

Resumo Artigo

The RISC Penalty – Tom Pittman

Nome: Gabriel Felipe

O artigo "The RISC Penalty" de Tom Pittman foi elaborado em um cenário de 1995 mas é muito relevante até para o contexto atual, que não mudou tanto assim. Inicialmente faz-se referência aos antigos modelos de computador CISC, em detrimento aos novos modelos RISC. A diferença real entre essas duas arquiteturas computacionais é pequena, entretanto RISC utiliza uma metodologia de: instruções menores e mais simples, diferentes formatos de instruções e modos de endereços, as instruções tem o mesmo tamanho e executam em um único ciclo, mais registradores e a arquitetura é baseada em registradores ao invés de memória, além de possuir a execução padrão do pipeline.

Para se ter uma ideia, Pittman faz uma comparação de uma máquina antecessora e PowerPC (que possui arquitetura RISC), sendo que a com RISC possui mais do dobro de instruções. Além disso antes existia 16 bits na palavra de instrução, enquanto que na máquina com RISC havia o padrão 32 bits. Afim de aprimorar mais a estrutura RISC, Motorola se esforçou para verificar instruções utilizadas mais vezes e fez essas executarem em um único ciclo.

As máquinas anteriores possuíam oito registradores de dados de 32 bits e oito registradores de endereços de 32 bits, quando vieram as máquinas sucessoras, houve a soma de oito registradores de ponto flutuante 80 e após isso foi estabelecido trinta e dois registradores gerais com 32 bits e trinta e dois registradores de ponto flutuante com 64 bits, aproximando-se da direção da arquitetura RISC.

Com RISC só é possível operações do tipo registrador-registrador, excluindo a memória. O que é feito é que instruções mais simples executam mais rapidamente em um único ciclo de clock, ainda existindo estruturas internas para melhorar a velocidade do modelo. As instruções serem do mesmo tamanho faz com que mais transistores restem para outras funcionalidades e seja mais fácil decodificar e utilizar as instruções que chegam.

A penalidade do uso do RISC que é citada no texto é quando ocorre um miss na cache, o que seria referente a um tempo de mais 3 ciclos por instrução com miss. Existem algumas arquiteturas com recompiladores que dão vantagens de 3 a 5 vezes em velocidade, existindo o caso de miss, os recompiladores com vantagens de 3 vezes não fariam diferença ao usuário, apenas os de 5 vezes que ainda assim não seriam tão efetivos, visto que existiria a penalidade de 3 ciclos eles seriam apenas $5/3 = 1.7$ vezes mais eficientes.

Entretanto, percebeu-se uma coisa: para o caso de softwares que utilizavam uma implementação com vários loops o RISC é recomendado, seria uma situação em que a arquitetura faz-se eficiente, na época o Photoshop era desse tipo e demonstrava bom desempenho com a

arquitetura. Contudo, o Word e Excel da Microsoft fazem apenas um grande loop em sua execução, não tendo uma boa performance em máquinas RISC. Assim programas desse tipo pagariam essa penalidade RISC.

Na época, várias empresas trabalharam para adequar os softwares e até mesmo as máquinas à arquitetura RISC, uma dessas empresas foi a Macintosh, que demorou bastante para adquirir a vantagem de 3 a 5 vezes em velocidade prometida pela arquitetura.

De tanto aprendizado adquirido ao tentar adequar RISC, os programadores da Macintosh fizeram um conjunto de passos para atingir a performance desejada: Análise da performance, focando a atenção e otimização do programa nos pontos fortes da arquitetura, melhorar algoritmos utilizados, modificar localização da memória cache para o local mais adequado, alinhar dados que serão utilizados e transformados, reduzir grandes operações da máquina por equivalentes menores (reduzindo multiplicações por shifts por exemplo), verificar as variáveis nos registradores dando a tipagem específica para tal e optar por pontos flutuantes quando necessários, que são rapidamente implementados na arquitetura RISC e além de tudo isso tentar fazer a utilização da execução em paralelo para diminuir o tempo, como prevê o pipeline.

Dessa forma as arquiteturas evoluíram gradualmente à arquiteturas mais padronizadas e organizadas (caso do RISC), entretanto os novos softwares que fazem essa escolha devem adequar-se para sofrer essa penalidade, procurando tratar todas as questões acima citadas de uma forma diferente para reduzir o tempo consumido pela máquina.