Consenso em Memória Compartilhada

Cátia Khouri[1], **Fabíola Greve**[2]

[1]Departamento de Ciências Exatas e Tecnológicas – Universidade Estadual do Sudoeste da Bahia (UESB). Aluna do PMCC UFBA/UEFS/UNIFACS

[2]Departamento de Ciência da Computação – Universidade Federal da Bahia (UFBA)

catia091@dcc.ufba.br, fabiola@dcc.ufba.br

Resumo. *O consenso é uma abstração fundamental no desenvolvimento de sistemas distribuídos confiáveis. As redes de área armazenamento e máquinas multinúcleo ExaScale são desafios que se apresentam na atualidade e para um futuro próximo da computação distribuída. O presente trabalho visa estudar o problema do consenso aplicado a estes ambientes no sentido de definir um modelo de sistema adequado para estas abordagens que expresse as condições mínimas necessárias para que o consenso possa ser atingido.*

1. Introdução

Sistemas distribuídos tolerantes a falhas devem continuar a prover serviços a despeito de falhas em seus nós ou canais de comunicação. É fundamental que mesmo com a falha de alguns participantes, os processos que operam corretamente possam colaborar de alguma maneira. Um importante bloco de construção em tais ambientes é o *acordo*. Em muitas situações, os processos livres de falhas precisam concordar com relação a uma determinada informação para manter a integridade do sistema. Dentre os problemas de acordo, o *consenso* [Chandra and Toueg 1996] é o mais importante. Ele pode ser visto como um arcabouço geral de acordo e a maneira mais natural de encapsular esse problema. Informalmente, o problema do consenso pode ser definido como segue. Cada processo propõe um valor (ou conjunto de valores) e todos os processos não falhos têm que decidir por adotar o mesmo valor entre os propostos.

Apesar da simplicidade do enunciado, não existe solução determinística para o consenso em sistemas assíncronos sujeitos a falhas de processos (*crashing*) [Fischer et al. 1985]. Esse resultado tem motivado pesquisadores a definirem um conjunto de propriedades minimais que, quando satisfeitas, permitem a resolução do consenso. O problema é ainda mais complexo quando se considera um modelo de sistema dinâmico com o conjunto de processos participantes desconhecido, como ocorre em sistemas baseados em coleções ad-hoc de dispositivos de computação distribuídos, redes P2P ou computação em nuvem. O foco deste trabalho está em estudar o problema do consenso em ambientes assíncronos, dinâmicos em que processos podem falhar por parada e trocam informações através de uma memória compartilhada.

Nesse intuito, já foram produzidos resultados que incluem a produção de um conjunto de algoritmos para resolução do consenso em ambiente dinâmico com a abstração detector de participantes [Khouri et al. 2012]; e um algoritmo genérico para o consenso que pode ser instanciado com um *detector de falhas forte após um tempo* ou um *detector de líder* [Khouri and Greve 2013]. No atual estágio do estudo, o nosso interesse volta-se

para classes de sistemas como as *redes de área de armazenamento* (SANs) e máquinas multinúcleo, com o intuito de definir as condições minimais para a resolução do consenso nesses ambientes. Para tanto, objetivamos (1) compreender os requisitos de SANs e máquinas multinúcleo, visando, frente à impossibilidade do consenso, (2) definir as suposições/condições mínimas necessárias para a resolução do consenso, considerando, (2.1) modelo de falhas: resiliência, mecanismos de tolerância a falhas; (2.2) conjunto de participantes: quanto ao anonimato de processos (processos devem possuir identidades únicas?), e quanto à cardinalidade (entre execuções; ao longo de uma execução; ou instantaneamente, em uma execução [Aguilera 2004]); (2.3) memória compartilhada (blocos de discos podem ser modelados como registradores? que restrições são necessárias?); e (2.4) sincronismo do sistema (nesse caso, devemos ainda definir qual abordagem utilizar – se baseado em tempo ou em um padrão de escrita na memória, seguindo raciocínio análogo a [Mostefaoui et al. 2006], para sistemas de passagem de mensagens).

2. Redes de Área de Armazenamento (SANs)

Uma rede de área de armazenamento é uma insfraestrutura cujo propósito básico é a transferência de dados entre sistemas de computador e elementos de armazenamento [Tate et al. 2012]. Uma SAN possui um caráter altamente flexível em vários sentidos. Em princípio, pode-se conectar a uma SAN qualquer número de servidores com distintos sistemas operacionais, bem como qualquer número de distintos dispositivos de armazenamento. Ela permite conexões do tipo *any-to-any*, eliminando a tradicional conexão dedicada entre um servidor e uma unidade de armazenamento. É possível mesmo a conexão de *discos ativos em rede* (NAD) permitindo que clientes acessem diretamente os discos, evitando o gargalo do servidor de arquivos [Aguilera et al. 2003].

Como não é necessário qualquer conhecimento prévio sobre o conjunto de processos que participam no sistema, uma SAN é uma candidata natural para sistemas dinâmicos. Uma vez que o conjunto NAD executa requisições de leitura e escrita sobre os blocos de dados, ele pode ser visto como uma memória compartilhada. Cada disco é dividido em blocos, os quais podem ser modelados como registradores compartilhados que são acessados concorrentemente por diversos processos. As vantagens desta estrutura têm motivado o projeto de algoritmos de consenso baseados em disco [Aguilera et al. 2003, Gafni and Lamport 2003], que são capazes de fornecer serviços de armazenamento distribuídos confiáveis.

3. Sistemas Multinúcleo

Outra classe de sistemas de interesse, consiste das máquinas multinúcleo, em especial, aquelas projetadas para um futuro próximo, a saber, *ExaScale*. O termo refere-se a um poder computacional de 10^{18} flops (*floating point operations per second*). Considerando o estado atual da arte, espera-se que máquinas *ExaScale* estejam em funcionamento até o fim desta década [Riesen et al. 2012]. Apesar da dificuldade em se predizer a configuração exata dessas máquinas alguns autores arriscam palpites baseados nas máquinas petaflops atuais e pesquisas em andamento.

Espera-se que o número de núcleos atinja a ordem de 10^9, conectados através de uma rede *Network on Chip* (NoC) [Riesen et al. 2012, Demmel and Nguyen 2013]. Parte da memória principal poderá ser local, isto é, no mesmo rack, enquanto que outra parte

ficará distante, em um servidor de armazenamento dedicado. É possível que algumas transferências de dados ocorram via MPI (*Message Passing Interface*), enquanto outros dados sejam escritos diretamente na memória [Riesen et al. 2012].

4. Considerações

Apesar da importância do consenso, raros são os trabalhos que abordam o problema em ambientes dinâmicos de memória compartilhada. Além disso, dada a impossibilidade do consenso em sistemas assíncronos sujeitos a falhas, é preciso ponderar a respeito do comportamento do sistema a fim de propor uma solução que possa ser, de fato, aplicada. Até onde sabemos, não existem estudos que apresentem uma análise cuidadosa dos modelos de sistema em questão que permitam fazer suposições e chegar às condições mínimas necessárias de sincronismo, resiliência, conectividade, tipos de objetos compartilhados e respetivas operações disponíveis, etc., para se atingir o consenso em tais sistemas.

References

Aguilera, M. K. (2004). A pleasant stroll through the land of infinitely many creatures. *SIGACT News*, 35(2):36–59.

Aguilera, M. K., Englert, B., and Gafni, E. (2003). On using network attached disks as shared memory. In *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, PODC '03, pages 315–324. ACM.

Chandra, T. and Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, 43(2):225–267.

Demmel, J. and Nguyen, H. D. (2013). Numerical reproducibility and accuracy at exascale. In *Computer Arithmetic (ARITH), 2013 21st IEEE Symposium on*, pages 235–237.

Fischer, M. J., Lynch, N. A., and Paterson, M. D. (1985). Impossibility of distributed consensus with one faulty process. *Journal of ACM*, 32(2):374–382.

Gafni, E. and Lamport, L. (2003). Disk paxos. *Distributed Computing*, 16(1):1–20.

Khouri, C. and Greve, F. (2013). A generic consensus algorithm for shared memory. In *The 19th IEEE Pacific Rim International Symposium on Dependable Computing*, PRDC 2013, Vancouver.

Khouri, C., Greve, F., and Tixeuil, S. (2012). Consensus with unknown participants in shared memory. In *32nd International Symposium on Reliable Distributed Systems*, SRDS 2013, Braga.

Mostefaoui, A., Raynal, M., and Travers, C. (2006). Time-free and timer-based assumptions can be combined to obtain eventual leadership. *IEEE Trans. Parallel Distrib. Syst.*, 17(7):656–666.

Riesen, R., Ferreira, K. B., Varela, M., Taufer, M., and Rodrigues, A. (2012). Simulating application resilience at exascale. In Alexander, M. e. a., editor, *Euro-Par 2011: Parallel Processing Workshops*, pages 221–230. Springer Berlin Heidelberg.

Tate, J., Beck, P., Ibarra, H. H., Kumaravel, S., and Miklas, L. (2012). *Introduction to Storage Area Networks and System Networking*. Redbooks. IBM Corp., New York, NY, USA.