

Qual é o problema que os autores tentam resolver?

A Azure é o segundo maior fornecedor de serviços cloud mundial e depara-se com problemas complexos, principalmente à medida que a tecnologia informática aumenta. Um destes problemas é o custo da receção de pacotes nas máquinas virtuais (VM) a correr nas máquinas host (HM). Para uma VM receber um pacote, é necessário que a HM receba o pacote na placa de rede, processe o pacote, perceba que é para a VM e finalmente “envie” o pacote para VM. Este processo é lento devido à execução processos.

À medida que é que possível uma maior velocidade da rede (2009 - 1GbE, 2015 - 40GbE), é necessário que a capacidade de processamento de pacotes da rede aumente também. No entanto, esse processamento era feito através do uso de CPU e, infelizmente, estes não sofreram um aumento em desempenho tão grande como a velocidade da rede.

Para combater este problema, é necessário desenvolver novas soluções, usando novo software ou hardware especializado, de maneira a que a solução seja viável futuramente quando a velocidade da rede aumentar ainda mais (e.g. 400 GbE), por outras palavras, a **solução tem de ser pronta para suportar novos avanços da rede.**

Este problema é relevante?

É cada vez mais comum o uso da cloud para fazer host de serviços dos clientes ou armazenamento de dados e, à medida que é possível e desejada uma maior velocidade da rede, é **imperativo** que o acesso às máquinas que fornecem o serviço de cloud **não sofram bottleneck** por nenhuma razão, e que consiga acompanhar o rápido crescimento da velocidade da rede.

Qual é a sua solução? Que novas técnicas foram usadas?

Para isto foi criado o Azure Accelerated Networking (AccelNet), uma solução que descarrega o processamento da rede da HM para hardware usando um Azure SmartNICs (uma interface para redes para intercomunicação entre VMs) próprio que se baseia em FPGAs (Field Programmable Gate Arrays).

Inicialmente, foram consideradas 4 opções de hardware para resolver o problema: continuar a usar CPU, usar SoCs (que se comportam similar a um GPU), usar FPGAs ou usar ASICs. A primeira opção é exatamente o problema que querem resolver, daí ter sido eliminada. A segunda opção tinha uma maior performance comparando com apenas o uso de CPUs, mas iria-se tornar demasiado caro em termos de energia e quantidade necessária quando a rede comesse a ter velocidades de 200GbE para cima. Notaram ainda algum aumento de latência e problemas quando se queria manter estado (devido à maneira de processamento dos GPUs - threads).

Usar ASICs parecia ser uma solução bastante viável quando se olhava para o desempenho, visto que são chips especializados e otimizados para a função que são desenhados. No entanto, estes chips demoram demasiado tempo a ser feitos (1-2 anos), o que faz com que os chips tinham de suportar o que a rede seria nos próximos 7 anos (2 a serem produzidas e 5 de tempo médio de vida útil dos servidores).

Fica, por fim, a solução que utiliza os FPGAs. Os FPGAs permitem ter quase todas as vantagens de ASICs, perdendo alguma performance, mas são programáveis permitindo a adição de novas funcionalidades quando necessário sem ter de esperar por entidades externas. Está

cientificamente provado que os FPGA são muito mais rápidos do que o software permitindo resolver o problema proposto.

No entanto, os FPGAs não são perfeitos e dão origem a várias dúvidas sobre a sua viabilidade e custo. Um dos problemas teóricos dos FPGAs seria o seu tamanho tendo em conta que a lógica do componente ocupa cerca de 10 a 20 vezes mais espaço que nos ASIC. Apesar disto, devido ao espaço ocupado por outros componentes, a diferença de tamanho é apenas de 2 a 3 vezes. Outro problema proposto seria o preço; no entanto, apesar de serem mais caros do que os ASICs, valem a diferença devido à sua maior longevidade. Existe ainda algum cepticismo em relação à dificuldade de programação dos FPGA; no entanto, apesar de estes não serem fáceis de programar para um programador de software comum, os especialistas em hardware têm as capacidades necessárias para fazer a programação destes chips. Outro problema parecia ser a ligação demasiado forte ao fabricante do chip que estamos a programar, porém já foi provado que é fácil compilar o código feito para outros fabricantes, nomeadamente devido à utilização de linguagens de descrição de hardware como o SystemVerilog. Tal como no sistema original, o processamento do *control plane* continua a ser exatamente igual; em contraste, o *data plane* é agora feito no FPGA.

Como é que se destaca de trabalhos anteriores?

Até à data do artigo, o controlo e processamento do tráfego da rede eram feitos através de software, o que, para ligações com baixa largura de banda, é facilmente factível, mas à medida que esta aumenta, torna-se cada vez mais complicado realizar estas operações.

Em vez de criarem software mais rápido, decidiram atacar o foco do problema (hardware) e arranjar uma solução de longo prazo, em vez de ser para curto prazo. Esta abordagem ao problema foi o que permitiu o sucesso que foi demonstrado em testes comparativos com a concorrência e que mostram uma enorme vantagem ganha com esta mudança.

Quais são os pontos mais fortes deste artigo? E os seus pontos fracos?

Apesar da leitura deste artigo ser demasiado difícil de acompanhar, devido a demasiadas siglas e termos a que não estamos acostumados, conseguimos ver pelos resultados apresentados que a solução consegue obter um alto desempenho, poupando os CPUs e melhorando o desempenho de *single connection*.

A explicação das escolhas tomadas (tais como porquê o uso de FPGAs e não de ASICs), as lições aprendidas (tais como unificar as equipas de hardware e software para melhor comunicação) e os resultados das comparações com outros serviços são fáceis de interpretar mas o desenvolvimento do sistema é complexo, talvez pela nossa falta de conhecimento nesta especificação da área.

Como seria uma extensão deste trabalho?

Tal como mencionado no artigo, uma extensão deste trabalho é a explicação da integração de novas funcionalidades que podem ser acrescentadas às NICs programáveis em todos os hosts.