

Paxos

Ricardo Anido

29 de Maio de 2014

O problema

Condições

Modelo

Algoritmo básico

Exemplo

Progresso

O problema

- ▶ Consenso com falhas (não bizantinas)

O problema

- ▶ Consenso com falhas (não bizantinas)
- ▶ Garantia de progresso (sem bloquear, como ocorre com o commit de duas fases)

Condições

► Consistência

Condições

- ▶ Consistência
 - ▶ Somente um valor proposto pode ser escolhido

Condições

- ▶ Consistência
 - ▶ Somente um valor proposto pode ser escolhido
 - ▶ Um único valor é escolhido

Condições

- ▶ Consistência
 - ▶ Somente um valor proposto pode ser escolhido
 - ▶ Um único valor é escolhido
 - ▶ Nenhum valor é divulgado antes de ser escolhido

Condições

► Progresso

Condições

- ▶ Progresso
 - ▶ Havendo suficientes processos não-falhos, algum valor será escolhido

Condições

- ▶ Progresso
 - ▶ Havendo suficientes processos não-falhos, algum valor será escolhido
 - ▶ Se um valor é escolhido, então ele é divulgado

Premissas

► Processos

Premissas

- ▶ Processos
 - ▶ Podem parar e reiniciar (necessitam memória persistente)

Premissas

- ▶ Processos
 - ▶ Podem parar e reiniciar (necessitam memória persistente)
 - ▶ Velocidade variável

Premissas

- ▶ Processos
 - ▶ Podem parar e reiniciar (necessitam memória persistente)
 - ▶ Velocidade variável
- ▶ Mensagens

Premissas

- ▶ Processos
 - ▶ Podem parar e reiniciar (necessitam memória persistente)
 - ▶ Velocidade variável
- ▶ Mensagens
 - ▶ Demora variável na entrega

Premissas

- ▶ Processos
 - ▶ Podem parar e reiniciar (necessitam memória persistente)
 - ▶ Velocidade variável
- ▶ Mensagens
 - ▶ Demora variável na entrega
 - ▶ Duplidade

Premissas

- ▶ Processos
 - ▶ Podem parar e reiniciar (necessitam memória persistente)
 - ▶ Velocidade variável
- ▶ Mensagens
 - ▶ Demora variável na entrega
 - ▶ Duplidade
 - ▶ Perda

Premissas

- ▶ Processos
 - ▶ Podem parar e reiniciar (necessitam memória persistente)
 - ▶ Velocidade variável
- ▶ Mensagens
 - ▶ Demora variável na entrega
 - ▶ Duplidade
 - ▶ Perda
 - ▶ Não há falha bizantina

Agentes

- ▶ Client
 - ▶ Quer realizar uma ação

Agentes

- ▶ Client
 - ▶ Quer realizar uma ação
- ▶ Proposer
 - ▶ Inicia o consenso sobre a ação

Agentes

- ▶ Client
 - ▶ Quer realizar uma ação
- ▶ Proposer
 - ▶ Inicia o consenso sobre a ação
- ▶ Acceptor
 - ▶ Aceita ou não uma proposta

Agentes

- ▶ Client
 - ▶ Quer realizar uma ação
- ▶ Proposer
 - ▶ Inicia o consenso sobre a ação
- ▶ Acceptor
 - ▶ Aceita ou não uma proposta
- ▶ Learner
 - ▶ Aprende o resultado do consenso sobre a ação e responde ao cliente

Quóruns

- ▶ Mecanismo para garantir que ao menos um processo não falho tenha conhecimento do resultado do consenso

Quóruns

- ▶ Mecanismo para garantir que ao menos um processo não falho tenha conhecimento do resultado do consenso
- ▶ Subconjunto de Acceptors; qualquer par de quoruns tem interseção não nula (tipicamente, usa-se maioria dos Acceptors)

Algoritmo básico - Fase 1

- ▶ Fase 1a: Preparar
 - ▶ Proposer seleciona número de proposta N e envia mensagem *Prepare* para um quorum de Acceptors.

Algoritmo básico - Fase 1

- ▶ Fase 1a: Preparar
 - ▶ Proposer seleciona número de proposta N e envia mensagem *Prepare* para um quorum de Acceptors.
- ▶ Fase 1b: Promise
 - ▶ Se Acceptor não prometeu não aceitar N , Acceptor promete não aceitar proposta menor do que N e envia o valor da proposta de maior número que ele aceitou para esta instância (se houver) para o Proposer. Senão, envia NACK.

Algoritmo básico - Fase 2

- ▶ Fase 2a: Accept
 - ▶ Se Proposer recebe respostas positivas de um quorum, para uma mensagem *prepare* numerada N , então escolhe um valor para consenso: o valor da proposta de maior número recebida como resposta, se houver; senão, qualquer valor. Proposer envia mensagem *Accept* para um quorum de acceptors, com o valor escolhido.

Algoritmo básico - Fase 2

- ▶ Fase 2a: Accept
 - ▶ Se Proposer recebe respostas positivas de um quorum, para uma mensagem *prepare* numerada N , então escolhe um valor para consenso: o valor da proposta de maior número recebida como resposta, se houver; senão, qualquer valor. Proposer envia mensagem *Accept* para um quorum de acceptors, com o valor escolhido.
- ▶ Fase 2b: Accepted
 - ▶ Se Acceptor recebe mensagem *Accept* para uma proposta que não prometeu não aceitar, aceita o valor. Cada Acceptor envia uma mensagem *Accepted* para o Proposer e para cada Learner.

Algoritmo básico - Definição de proposta “Escolhida”

Um valor é *escolhido* na proposta numerada N se e somente se uma maioria de acceptors aceita esse valor na Fase 2b.

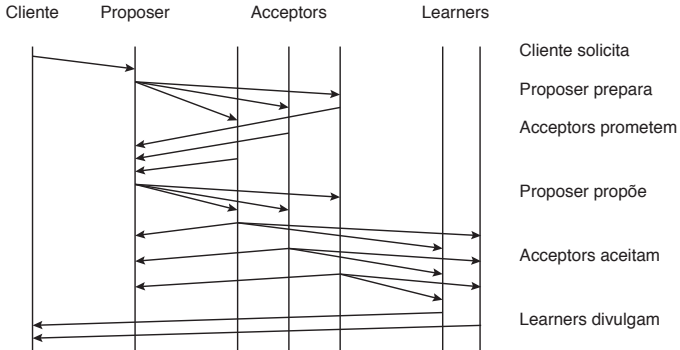
Propriedades do algoritmo Paxos

- ▶ P1: propostas têm números distintos.
- ▶ P2: cada par de quoruns de acceptors tem pelo menos um acceptor em comum.

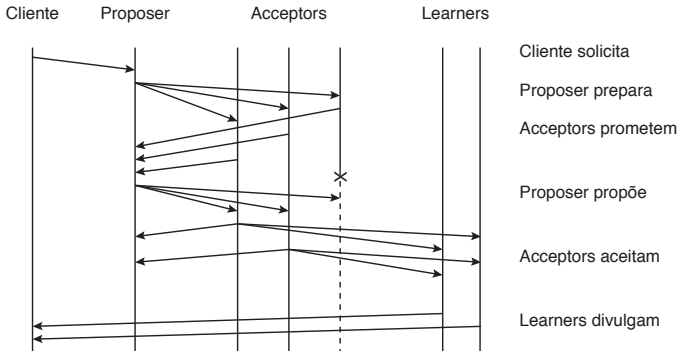
Propriedades do algoritmo Paxos

- ▶ P1: propostas têm números distintos.
- ▶ P2: cada par de quoruns de acceptors tem pelo menos um acceptor em comum.
- ▶ P3: o valor enviado na Fase 2a é o valor da proposta numerada com o maior N , entre todas as respostas da fase 1.

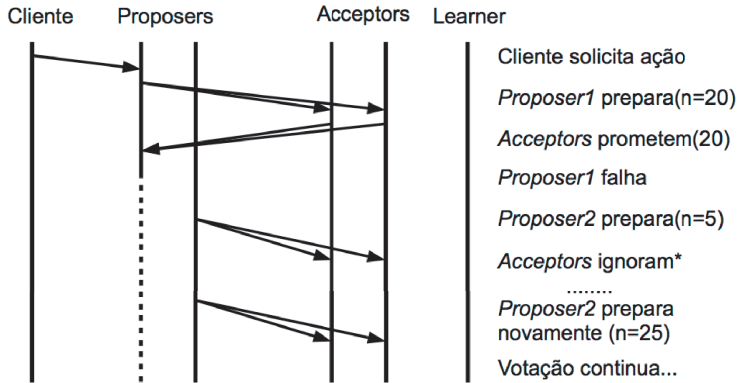
Exemplo



Exemplo



Exemplo



Garantia de progresso

Protocolo básico não garante progresso nos casos:

- ▶ Todos os Proposers falham

Garantia de progresso

Protocolo básico não garante progresso nos casos:

- ▶ Todos os Proposers falham
- ▶ Acceptors ativos não constituem quorum

Garantia de progresso

Protocolo básico não garante progresso nos casos:

- ▶ Todos os Proposers falham
- ▶ Acceptors ativos não constituem quorum
- ▶ Falhas nas entregas de mensagens são “infinitas”

Garantia de progresso

Protocolo básico não garante progresso nos casos:

- ▶ Todos os Proposers falham
- ▶ Acceptors ativos não constituem quorum
- ▶ Falhas nas entregas de mensagens são “infinitas”
- ▶ Disputa entre proposers

Garantia de progresso

Protocolo básico não garante progresso nos casos:

- ▶ Todos os Proposers falham
- ▶ Acceptors ativos não constituem quorum
- ▶ Falhas nas entregas de mensagens são “infinitas”
- ▶ Disputa entre proposers
- ▶ Não há limite para latência de respostas

Garantia de progresso

Condições para garantir progresso:

- ▶ Um único Proposer (eleição de líder antes do protocolo)

Garantia de progresso

Condições para garantir progresso:

- ▶ Um único Proposer (eleição de líder antes do protocolo)
- ▶ Pelo menos um quorum não falho (dimensionar o conjunto)

Garantia de progresso

Condições para garantir progresso:

- ▶ Um único Proposer (eleição de líder antes do protocolo)
- ▶ Pelo menos um quorum não falho (dimensionar o conjunto)
- ▶ Pelo menos um learner não falho (dimensionar o conjunto)

Garantia de progresso

Condições para garantir progresso:

- ▶ Um único Proposer (eleição de líder antes do protocolo)
- ▶ Pelo menos um quorum não falho (dimensionar o conjunto)
- ▶ Pelo menos um learner não falho (dimensionar o conjunto)
- ▶ Acceptors respondem tão logo estejam prontos