, o valor da restrição magnética $M_c(i)$ utilizado nos diferentes cenários foi definido como a quantidade “desejável” de tuplas a ser depositada em cada nó. De tal modo, $M_c(i) = T/N, \forall i \in V$, onde T é o número total de tuplas no sistema (independentemente do seu “tipo”) e $N = |V|$ é o número de nós.

Tabela 1. Cenários de Avaliação de Desempenho

Cenário	Parâmetro	Valores	Avaliação de desempenho para
1	I	3, 4, 5, 6 e 7	altas taxas de requisição de operações
2	I	5, 15, 25 e 35	altas e baixas taxas de requisição de op.
3	τ^v	500, 1000, 1500 e 2000	diferentes valores para cada tipo de tupla
4	k	4, 6, 8 e 10	diferentes números médios de arestas
5	T	(1, 2, 3 e 4) * 10^5	diferentes números de tuplas no sistema
6	τ^t	3, 4, 5, 6, 7 e 8	diferentes números de tipos de tuplas

Nos Cenários 1 e 2, o desempenho é avaliado para diferentes intervalos I entre operações, ou seja, distintas frequências de operações. No Cenário 3, o desempenho é avaliado para diferentes níveis de “raridade” de tuplas, uma vez que o número de tuplas idênticas é maior conforme o número τ^v de valores possíveis de tuplas é menor. No Cenário 4, o desempenho é avaliado quando aumenta a densidade do grafo. No Cenário 5, o desempenho é avaliado para diferentes números T de tuplas depositadas no Espaço de Tuplas, sendo que a quantidade permanece constante já que cada operação de inserção é alternada com uma de recuperação. Finalmente, no Cenário 6, o desempenho é avaliado para diferentes números de tipos de tuplas. Isso permite medir o impacto de diferentes quantidades de *clusters* de nós, uma vez que cada *cluster* - no modelo magnético - está associado a um determinado tipo de tupla. O nível de confiança dos resultados é de 95%.

Os resultados do Cenário 1 apresentados na Figura 3(a) mostram os limites para os níveis de carga (frequência com que operações são executadas) a partir do qual o Espaço de Tuplas deixa de funcionar adequadamente. Este ponto de saturação ocorre quando o tempo médio para recuperar uma tupla é superior ao dobro do intervalo I entre operações, pois o tempo entre as inserções - devido à alternância de operações - é a metade do valor de I . Neste momento, o Espaço de Tuplas não consegue absorver a carga, provocando uma degradação cumulativa do desempenho do sistema.

A identificação do ponto de saturação das três abordagens constata que a estratégia proposta suporta níveis superiores de requisições quando comparada às demais abordagens. O ponto de saturação do *Magnetic SwarmLinda* ocorre para $I = 3$. Para as outras

duas abordagens, este ponto ocorre quando $I = 4$. É importante mencionar que o valor de eixo y foram “aparados” em 50 a fim mostrar melhor o comportamento das três abordagens para $I \geq 4$. Além disso, para a abordagem tradicional, quando $I = 4$, o tempo médio para recuperar uma tupla tende ao infinito para simulações mais longas. Isso acontece porque para $I = 4$, os processos clientes estão inserindo tuplas a cada 8 UTIs, enquanto que o tempo médio para recuperar uma tupla é cerca de 20 UTIs.

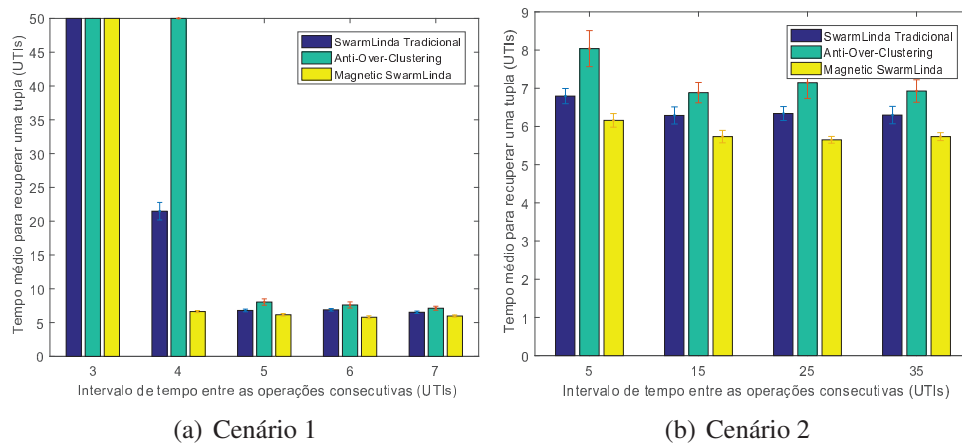


Figura 3. Avaliação de Desempenho dos Cenários 1 e 2

Os resultados do Cenário 2, que são apresentados na Figura 3(b), demonstram que a abordagem proposta apresenta um desempenho superior em momentos nos quais o Espaço de Tuplas está em funcionamento normal, ou seja, não está próximo ao ponto de saturação. Conforme esperado, todas as abordagens têm um desempenho melhor em cenários com baixos níveis de requisições, pois a carga sobre o Espaço de Tuplas é menor.

Os resultados do Cenário 3 na Figura 4(a) demonstram que o modelo proposto apresenta um desempenho melhor quando há uma quantidade menor de possíveis valores por tipo de tupla. Conforme diminui o número de valores possíveis, maior é o número de tuplas idênticas. Desta forma, a quantidade de nós que uma *template-ant* precisa visitar tende a ser menor, uma vez que é maior a probabilidade de existirem tuplas compatíveis em múltiplos nós de um *cluster*. Por outro lado, a abordagem magnética apresenta um desempenho inferior as outras abordagens quando há uma pequena quantidade de tuplas idênticas (ou seja predominam tuplas “raras”) pois a *template-ant* é obrigada a visitar um número maior de nós do *cluster* magnético, gerando muito movimento dentro do *cluster*.

Os resultados do Cenário 4 na Figura 4(b) demonstram que a abordagem magnética possui um melhor desempenho no tempo de resposta para recuperação de tuplas para diferentes topologias. Nas topologias com baixa densidade a probabilidade de uma formiga alcançar qualquer *cluster* com um pequeno número de saltos é alta. Como cada *cluster* magnético é composto de múltiplos nós, é provável que exista pelo menos um caminho curto entre qualquer nó do Espaço de Tuplas e um dos nós de cada *cluster*.

Os resultados do Cenário 5 na Figura 5(a) demonstram que a abordagem proposta possui um desempenho superior às demais abordagens para diferentes quantidades de tuplas depositadas no Espaço de Tuplas. Além disso, o tempo médio para recuperar uma tupla diminui de forma muito expressiva na *Magnetic SwarmLinda* conforme a quantidade de tuplas aumenta. Todas as abordagens apresentam um desempenho melhor para

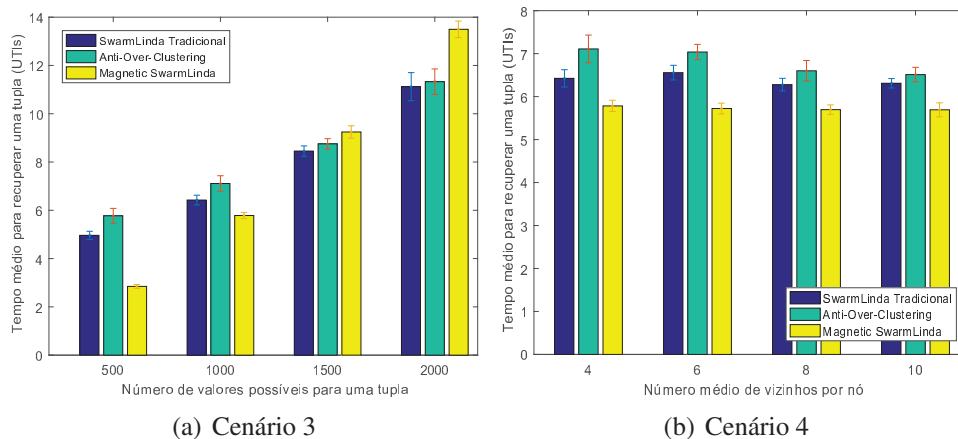


Figura 4. Avaliação de desempenho dos cenários 3 e 4.

grandes quantidades de tuplas, pois há mais tuplas disponíveis no Espaço de Tuplas que são compatíveis com os *templates* das operações de leitura.

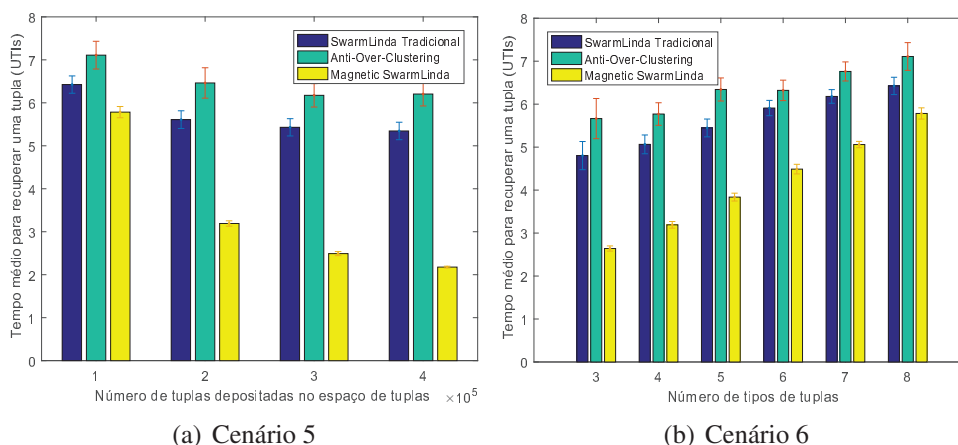


Figura 5. Avaliação de desempenho dos cenários 5 e 6.

Os resultados do Cenário 6 na Figura 5(b) demonstram que a abordagem proposta é capaz de tirar proveito de cenários onde o número de tipos de tuplas τ^t é menor que o número nós $N = |V|$. Isto deve-se ao fato que conforme diminui o número de tuplas, menor é o número de *clusters*. Desta forma, cada *cluster* é formado por um número maior de nós, provocando uma expressiva melhora do desempenho. As outras abordagens se destacam quando há poucos tipos de tuplas, porém a melhora ocorre de maneira menos expressiva.

5. Conclusão

Espaços de Tuplas são um modelo simples e elegante para a construção de sistemas paralelos e distribuídos. Abordagens distribuídas têm sido propostas para aumentar a escalabilidade do sistema. Este trabalho apresentou o *Magnetic SwarmLinda* que combina inteligência de enxames e campos magnéticos virtuais para não apenas garantir a escalabilidade mas também permitir a recuperação eficiente de tuplas. Em comparação com outras abordagens, o *Magnetic SwarmLinda* evita a dispersão excessiva de tuplas semelhantes, formando *clusters* de *clusters* contendo tuplas similares evita a sobrecarga de nós

específicos. O sistema foi implementado e avaliado em diferentes cenários, demonstrando sua superioridade com relação a outras estratégias exceto em cenários com tuplas “raras” – isto é, pequenas quantidades de tuplas similares – que não são típicos na computação de alto desempenho. Os benefícios são comprovados sempre sempre que o número de tuplas é elevado. Em particular, houve ganho significativo de desempenho em cenários onde a quantidade de nós que compõem o Espaço de Tuplas é maior que o número de tipos de tuplas. Além disso, o *Magnetic SwarmLinda* apresentou um comportamento estável independentemente da densidade da rede. Trabalhos futuros incluem definir uma abordagem autônoma para ajustar o valor da restrição magnética ao longo do tempo.

Referências

- Atkinson, A. (2008). Tupleware: A distributed tuple space for cluster computing. In *Parallel and Distributed Computing, Applications and Technologies, 2008. PDCAT 2008. Ninth International Conference on*, pages 121–126. IEEE.
- Banks, A. N. and Srygley, R. B. (2003). Orientation by magnetic field in leaf-cutter ants, *atta colombica* (hymenoptera: Formicidae). *Ethology*, 109(10):835–846.
- Casadei, M., Menezes, R., Tolksdorf, R., and Viroli, M. (2007). On the problem of over-clustering in tuple-based coordination systems. In *Self-Adaptive and Self-Organizing Systems, 2007. SASO'07. First International Conference on*, pages 303–306. IEEE.
- de Paula Lima Jr., L. A. and Calsavara, A. (2010). Autonomic application-level message delivery using virtual magnetic fields. *Journal of Network and Systems Management*, 18:97–116.
- Gelernter, D. and Bernstein, A. J. (1982). Distributed communication via global buffer. In *Proceedings of the first ACM SIGACT-SIGOPS Symposium on Principles of distributed computing*, pages 10–18. ACM.
- Hari, H. (2012). Tuple space in the cloud. Master’s thesis, Uppsala University, Department of Information Technology.
- Jiang, Y., Xue, G., Jia, Z., and You, J. (2006). Dtuples: A distributed hash table based tuple space service for distributed coordination. In *Grid and Cooperative Computing, 2006. GCC 2006. Fifth International Conference*, pages 101–106. IEEE.
- Maia, M. E., Andrade, R., and Viana, W. (2016). Towards a component infrastructure for cyber-physical systems. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 626–628. ACM.
- Menezes, R. and Tolksdorf, R. (2003). A new approach to scalable linda-systems based on swarms. In *Proceedings of the 2003 ACM Symposium on Applied computing*, pages 375–379. ACM.
- Picco, G. P., Murphy, A. L., and Roman, G.-C. (1999). Lime: Linda meets mobility. In *Proceedings of the 21st international conference on Software engineering*, pages 368–377. ACM.
- Rowstron, A. (1998). Wcl: A co-ordination language for geographically distributed agents. *World Wide Web*, 1(3):167–179.
- Wajnberg, E., Acosta-Avalos, D., Alves, O. C., de Oliveira, J. F., Srygley, R. B., and Esquivel, D. M. (2010). Magnetoreception in eusocial insects: an update. *Journal of the Royal Society Interface*, page rsif20090526.
- Waldo, J. et al. (1997). Javaspace specification-revision 0.4. Technical report, Technical report, Sun Microsystems, JavaSoft Lab.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of “small-world” networks. *nature*, 393(6684):440–442.