**Análise da Hipótese**

1. **Impacto do Filtro de Atualizações Desnecessárias**:
   * O **v1** foi projetado para filtrar atualizações desnecessárias, o que significa que ele processa apenas um subconjunto dos dados recebidos, descartando atualizações redundantes ou irrelevantes (por exemplo, mudanças insignificantes no livro de ordens ou dados fora de certos critérios).
   * Isso reduz o número total de atualizações processadas, o que é refletido diretamente na **taxa de atualização menor** (31.00/s no v1 contra 44.04/s no v0). Menos atualizações processadas por segundo indicam que o filtro está funcionando, diminuindo a carga de trabalho do sistema.
   * A redução no volume de atualizações também é consistente com os **tamanhos menores dos heaps** observados no v1 (por exemplo, bidHeapSize de 4-4 para 2-2 em BTCUSDT), já que menos dados relevantes são armazenados.
2. **Por que o Tempo Médio de Atualização é Mais Lento?**:
   * Embora o v1 processe menos atualizações, o **tempo médio de atualização** é maior (87.497ms vs. 74.139ms). Isso pode parecer contra-intuitivo, já que menos dados geralmente implicam menos processamento. No entanto, o filtro de atualizações desnecessárias pode introduzir uma **sobrecarga computacional adicional**:
     + **Custo do filtro**: Avaliar quais atualizações são desnecessárias requer verificações adicionais (por exemplo, comparar preços, volumes ou outros critérios com thresholds). Esse processo de filtragem consome tempo de CPU, aumentando o tempo necessário para processar cada atualização válida.
     + **Complexidade das atualizações processadas**: As atualizações que passam pelo filtro podem ser mais complexas ou exigir mais processamento (por exemplo, atualizações que alteram significativamente o livro de ordens). Isso pode aumentar o tempo médio por atualização, mesmo que o número total de atualizações seja menor.
     + **Possível gargalo**: O filtro pode estar mal otimizado ou implementado de forma que introduza latência adicional, como verificações redundantes ou alocações de memória frequentes.
3. **Consistência com Outras Métricas**:
   * A **taxa de atualização menor** (31.00/s vs. 44.04/s) corrobora a ideia de que o v1 está processando menos atualizações, o que é esperado com um filtro ativo.
   * A **menor contagem de arbitragem** no v1 (máximo de 35 vs. 37) pode ser um efeito colateral do filtro, já que algumas atualizações descartadas poderiam conter pequenas discrepâncias de preço que desencadeariam oportunidades de arbitragem no v0.
   * As **latências menores** nas exchanges (Bybit, Binance, Deribit) no v1 sugerem que o sistema está mais eficiente na comunicação externa, o que pode ser independente do filtro, mas indica que o aumento no tempo de atualização não está relacionado a problemas de rede.
   * Os **tamanhos menores dos heaps** no v1 reforçam que menos dados estão sendo armazenados, o que é consistente com a filtragem de atualizações desnecessárias.
4. **Possíveis Explicações Alternativas**:
   * **Mudanças no algoritmo**: Além do filtro, o v1 pode ter outras otimizações ou mudanças no pipeline de processamento (por exemplo, estruturas de dados diferentes ou lógica de atualização mais complexa) que aumentam o tempo por atualização.
   * **Overhead de sincronização**: Se o v1 introduziu mecanismos adicionais de sincronização ou validação (por exemplo, para garantir consistência entre threads ou fontes de dados), isso poderia aumentar o tempo de processamento.
   * **Configuração do ambiente**: Diferenças no ambiente de teste (como carga do servidor ou concorrência) poderiam afetar o tempo de atualização, embora o runtime idêntico (1800.02s) sugira condições semelhantes.

**Avaliação da Hipótese**

A explicação mais provável para o **tempo médio de atualização mais lento** e a **taxa de atualização menor** no v1 é que o filtro de atualizações desnecessárias reduz o número de atualizações processadas (explicando a taxa menor), mas introduz uma sobrecarga computacional que aumenta o tempo necessário para processar cada atualização válida. Isso é consistente com:

* A redução na taxa de atualização (menos atualizações por segundo).
* Os tamanhos menores dos heaps (menos dados armazenados devido ao filtro).
* A ligeira redução na contagem de arbitragem (menos atualizações podem significar menos oportunidades detectadas).

No entanto, o aumento no tempo médio de atualização sugere que o filtro pode não estar otimizado ou que as atualizações filtradas são mais custosas de processar. Para confirmar, seria necessário investigar:

* O código ou a lógica do filtro no v1 (por exemplo, quais critérios são usados para descartar atualizações e quão custoso é esse processo).
* A natureza das atualizações que passam pelo filtro (são mais complexas ou exigem mais cálculos?).
* Possíveis gargalos no pipeline de processamento do v1 (como alocações de memória, locks ou verificações redundantes).

**Recomendações**

1. **Otimizar o Filtro**:
   * Analise a implementação do filtro para identificar se ele está introduzindo overhead desnecessário (por exemplo, verificações redundantes ou alocações frequentes).
   * Considere usar estruturas de dados mais eficientes ou algoritmos de filtragem mais rápidos (como filtros baseados em hash para comparações rápidas).
2. **Perfilamento do Código**:
   * Execute um perfilamento (profiling) do v1 para determinar onde o tempo está sendo gasto durante o processamento de atualizações. Isso pode revelar se o filtro ou outra parte do pipeline é o gargalo.
3. **Ajustar Critérios de Filtragem**:
   * Revise os critérios usados para descartar atualizações. Se o filtro está sendo muito agressivo, pode estar descartando atualizações que seriam úteis para detectar arbitragem. Se for muito leniente, pode não estar reduzindo a carga o suficiente.
4. **Testar com Diferentes Cargas**:
   * Teste o v1 com diferentes volumes de dados para verificar se o tempo médio de atualização aumenta proporcionalmente ao número de atualizações filtradas. Isso pode ajudar a isolar o impacto do filtro.
5. **Comparar Arbitragem**:
   * Investigue as oportunidades de arbitragem perdidas no v1 (35 vs. 37). Se o filtro está descartando atualizações que poderiam levar a arbitragens válidas, pode ser necessário ajustar os critérios de filtragem.