Seleção de Coordenadores em Sistemas Transaccionais Geo-Distribuídos

Inês Cardeira, Rafael Soares, and Luís Rodrigues

INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
{ines.cardeira,joao.rafael.pinto.soares,ler}@tecnico.ulisboa.pt

Resumo A replicação parcial permite oferecer capacidade de escala desde que possa ser oferecida usando protocolos de ordenação genuínos, isto é, nos quais apenas os nós envolvidos numa transacção necessitam de participar. O algoritmo de ordenação de Skeen[6] baseado em coordenador é genuíno e usa um número linear de mensagens, mas o seu desempenho é afetado pela localização do coordenador. Neste trabalho avaliamos as vantagens selecionar de forma informada o conjunto de coordenadores em sistemas transaccionais geo-distribuídos. Apresentamos uma avaliação experimental destas técnicas, para aferir as suas vantagens em relação a algoritmos não informados.

Palavras-chave: Transacções Distribuídas, Replicação Parcial, Registos Distribuídos

1 Introdução

Muitos sistemas de armazenamento modernos são simultaneamente geo-distribuídos e parcialmente replicados. Estes sistemas incluem múltiplos nós, normalmente alojados em centros de dados na nuvem, localizados em diferentes regiões geográficas e separados por latências de rede elevadas. Os dados são tipicamente divididos em partições e cada nó replica uma ou mais dessas partições.

Para garantir alta disponibilidade e durabilidade dos dados, estes sistemas replicam informação por múltiplas regiões. Contudo, a replicação geográfica levanta desafios significativos ao nível da coordenação entre réplicas, especialmente quando se pretende oferecer garantias de coerência forte. Muitos serviços requerem garantias transaccionais como a serializabilidade ou o isolamento instantâneo, o que implica a necessidade de estabelecer uma ordem total entre transacções.

Nos últimos anos, muitos sistemas transaccionais têm vindo a ser construídos com base em históricos distribuídos partilhados, que fornecem uma ordenação total dos registos como propriedade fundamental, permitindo a execução determinista das operações em todos os nós. Contudo, oferecer uma ordenação total num sistema geo-distribuído e parcialmente replicado não é trivial. A ordenação de transacções pode exigir comunicação entre múltiplas regiões, o que degrada o desempenho em cenários com latências inter-regionais elevadas. Neste artigo,

abordamos técnicas para impor uma ordenação total de transacções em sistemas parcialmente replicados e geo-distribuídos.

Mais concretamente, focamos-nos no dilema entre a escolha de algoritmos de ordenação e algoritmos de ordenação centralizada (i.e. baseada em sequenciadores). Algoritmos de ordenação centralizados ordenam operações com dois passos de comunicação e com um número linear de mensagens. No entanto, o custo desta ronda é proporcional apenas à distância entre a região originadora da transacção e a região sequenciadora. Como tal, transacções com graus fortes de localidade, cujas transacções interagem maioritariamente entre regiões geograficamente próximas, terão de pagar um custo elevado para contactar o possivelmente remoto sequenciador. Algoritmos de ordenação genuínos, como o algoritmo de Skeen [6], mitigam este problema ao necessitar de coordenação apenas entre as regiões participantes da transacção. No entanto, de modo a manter um número linear de mensagens, o algoritmo de Skeen seleciona um coordenador para agregar e disseminar mensagens, usando três passos de comunicação. Por sua vez, a latência da coordenação é dependente da localização do nó coordenador.

Uma vez que o algoritmo de Skeen usa um passo de comunicação adicional em relação a um algoritmo baseado num único sequenciador centralizado, a escolha apropriada do coordenador pode ajudar a reduzir este impacto. Assim, apresentamos uma avaliação experimental que pretende aferir as vantagens de selecionar de forma informada o conjunto de coordenadores em sistemas transaccionais geo-distribuídos, que tentam tirar partido da localidade dos dados e da topologia da rede de servidores.

2 Trabalho Relacionado

Sistemas distribuídos modernos dependem frequentemente de históricos (do Inglês, *logs*) distribuídos para garantir a coerência entre réplicas e assegurar uma ordem global das operações. Um histórico distribuído é, conceptualmente, uma sequência linear de eventos partilhada por múltiplos nós, permitindo uma execução ordenada e coerente de comandos em sistemas geo-replicados. Estes históricos são úteis em serviços que requerem coerência forte, como as bases de dados replicadas com suporte para transacções.

Históricos distribuídos como os descritos em [4,10,17], utilizam protocolos de consenso como o Paxos [13] para ordenar todas as operações. Em sistemas geo-replicados, estas soluções apresentam uma latência elevada. Por outro lado, trabalhos como [2,5,8,12], exploram a utilização de estruturas baseadas em árvores para melhorar a disseminação e agregação dos registos num histórico comum. Estas abordagens tentam reduzir o custo de coordenação ao tirar partido da hierarquia da rede, mas podem aumentar a latência das operações que necessitam atravessar toda a árvore de disseminação.

Mais recentemente, vários trabalhos têm analisado a ideia de explorar a localidade em históricos distribuídos [14], onde se procura reduzir o custo de coordenação sem abdicar da coerência, tirando partido da afinidade geográfica das operações. Um exemplo notável destes sistemas é o SLOG [16]. No SLOG, cada

Sistemas	Modelo de Coerência	Transacções	Coordenação
CORFU[4]	Forte	✗	Sequenciador
Scalog[10]	Forte	✗	Sequenciador com Agregadores
vCorfu[17]	Forte	✓	Sequenciador
FuzzyLog[14]	Forte	✓	Skeen (cliente)
SLOG[16]	Forte	✓	Sequenciador
Detock[15]	Forte	✓	Optimista
Eunomia[12]	Fraca	✗	Árvore
Saturn[8]	Fraca	✗	Árvore
ENGAGE[5]	Fraca	✗	Árvore
Amaro [2]	Forte	✓	Árvore

Tabela 1. Comparação entre Sistemas

geo-localização é responsável por um subconjunto de objetos. As transacções são atribuídas a uma região quando todos os objetos a que acedem são geridos por essa região. Nestes casos, a transacção pode ser ordenada localmente, com baixa latência e sem coordenação externa. No caso de transacções globais - isto é, que acedem a objetos espalhados por várias regiões - a ordenação é realizada por uma região pré-definida pelo sistema, que age como sequenciador. Este modelo reduz drasticamente a latência para transacções locais mas a latência das operações globais é penalizada pelo acesso a um nó central, independentemente da localização dos dados.

A Tabela 1 compara as principais técnicas para ordenação de transacções propostas na literatura, com base em três dimensões.

- **Modelo de Coerência:** Define as garantias sobre a ordem e visibilidade das atualizações. Sistemas com coerência forte (ex.: CORFU, Scalog, SLOG) garantem serializabilidade, mas sofrem com maior custos de coordenação em ambientes geo-distribuídos. Já sistemas com coerência mais fraca (ex.: Eunomia, Saturn, ENGAGE) são mais eficientes e escaláveis, mas menos adequados para cargas que exigem garantias rigorosas.
- **Suporte Transaccional:** É fundamental para garantir atomicidade em operações distribuídas. Sistemas como vCorfu, SLOG e Detock suportam transacções, enquanto CORFU, Scalog e Saturn não, o que limita a sua aplicabilidade em cenários transaccionais.
- **Mecanismos de Coordenação:** Afeta diretamente a escalabilidade e desempenho. CORFU e vCorfu usam sequenciadores centralizados (mais simples, mas com grandes limitações). O Scalog tenta reduzir estas fraquezas agregando informação relativa às atualizações, à custa de uma maior latência. O FuzzyLog adota um mecanismo baseado no cliente utilizando o algoritmo de Skeen, eliminando a coordenação centralizada, mas podendo introduzir latência se o cliente estiver geograficamente distante dos nós participantes. Detock segue uma abordagem otimista, mais leve, mas sensível a conflitos. Por fim, sistemas baseados em árvores (ex.: Saturn) propagam metadados de forma eficiente e escalável.

3 Ordenação Genuína Ciente da Topologia

3.1 Modelo de Sistema

Assumimos um sistema geo-distribuído com múltiplos centros de dados e localizações geográficas distintas. Cada região possui vários servidores e a capacidade de replicar dados e informação de controlo localmente. Sem perda de generalidade, assumimos que a replicação e coordenação das réplicas de uma dada região é assegurada por um algoritmo do tipo Paxos que mantém um histórico local. Assumimos que o sistema suporta replicação parcial dos dados, sendo que um fragmento dos dados pode ser replicado num conjunto distintos de regiões. Alguns dados podem estar armazenados apenas numa região. Transacções que acedem a dados armazenados numa única região são designadas por *transacções locais* e transacções que acedem a dados replicados em múltiplas regiões são designadas por *transacções globais*.

As transacções locais não requerem a coordenação de múltiplas regiões. São ordenadas entre si pelo histórico partilhado local e executadas na ordem total definida pelo Paxos. As transacções globais são ordenadas em duas fases. Primeiro, é realizado um processo de coordenação inter-regional para garantir uma ordem total entre todas as operações globais. Depois de uma transacção global ser ordenada em relação às restantes transacções globais, esta é inserida no histórico local das regiões participantes, sendo desta forma intercalada na sequência de transacções locais em cada região. Em cada região, o Paxos garante que todas as réplicas dessa região fazem este intercalamento de forma mutuamente coerente. Tal como acontece em sistemas como o Spanner [9], assumimos que em cada região o processo que serve como líder do Paxos serve como representante para estabelecer a coordenação inter-regiões. Cada passo do algoritmo de coordenação global é registado no histórico local, de forma a que a coordenação possa continuar mesmo que a réplica líder tenha de ser substituída.

3.2 Ordenação Global

Para reduzir a latência do processo de ordenação de transacções globais, recorremos a algoritmos *genuínos* [11]. Informalmente, um algoritmo de ordenação é genuíno se, para ordenar uma dada transacção, só é necessário trocar mensagens entre as regiões envolvidas na transacção. Concretamente, recorremos ao algoritmo de Skeen [6] para ordenar as transacções globais. Este algoritmo impõe uma ordem total sobre eventos distribuídos através da atribuição de estampilhas temporais. O funcionamento decorre em duas fases: numa primeira fase, cada participante calcula um estampilha temporal local e envia esse valor a um coordenador; na segunda fase, a estampilha temporal global é definida como o máximo entre as estampilhas temporais propostas. Este valor determina a posição da transacção na ordem total e garante consistência entre regiões.

O algoritmo de Skeen pode ser executado em modo todos-para-todos, em que as regiões envolvidas na transacção comunicam diretamente umas com as outras. Esta solução tem a desvantagem do custo de mensagens ser quadrático com o

número de participantes da transacção. Alternativamente, pode ser eleito um coordenador (entre as regiões envolvidas na transacção), que agrega e dissemina a informação necessária para estabelecer a ordem, com a vantagem de reduzir a complexidade de mensagens para um custo linear. Na nossa solução optamos por esta segunda alternativa.

3.3 Selecção do Coordenador

Nalguns casos, o cliente que submete a transacção pode servir como coordenador do algoritmo de Skeen. No nosso caso não usamos esta solução por duas razões. Em primeiro lugar, se o cliente estiver numa localização remota, a latência será grande. Em segundo lugar, tolerar a falha do cliente pode ser mais complicado do que tolerar a falha de um servidor, uma vez que os servidores replicam o seu estado através do Paxos.

A seleção do coordenador no algoritmo de Skeen pode ser aleatória, isto é, ser escolhida à sorte uma das regiões envolvidas na transacção. Neste trabalho avaliamos uma alternativa que consiste em usar o conhecimento sobre a topologia (e também, quando possível, sobre a carga no sistema) para escolher o coordenador de forma informada. Mais concretamente, quando uma transacção é submetida, o sistema verifica quais as regiões envolvidas e estima a latência de execução do algoritmo de Skeen em função da localização do coordenador, escolhendo para coordenador a região que minimiza a latência estimada. Esta escolha pode ser feita previamente, numa fase de configuração do sistema. Assim, assumimos que para cada combinação de regiões possível, é previamente escolhido um coordenador que é conhecido por todos os participantes no sistema. A Figura 1 ilustra uma possível atribuição de coordenadores num sistema com três regiões. Cada caixa colorida representa uma região geográfica distinta do sistema, enquanto os círculos no interior de cada região representam os coordenadores atribuídos a essa localização. Estes coordenadores incluem tanto coordenadores locais ("A", "B" e "C") como coordenadores responsáveis por transacções globais (por exemplo, "AB", "BC" ou "ABC"), consoante as combinações das regiões envolvidas. A figura evidencia assim a distribuição física dos coordenadores no sistema, refletindo decisões de colocação que procuram minimizar a latência e distribuir a carga de forma equilibrada entre nós.

Na atual versão do protótipo, a escolha dos coordenadores é feita estaticamente e configurada através de um ficheiro. Realizar este processo de forma dinâmica não é conceptualmente complexo e será alvo de trabalho futuro. Nesta versão usamos exclusivamente a latência da rede como critério para escolher o coordenador e não temos em consideração a carga em cada região, mas esse é um parâmetro que pensamos também vir a incluir no futuro, para promover uma melhor distribuição de carga no sistema.

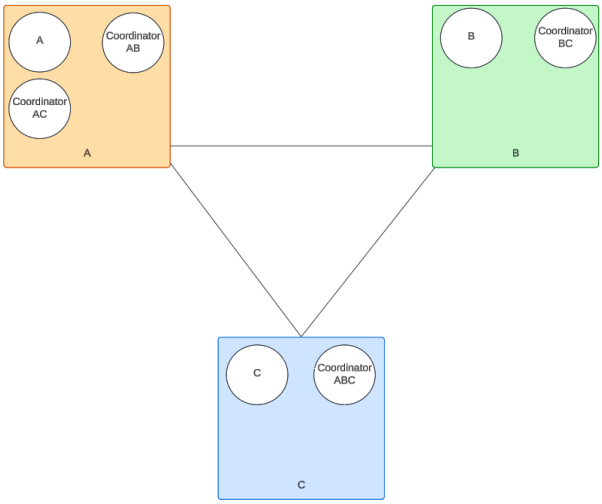

Figura 1. Selecção de Coordenadores

3.4 Processamento de transacções

Como já mencionado anteriormente, as transacções neste sistema são classificadas em transacções locais e globais, sendo o respetivo tratamento distinto consoante o tipo.

Transacções Locais As transacções locais, conforme já referido, envolvem apenas uma região e podem ser processadas e confirmadas localmente, sem necessidade de comunicação com outras regiões ou coordenadores.

1. **Início da transacção pelo Cliente:** O cliente envia a transacção à região correspondente. Por exemplo, se a transacção envolver o objeto "A", o cliente envia-a diretamente para a região responsável por esse objeto.
2. **Verificação Local e Confirmação:** A região verifica que se trata de uma transacção local, uma vez que apenas acede a objetos da sua responsabilidade. Como não é necessária coordenação externa, a transacção é diretamente inserida no histórico local da região, o qual mantém uma sequência totalmente ordenada de transacções através de um mecanismo baseado em Paxos. Esta abordagem minimiza a latência, permitindo um processamento rápido e eficiente de transacções locais.

Transacções Globais Transacções globais, como já mencionado, abrangem várias regiões e requerem coordenação de forma a assegurar uma ordem total globalmente consistente.

1. **Início da transacção pelo Cliente:** O cliente envia a transacção a todas as regiões envolvidas.
2. **Verificação Local e Encaminhamento para o Coordenador:** Cada região verifica se a transacção afeta outras regiões. Se sim, a transacção é classificada como global, sendo necessária coordenação com o auxílio de um coordenador. Para garantir uma ordem consistente, cada região mantém um relógio lógico, que é incrementado monotonamente sempre que uma nova transacção é processada. Assim, é atribuída à transacção uma estampilha temporal local superior a qualquer uma previamente emitida naquela localização. Além disso, cada região guarda internamente um registo ordenado das transacções às quais atribuiu estampilhas, o que permitirá mais tarde assegurar que estas são confirmadas pela ordem correta.
3. **Coordenação via Algoritmo de Skeen:** As regiões e o coordenador responsável pela combinação de regiões (por exemplo, "AB" para "A" e "B") executam colaborativamente o algoritmo de Skeen. Como referimos, a atribuição da região que serve como coordenadora é feita previamente, numa fase de configuração do sistema. Cada região envia a sua estampilha temporal proposta ao coordenador, que as agrega e determina a ordem global tomando o máximo entre elas. Este processo assegura uma ordem total entre todas as geo-localizações participantes, minimizando ao mesmo tempo a sobrecarga de comunicação.
4. **Atualização dos Relógios Lógicos:** Após receberem a estampilha temporal global determinada pelo coordenador, todas as regiões participantes atualizam os seus relógios lógicos para garantir que transacções futuras não sejam atribuídas estampilhas inferiores àquela já acordada.
5. **Verificação de Ordem antes da Confirmação:** Antes de enviar uma transacção para a fila de confirmação, cada geo-localização tem de garantir que não existem transacções pendentes às quais tenha atribuído estampilhas inferiores. Ou seja, uma transacção só pode avançar para a fase de confirmação se todas as transacções com estampilhas mais antigas já tiverem sido confirmadas. Esta verificação evita a confirmação fora de ordem, preservando a consistência sequencial local.
6. **Confirmação da transacção:** Uma vez estabelecida a estampilha temporal global, a transacção é adicionada a uma fila em cada geo-localização participante. A transacção aguarda até atingir o topo da fila, momento em que é utilizado o algoritmo Paxos para inserir a transacção no histórico local de cada região segundo a ordem global acordada.

4 Avaliação

Nesta secção, avaliamos o impacto da escolha cuidadosa do coordenador de cada transacção considerando vários níveis de localidade de transacções globais. Avaliamos a nossa arquitetura contra dois outros sistemas: SLOG, que utiliza um coordenador central para ordenar transacções globais, e o protocolo de Skeen com escolha aleatória de coordenador.

Para garantir uma comparação justa, reutilizámos e estendemos o código C++ público do sistema SLOG, desenvolvendo uma base comum que suporta tanto o nosso algoritmo como o SLOG com coordenador central. Ambas as variantes partilham a mesma infraestrutura de rede e implementação base, assegurando condições uniformes de avaliação.

No SLOG, a ordenação das transacções globais é sempre realizada por uma única região central, independentemente das regiões efetivamente envolvidas. No nosso cenário, esse coordenador central está localizado na região dos Estados Unidos, mais concretamente em *us0*. Esta escolha fixa penaliza a latência de transacções globais, sobretudo quando os dados envolvidos estão longe dessa região.

Para os nossos testes, assumimos uma carga composta exclusivamente por transacções globais. Avaliamos o impacto da localidade no desempenho das transacções variando o número de regiões participantes e o grau de localidade entre participantes da transacção.

4.1 Ambiente de Testes e Configuração

Os testes foram realizados com nove máquinas físicas, cada uma representando uma região distinta do sistema geo-distribuído. As mesmas máquinas foram também utilizadas para simular clientes concorrentes que geram transacções, simulando 9 clientes por máquina. Cada transacção realiza operações de leitura e escrita sobre 9 objetos, distribuídos de forma uniforme pelas regiões envolvidas. Controlamos o grau de contenção no sistema através de um parâmetro de dispersão, que designamos por d. Este parâmetro regula a proporção de objetos "quentes" — i.e., os mais frequentemente acedidos. Um valor mais elevado de d implica uma maior dispersão do acesso, ou seja, mais objetos diferentes são tocados, o que reduz a contenção no sistema. Para as nossas experiências, utilizamos um valor de dispersão alto de 10000, correspondente a um caso de contenção bastante baixa, semelhante ao utilizado no estado da arte [15].

Para simular um ambiente geo-distribuído realista, introduzimos latências artificiais entre regiões. Estes valores foram definidos manualmente com o objetivo de refletir relações geográficas típicas, como a proximidade entre regiões dentro do mesmo continente e maior distância entre continentes distintos. A Figura 2 ilustra a topologia do sistema e os tempos de ida-e-volta (RTT) configurados entre regiões e continentes. A região *us0* foi escolhida como coordenador central para o SLOG (representada a vermelho na Figura 2). As regiões centrais de cada continente, *us0*, *eu0*, e *as0* respetivamente.

4.2 Localidade Intra-Continental

Começamos por avaliar o impacto de localidade alta nas transacções globais. Nesta experiência, as transacções globais interagem apenas com regiões do seu continente originado, tocando em todas as regiões deste. Chamamos a estas transacções intra-continente.

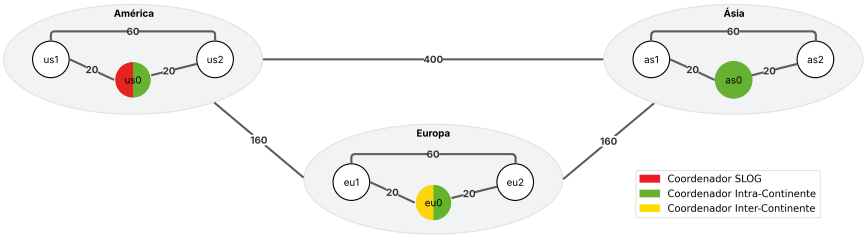

Figura 2. Topologia do Sistema. A região vermelha representa o coordenador central do SLOG. As regiões verdes representam os coordenadores do algoritmo de Skeen de transacções intra-continente. A região amarela representa o coordenador do algoritmo de Skeen de transacções inter-continente.

A Figura 3 apresenta a CDF da latência das transacções globais originadas de cada continente: América, Europa, e Ásia. Quando a localidade é alta, o algoritmo de Skeen oferece melhor desempenho que o SLOG nas regiões afastadas do sequenciador, oferecendo até 4 vezes melhor latência no caso do continente Asiático. Embora o algoritmo de Skeen necessite do dobro dos passos de comunicação de coordenação em comparação ao SLOG, o custo da comunicação intra-continente agregado é significativamente menor do que o custo da comunicação inter-continental. Este custo adicional do Skeen é aparente no continente Americano, onde o SLOG apresenta melhores resultados.

Adicionalmente, podemos observar que a escolha do coordenador dentro de uma dada região impacta significativamente o desempenho do algoritmo de Skeen. Tomemos o caso do continente Europeu, onde a região *eu0* encontra-se numa posição central relativamente às duas outras regiões *eu1* e *eu2*. O posicionamento do coordenador na região *eu0* minimiza o tempo de execução do Skeen em dois aspetos: primeiro, este minimiza a latência máxima do protocolo (i.e. o tempo desde o inicio do protocolo ate à chegada da estampilha final a todas as regiões). Dado que o cliente terá de esperar pela execução desta em todas as regiões participantes, minimizar esta latência impacta diretamente a latência da transacção em si. Segundo, este também minimiza a latência nas regiões que mantêm as transacções num estado tentativo. Ao minimizar este tempo, as transacções irão desbloquear mais rapidamente o envio das transacções que já tenham obtido a estampilha final mas que estejam bloqueadas devido a transacções concorrentes. A combinação destes dois fatores leva a uma redução da latência média de transacções entre regiões de 35%.

4.3 Localidade Inter-Continental

Avaliamos agora o caso em que para além das transacções intra-continentais, uma pequena percentagem (10%) das transacções são inter-continentais, interagindo com uma região de cada continente.

A Figura 4 apresenta a CDF da latência das transacções originadas em cada região. A introdução destas transacções afeta significativamente o desem-

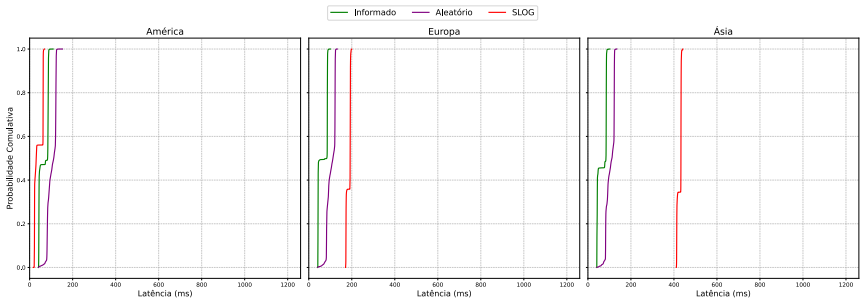

Figura 3. CDF das latências por região, comparando a escolha aleatória do coordenador com a seleção informada da região mais central.

penho dos protocolos baseados em Skeen, enquanto que o desempenho do SLOG manteve-se estável. Este agravamento deve-se à diferença significativa entre a comunicação intra- e inter-continental entre regiões. Transacções intra-continentais obtêm as suas estampilhas temporais finais significativamente mais rápido do que as transacções inter-continentes, causando uma fila de espera pela inferência da ordenação final do Skeen inter-continental. Este efeito é conhecido como o Efeito Caravana [7]. Inicialmente observado no Sistema R [3], este efeito dita que o desempenho de um sistema é geralmente determinado pela sua transacção mais lenta, tendo sido também detetado em sistemas de ordenação genuínos [1].

Na presença deste efeito, os ganhos introduzidos pela utilização do algoritmo de Skeen são reduzidos significativamente, apresentando pior latência média que o SLOG nos continentes americanos e europeus. No entanto, ambos os sistemas continuam a apresentar melhor latência média no continente asiático, onde o custo de comunicação com o coordenador centralizado no continente Americano domina a latência de transacções do SLOG.

Embora a escolha informada de coordenador continue a apresentar melhorias de desempenho, o seu impacto é também reduzido devido ao efeito caravana, apresentando uma melhoria de apenas 10% na latência média das transacções originadas nos continentes Americanos e Asiáticos. No entanto, os ganhos continuam significativos nas transacções originadas na Europa, apresentando uma melhoria de 40% relativamente à escolha aleatória. Esta discrepância de ganhos deve-se ao diferente impacto que a escolha informada tem na redução do tempo de execução do Skeen nos vários continentes. Considere uma transacção inter-continental originada no continente Europeu. O tempo pelo qual a transacção irá manter-se em um estado pendente no continente Europeu, caso o coordenador do algoritmo de Skeen esteja localizado nos continentes Americanos, Europeus ou Asiáticos é, respetivamente, de 360, 160 e 360 milissegundos, correspondente a uma latência médio de 293 milissegundos no caso aleatório - um valor 80% maior do que o obtido pelo sistema informado, cuja latência são uns constantes 160 milissegundos. No entanto, nos casos do continente Asiático e Americano, a diferença entre o tempo médio aleatório e o informado é de apenas 7%, dado que

independentemente a escolha de coordenador, a latência de coordenação do algoritmo será dominada pelos custos de coordenação entre os continentes Asiáticos e Americanos.

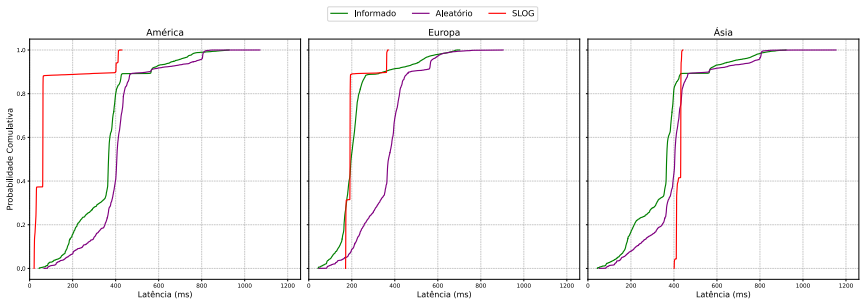

Figura 4. CDF das latências por região, comparando o impacto da seleção de coordenador na presença de transações inter-região.

4.4 Impacto do Efeito Caravana

Para compreender melhor o impacto do efeito caravana no desempenho do sistema e na escolha informada, modificamos a experiência anterior de modo a que a transacção inter-continental interaja apenas com os continentes europeus e asiáticos, escolhendo uma região de cada continente. Dado que estas transacções interagem apenas com dois continentes, a escolha de coordenador destas transacções foi aleatória.

A Figura 5 apresenta a CDF da latência das transacções originadas em cada região. Os sistemas baseados no algoritmo de Skeen apresentam uma melhoria de desempenho significativa nos continentes Europeus e Asiáticos relativamente aos apresentados na Figura 4. Esta melhoria deve-se à remoção da demorada coordenação efetuada pelo continente Americano e Asiático das transacções inter-continentais, reduzindo a latência de execução do algoritmo de Skeen e, por consequência o seu efeito caravana associado.

Curiosamente, nos continentes ainda afetados pelo efeito caravana, a utilização da escolha informada não introduz qualquer melhoria significativa relativamente à utilização de escolha aleatória. Isto porque os ganhos introduzidos pela escolha informada das transacções intra-continente são em grande parte ofuscada pelo efeito caravana associado às transacções inter-continentais.

5 Conclusão

Os algoritmos de ordenação genuínos, tal como o algoritmo de Skeen, possuem várias vantagens em relação aos algoritmos de ordenação centralizados. Infelizmente, quando são concretizados com ajuda de um coordenador (o que permite

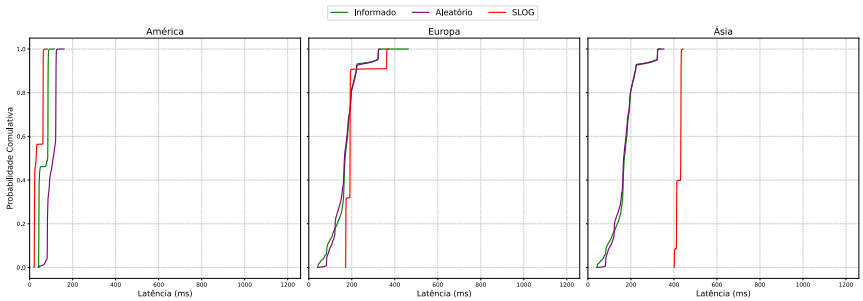

Figura 5. CDF das latências por região, considerando transações inter-região apenas entre Europa e Asia.

manter uma complexidade de mensagens linear), utilizam um passo de comunicação adicional. Neste trabalho avaliamos o papel que a escolha informada do coordenador pode ter no desempenho destes sistemas. Os nossos resultados mostram que a escolha do coordenador consegue de facto reduzir a latência de execução do algoritmo do Skeen. No entanto, quando o sistema combina transacções com elevado grau de localidade com transacções que abrangem em todas as regiões, a interferência das últimas nas primeiras reduz as vantagens que advêm da escolha informada dos coordenadores. Isto sugere que mais investigação é necessária para optimizar o desempenho de algoritmos genuínos em sistemas transaccionais geo-replicados.

Agradecimentos: Este trabalho foi suportado pela FCT – Fundação para a Ciência e a Tecnologia, através dos projectos UIDB/50021/2020 e GLOG (financiado pelo OE com a ref. LISBOA2030-FEDER-00771200).

Referências

1. Ahmed-Nacer, T., Sutra, P., Conan, D.: The convoy effect in atomic multicast. In: SRDS ' 16 (Sep 2016)
2. Amaro, J.: A distributed and hierarchical architecture for deferred validation of transactions in key-value stores (2018)
3. Astrahan, M.M., Blasgen, M.W., Chamberlin, D.D., Eswaran, K.P., Gray, J., Griffiths, P.P., III, W.F.K., Lorie, R.A., McJones, P.R., Mehl, J.W., Putzolu, G.R., Traiger, I.L., Wade, B.W., Watson, V.: System R: relational approach to database management. ACM Trans. Database Syst. **1**(2), 97–137 (1976)
4. Balakrishnan, M., Malkhi, D., Davis, J.D., Prabhakaran, V., Wei, M., Wobber, T.: CORFU: A distributed shared log. ACM Trans. Comput. Syst. **31**(4), 10 (2013)
5. Belém, M., Fouto, P., Lykhenko, T., Leitão, J., Preguiça, N.M., Rodrigues, L.: Engage: Session guarantees for the edge. In: 31st International Conference on Computer Communications and Networks, (ICCCN) (Jul 2022)
6. Birman, K.P., Joseph, T.A.: Reliable communication in the presence of failures. ACM Trans. Comput. Syst. **5**(1), 47–76 (1987)

7. Blasgen, M.W., Gray, J., Mitoma, M.F., Price, T.G.: The convoy phenomenon. ACM SIGOPS Oper. Syst. Rev. **13**(2), 20–25 (1979)
8. Bravo, M., Rodrigues, L.E.T., Roy, P.V.: Saturn: A distributed metadata service for causal consistency. In: Proceedings of the Twelfth European Conference on Computer Systems (EuroSys) (Apr 2017)
9. Corbett, J.C., Dean, J., Epstein, M., Fikes, A., Frost, C., Furman, J.J., Ghemawat, S., Gubarev, A., Heiser, C., Hochschild, P., Hsieh, W.C., Kanthak, S., Kogan, E., Li, H., Lloyd, A., Melnik, S., Mwaura, D., Nagle, D., Quinlan, S., Rao, R., Rolig, L., Saito, Y., Szymaniak, M., Taylor, C., Wang, R., Woodford, D.: Spanner: Google's globally distributed database. ACM Trans. Comput. Syst. **31**(3), 8 (2013)
10. Ding, C., Chu, D., Zhao, E., Li, X., Alvisi, L., van Renesse, R.: Scalog: Seamless reconfiguration and total order in a scalable shared log. In: 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20) (Feb 2020)
11. Guerraoui, R., Schiper, A.: Genuine atomic multicast in asynchronous distributed systems. Theoretical Computer Science **254**(1), 297–316 (2001)
12. Gunawardhana, C., Bravo, M., Rodrigues, L.E.T.: Unobtrusive deferred update stabilization for efficient geo-replication. In: 2017 USENIX Annual Technical Conference (ATC) (Jul 2017)
13. Lamport, L.: Paxos made simple, fast, and byzantine. In: Procedings of the 6th International Conference on Principles of Distributed Systems. pp. 7–9 (2002)
14. Lockerman, J., Faleiro, J., Kim, J., Sankaran, S., Abadi, D., Aspnes, J., Sen, S., Balakrishnan, M.: The FuzzyLog: A partially ordered shared log. In: 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18). pp. 357–372 (Oct 2018)
15. Nguyen, C.D.T., Miller, J.K., Abadi, D.J.: Detock: High performance multi-region transactions at scale. Proceedings of the ACM on Management of Data **1**(2), 1–27 (2023)
16. Ren, K., Li, D., Abadi, D.J.: SLOG: serializable, low-latency, geo-replicated transactions. Proceedings of the VLDB Endowment **12**(11), 1747–1761 (2019)
17. Wei, M., Tai, A., Rossbach, C.J., Abraham, I., Munshed, M., Dhawan, M., Stabile, J., Wieder, U., Fritchie, S., Swanson, S., Freedman, M.J., Malkhi, D.: vcorfu: A cloud-scale object store on a shared log. In: 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17) (Mar 2017)