Processadores Fail-Stop

Raul Ceretta Nunes

Definição do Problema

- Sistemas TF devem:
  - ser programados “tendo em mente” que falhas podem causar defeitos.
  - evitar que falhas causem defeitos
  - considerar algumas hipóteses sobre o comportamento do sistema sob falhas
- Para reduzir complexidade:
  - A maioria dos STF assumem que processadores falham parando completamente suas atividades
- Problema:
  - Quando um processador falha, nada pode ser dito sobre seu estado interno (registradores e memória)

Processadores Fail-Stop

- Idéia básica é implementar um processador que ao falhar:

FS1. pare de executar;

FS2. o estado interno e o conteúdo do armazenamento volátil conectado ao processador seja perdido (o estado do armazenamento estável não é afetado); e

FS3. qualquer outro processador possa detectar o seu defeito.

- Dada a impossibilidade de construção, a solução aproximada é a implementação de um processador *k-fail-stop* com o uso de *k+1* processadores ordinários

Aspectos de Implementação

- Separação do estado interno do armazenamento estável (externo)
- Para ser estável armazenamento precisa ser controlado por outro processador (controlador)
- Modelo computacional:
  - p-processador => executa programas usr
  - s-processador => provê estado estável
- Se s-processador funciona é confiável, implementação do processador *k-fail-stop* é simplificada

Implementação com Armazenamento Estável Confiável

- k+1 p-processadores e 1 s-processador
  - p-processadores podem falhar de maneira arbitrária
  - s-processador não falha (confiável)
- processadores se comunicam por troca de mensagens
- rede de comunicação confiável
- mensagens são assinadas (emissor identificável)
- relógios são sincronizados e δ é o atraso de comunicação máximo

Algoritmo para processador k-fail-stop com armazenamento estável confiável

1. Para escrever no armazenamento estável, um p-processo p_j envia uma mensagem de escrita ao s-processador

2. Para ler do armazenamento estável, um p-processo p_j envia uma mensagem de leitura ao s-processador e usa o valor recebido

3. Um s-processo s_i, ao receber uma requisição de todos p-processos:

```
M = lote de requisições recebidas com timestamps apropriados
if |M| = k + 1 and
        todas as requisições são idênticas and
        todas as requisições são de diferentes processos and
        ¬failed then
                if escrita then
                        escreva o valor
                if leitura then
                        envia valor para todos os p-processos cuja requisição está em M
else
        failed = true
```

Comentários sobre o Algoritmo

- Só opera o armazenamento estável se processador k-fail-stop não estiver falho
- No caso de todos falharem de maneira maliciosa (conluio) o armazenamento estável pode escrever um valor errado, mas para tal TODOS os p-processos precisam tentar escrever o mesmo valor malicioso.

Implementação sem Armazenamento Estável Confiável

- Definição do problema:
  - s-processo não é confiável
  - logo deve-se implementar um s-processo confiável, o que implica em múltiplos s-processadores
  - p-processos e s-processos podem falar de maneira arbitrária
  - no mínimo:
    - k+1 p-processadores
    - 2k+1 s-processadores (se mensagens assinadas)
    - 3k+1 s-processadores (se mensagens não assinadas)

Algoritmo para processador k-fail-stop sem armazenamento estável confiável

1. Para escrever no armazenamento estável, um p-processo p_j inicia um acordo bizantino com todos os s-processos

2. Para ler do armazenamento estável, um p-processo p_j:
   (a) broadcast a requisição para os s-processos
   (b) usa o valor maioria de no mínimo k+1 s-processos

3. Um s-processo s_i, ao receber uma requisição de todos p-processos:

```
M = lote de requisições recebidas com timestamps apropriados
if leitura then
    envia valor requisitado para todos os p-processos cuja requisição está em M
if escrita then
    if |M| = k + 1 and
       todas as requisições são idênticas and
       todas as requisições são de diferentes processos and
       ¬failed then
            escreva o valor
    else
            failed = true
            envia mensagens HALT para todos os p-processos
```

Conclusões

- Defeitos num processador k-fail-stop são detectados pelos processos do armazenamento estável (s-processos)
- O defeito é detectado durante tentativa de escrita
- Atualização só são realizadas se processador k-fail-stop é considerado não falho
- Após ser considerado falho, nenhuma atualização é realizada
- Alguma hipótese sobre sincronismo do sistema é necessária (por exemplo, tempo máximo de transmissão)
- Sem falhas arbitrárias e s-processor confiável (livre de falhas)
  - k+1 p-processor e 1 s-processor
- Com falhas arbitrárias
  - Com msgs assinadas => k+1 p-processor e 2k+1 s-processor
  - Sem msgs assinadas => k+1 p-processor e 3k+1 s-processor