Protocolos em Redes de Dados – Crítica 6
João Vicente 44294 Ana Salvado 44299 Inês de Matos 43538

A scalable, Commodity Data Center Network Arquitecture

Hoje em dia, os data centers podem conter milhares de computadores que necessitam de uma quantidade significativa de largura de banda. Porém, o ponto de estrangulamento numa rede de *clusters* em larga escala é a comunicação entre os nodos, na qual não existe o total benefício (usar a largura de banda na sua totalidade) e, ainda assim, implica elevados custos.

Neste artigo os autores tentam tirar vantagem de switches de Ethernet comuns que suportem a banda larga e sugerem que arquiteturas apropriadas e *commodity switches* interligados podem dar mais desempenho, com um menor custo que as soluções que existem hoje em dia.

De acordo com o que já foi dito, a comunicação é o ponto crucial para uma arquitetura ser bem sucedida e já existem duas abordagens, que pelos autores não são suficientes ou não se enquadram no objetivo. Existe, o uso de protocolos como o InfiniBand, que apesar de ser escalável e oferecer uma largura de banda alta, não são apropriados para *commodity switches* e não são compatíveis com aplicações TCP/IP; a outra abordagem parte pela intercomunicação entre *(commodity Ethernet*) *switches* e máquinas do *cluster*, porém a banda larga escala pouco com o tamanho do *cluster* e aumenta o custo por crescimento. Assim, os autores tiveram em consideração três pontos fulcrais para a comunicação: cada host tem de conseguir comunicar com qualquer outro host, usando toda a banda larga que é suposta naquela rede; esta comunicação tem de ser mediada através de *switches* de baixo custo mas suficientes para a tarefa, por fim, a arquitetura tem de ser capaz de se adaptar a qualquer sem ter de alterar o *background*.
A topologia adoptada para esta arquitetura é uma fat-tree de três níveis, pelo que os autores pretendem satisfazer cerca de 10 mil *hosts*; existe um nível onde se encontram os *cores*, outro nível pertence aos *aggregators* que fazem a comunicação entre os *cores* e os *edges*, que por sua vez fazem ligação com os *hosts* (idêntico ao nosso projecto). Cada camada irá ter uma configuração específica das ligações e dos *switches*. Em relação ao enderaçamento dos *switches* é usado o sistema IPV4: 10.*pod*.*switch*.ID, onde *pod* é o número do *pod* correspondente, o *switch* é o número do próprio dentro da subnet desse *pod* e o ID é a posição do *host*. Relativamente ao *routing*, cada *switch* terá uma tabela de dois níveis, ou seja, o primeiro verifica o prefixo do IP (pref.pref.pref.0/0) e reencaminha para a porta respetiva e o segundo corresponde a porta pelo sufixo (0.0.0.suf/suf), dito pelos autores poderá gerar uma pequena latência, no entanto estas tabelas serão pequenas, logo não deve causar atrasos significativos. Estas tabelas foram adaptadas, em hardware, usando o TCAM, para guardar os endereços (prefixos e sufixos) e indexando-os numa outra tabela RAM que guarda o próximo *hop* (e porta) respetivo a cada IP.
Para evitar a congestão de *flows*, que querem usar a mesma porta, é usada uma classificação com redesignação dinâmica das portas com o mesmo custo que a porta original.
Para a tolerância a faltas, o que numa fat-tree é susceptível a acontecer falhas, os autores propuseram um protocolo de *broadcast* para determinar quando é que uma ligação ou um *switch* vizinho está desconectado. Para o caso de serem *switches* que comprometem a ligação a um *host*, são considerados elementos de redundância, para o caso de serem elementos superiores, por exemplos *cores*, é renomeado outro *core* para substituir ou são evitados pedidos que passem por esse elemento.
Através de alguns experimentos, eles conseguiram observar que uma fat-tree necessita de menos potência e tem uma menor dissipação de calor, com uma diferença de mais de 50%, do que uma arquitetura que esteja feita de forma hierárquica.
Algo que entenderam, foi a necessidade de haver imensos cabos de ligação entre as máquinas, no entanto os autores adotaram algumas técnicas de forma a reduzir o custo e a facilitar a manutenção do *cluster*.

Existem outras topologias, tal como o Torus, que dito pelos autores, torna-se impraticável devido à complexidade de fios que são necessários e das funções de *forwarding* que não são propícias a *commodity hosts.*