

RAID 6 Hardware Acceleration.

Gilroy, M.; Irvine, J.; Atkinson, R. RAID 6 Hardware Acceleration. ACM TECS, vol.10, no. 4, artigo 43. Novembro, 2011.

Aluno: 123789 – Gerson Nunho Carriel

O artigo apresenta um trabalho sobre RAID 6 e uso do algoritmo de Reed-Salomon. Em seu início apresenta a definição de Patterson de que o uso de RAID garante mais confiabilidade ao armazenamento de dados, devido ao uso combinado de unidades de disco, oferecendo melhor performance e tolerância a falhas. Outro ponto importante para justificar o trabalho é a probabilidade de mais um disco apresentar defeito cresce à medida que existem mais discos.

Os conceitos sobre o RAID 5 são apresentados para indicar que o RAID 6 trabalha sobre os mesmos conceitos, em que grava blocos de dados em faixa transversalmente nos discos e grava também os blocos de paridade em todo o array.

O RAID 6, objeto de estudo do artigo, oferece suporte para a recuperação de problemas em dois discos. Para isso, é necessário dois discos de checksum, que são aplicados ao conceito de ter as faixas usadas no RAID 5, que também é utilizado para garantir performance. Os autores deixam claro que o RAID 6 não está padronizado, embora exista uso Comercial.

Para compreensão do trabalho, é importante um entendimento mínimo do algoritmo utilizado: o Reed-Salomon(RS).

O RS é uma classe de codificação de correção de erros que permite a recuperação de qualquer n erros de um sistema com $m+n$ entradas de dados. Este algoritmo possui uma capacidade de correção ótima à medida que precisa de n locais de armazenamento para recuperar n falhas. O controle da paridade feito pela codificação de RS e que são necessários dois discos redundantes e duas gravações por faixa, nesse algoritmo, é utilizada a Teoria de Campo de Galois, em que um campo de Galois q é denotado por $GF(q)$.

No artigo, são mostrados os trabalhos de desenvolvimento para RAID 6. A primeira descrita é a implementação do Controlador RAID 6 de quatro discos que foi projetado para utilizar um mínimo de recursos de hardware. O codec do hardware foi implementado com RS codec, operando em $GF(4)$ e considerando as seguintes regras: os discos de dados e de checksum estão em locais fixos, ambos os dados de entrada passam pelo codec simultaneamente e 2 bits por entrada são codificados ou decodificados simultaneamente. A primeira codificação é feita por um XOR de dois dados de um bit e o segundo valor é feito por meio de uma multiplicação por um valor fixo do GF. Caso um ou os dois discos de checagem sejam apagados ou tenham perda de dados, eles podem ser regenerados, passando novamente pelo codificador Reed-Salomon.

Nesse primeiro momento, foi colocar uma interface PCI ao projeto, além disso, uma memória foi adicionada como buffer antes e depois da codificação e decodificação.

A arquitetura com PCI foi analisada quanto aos benefícios de performance. O PCI foi implementado com um Kernel linux 6.2 modificado para atender a demanda. Todas as comparações de performance foram feitas em relação a outro kernel 6.2 não alterado. Para o teste de throughput foi utilizado o Bonnie++, para evitar o cache foi usado um arquivo de 4G.

Foi realizado o teste de Confiabilidade para geração e Recuperação do Array. Para este teste, foi utilizado o mdadm do linux. A verificação da integridade dos dados foi feita comparando um arquivo escrito com o original e checando diferenças. Testes exaustivos foram realizados e os dados do array foram reconstruídos com sucesso.

A performance de Leitura /Gravação, tanto do hardware quanto do software de RAID 6, foram testados com Bonnie+. E ocorreu uma melhora clara quando usado o acelerador de Hardware em comparação ao uso apenas da solução de software no teste com falha dupla de discos.

O Acelerador de Hardware RAID 6 produzia taxa de throughput, próximo ao encontrado a solução de software sob operação normal, oferecendo um ligeiro ganho na falha de um disco e grande benefício durante problemas com dois disco. O uso de CPU ficou entre 3% e 4% pelo Driver do Hardware, enquanto o software utilizava acima de 33% durante operação normal e acima de 50% durante falha em dois discos.

Os autores trabalharam também com uma implementação de 15 discos em $GF(16)$, utilizando entradas de 4 bits de largura e para arrays de até 255 drives foi utilizado $GF(256)$ com 8 bits de comprimento.

Os aceleradores baseados em $GF(16)$ e $GF(256)$ foram feitos para serem mais flexíveis. O Acelerador RAID 6 Baseado em $GF(16)$ conseguiu um consumo de 20% de CPU em relação a outro projeto já publicado. Os testes foram realizados com cinco discos, que foram particionados para atender o projeto.

Para conseguir verificações rápidas dos codec $GF(16)$ e $GF(256)$, um barramento Avalon foi utilizado. Essa arquitetura permitiu gerar métricas de performance.

Ao final os autores acreditam que o desenvolvimento dos três controladores RAID 6 baseados em Reed-Salomon resultaram em solução ótima na primeira implementação, e que a segunda, para 15 discos mostrou que conseguiu realizar a tarefa, em menor taxa de clock, usando apenas 20% dos recursos fornecidos em relação a uma melhor implementação de outro projeto. O terceiro projeto mostrou suporte para os controladores Reed-Salomon RAID, na qual a escalabilidade do codec de $GF(256)$ tem se mostrado ser opção ideal dos fabricantes. Além disso, apresentam que da investigação feita foi demonstrado que reduções nos recursos de hardware podem ser feitas com adoção adequada de algoritmos direcionados a aplicações específicas e os testes mostraram que o Throughput foi mantido para os três projetos.