

Uso de Algoritmos e Estruturas de Dados com Memória Flash

A memória flash é um tipo de memória somente leitura programável apagável eletricamente, é não volátil, por isso é usada para armazenar arquivos e outros objetos persistentes em estações de trabalho e servidores, em computadores de mão e telefones celulares, em câmeras digitais e em tocadores de música portáteis. Os comportamentos de leitura / gravação / exclusão da memória flash são radicalmente diferentes dos de outras memórias programáveis, como ram volátil e discos magnéticos.

Uma abordagem para usar a memória flash é tratá-la como um dispositivo de bloco que permite que blocos de dados de tamanho fixo sejam lidos e gravados, como setores de disco. O mapeamento dos blocos em endereços flash de maneira linear simples apresenta dois problemas. Primeiro, alguns blocos de dados podem ser gravados para muito mais que outros. Isso não apresenta problemas para discos magnéticos; portanto, os sistemas de arquivos convencionais não tentam evitar tais situações. O segundo problema que o mapeamento de identidade coloca é a incapacidade de gravar blocos de dados menores que uma unidade de apagamento instantâneo. A idéia básica por trás de todas as técnicas de nível de desgaste é mapear o número de bloco apresentado pelo host, chamado número de bloco virtual, para um endereço de flash físico chamado setor. Normalmente, os setores têm um tamanho fixo e ocupam uma fração de uma unidade de apagamento. Fundamentalmente, existem dois tipos de estruturas de dados que representam esses mapeamentos. Os mapas diretos são essencialmente matrizes que armazenam no i -ésimo local o índice do setor que atualmente contém o bloco i . Os mapas inversos armazenam no i -ésimo local a identidade do bloco armazenado no i -ésimo setor. Ou seja, mapas diretos permitem o mapeamento eficiente de blocos para setores e mapas inversos permitem mapeamento eficiente de setores para blocos.

A abordagem de mapeamento de blocos tem várias vantagens sobre os sistemas de arquivos específicos do flash. Primeiro, que permite que os desenvolvedores utilizem implementações de sistemas de arquivos existentes, reduzindo assim os custos de desenvolvimento e teste. Segundo, dispositivos flash removíveis, como CompactFlash e SmartMedia, devem usar formatos de armazenamento que sejam entendidos por todas as plataformas nas quais os usuários precisam usá-los. Os sistemas de arquivos convencionais modificam as informações no local. Quando um bloco de dados, que contém parte de um arquivo ou metadados do sistema de arquivos, deve ser modificado, o novo conteúdo substitui os dados antigos exatamente nos mesmos locais físicos do disco. A substituição facilita a localização de dados: a mesma estrutura de dados que permitiu ao sistema encontrar os dados antigos agora encontrará os novos dados.

Alguns aplicativos mantêm estruturas de dados sofisticadas no armazenamento não volátil. As árvores de pesquisa, que permitem que os sistemas de gerenciamento de banco de dados e outros aplicativos respondam rapidamente às consultas, são as mais comuns dessas estruturas de dados. Normalmente, essa estrutura de dados é armazenada em um arquivo. Em aplicativos mais exigentes, a estrutura de dados pode ser armazenada diretamente em um dispositivo de bloco, como uma partição de disco, sem um sistema de arquivos. Alguns autores argumentam que, ao implementar estruturas de dados específicas de aplicativos com reconhecimento de flash, o desempenho e a resistência podem ser aprimorados com a implementação em um sistema de arquivos ou mesmo em um dispositivo de bloco. O principal problema das estruturas de dados específicas do aplicativo com reconhecimento de flash é que elas exigem que o dispositivo flash seja particionado.

Em geral, os desafios dos dispositivos flash mais recentes são maiores do que os dos dispositivos mais antigos - os dispositivos estão se tornando mais difíceis de usar. Isso acontece porque a tecnologia de hardware flash é impulsionada principalmente pelo desejo de maior capacidade e desempenho, geralmente à custa da facilidade de uso. Essa tendência requer o desenvolvimento de novas técnicas de software e novas arquiteturas de sistema para novos tipos de dispositivos.