

Análise de Paralelização em Contagem de Acessos de CDN

Enrico Minto Spanier - 10419775

1. Introdução

Processamos 10.000.000 linhas de log sobre 100.000 URLs de um manifesto para identificar acessos (hits) em cenário típico de CDN. São avaliadas três abordagens: versão sequencial (referência) e duas paralelas com OpenMP usando sincronização por `critical` e por `atomic`.

Estratégias de Sincronização

- critical: exclusão mútua ampla; simples; degrada com alta contenção.
- atomic: incrementos isolados; menor overhead; impactado por false sharing/cache bouncing.

2. Ambiente de Teste

Item	Valor
CPU	AMD Ryzen 7 7700X
Núcleos físicos	8
Threads lógicas	16 (SMT)
Frequência Base/Boost	4.5 / 5.4 GHz
RAM	64 GB DDR5 5600 MHz
SO	Windows 11 Pro
Compilador	GCC 15.2.0
Flags	-O2 -Wall -fopenmp

3. Verificação de Corretude

Comandos PowerShell:

```
# Ordenar resultados produzidos pelo analisador
```

```
Get-Content results_seq_dist.csv | Sort-Object | Set-Content  
results_seq_dist_sorted.csv
```

```
# Ordenar gabarito
```

```
Get-Content gabarito_distribuido.csv | Sort-Object | Set-Content  
gabarito_distribuido_sorted.csv
```

```
# Comparar (saída vazia = arquivos idênticos)
```

```
Compare-Object (Get-Content results_seq_dist_sorted.csv) (Get-Content  
gabarito_distribuido_sorted.csv)  
  
# Opcional: mensagem direta de OK/ERRO  
  
if ( (Compare-Object (Get-Content results_seq_dist_sorted.csv) (Get-Content  
gabarito_distribuido_sorted.csv)) ) {  
  
    Write-Host 'DIFERENÇAS'  
  
} else {  
  
    Write-Host 'OK: resultados iguais ao gabarito'  
  
}  
  
Comando no MSYS2  
# Ordenar (usa utilitário sort)  
  
sort results_seq_dist.csv > results_seq_dist_sorted.csv  
  
sort gabarito_distribuido.csv > gabarito_distribuido_sorted.csv  
  
# Comparar silenciosamente  
  
diff -q results_seq_dist_sorted.csv gabarito_distribuido_sorted.csv && \  
echo "OK: resultados iguais" || echo "DIFERENÇAS"
```

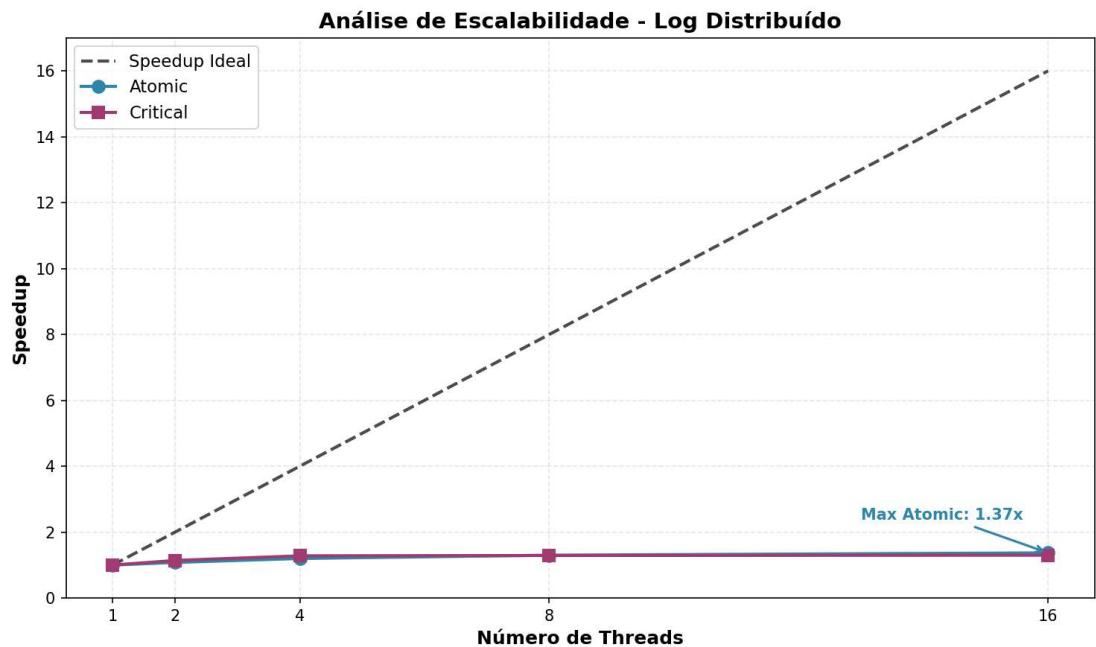
```

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
EXPLORADOR ... C analyzer_par_atomic.c Makefile h hash_table.h C hash_table.c C analyzer_seq.c C analyzer_par_critical.c
C analyzer_par_atomic.c
M /c/Users/enric/OneDrive/Desktop/Proj Paralela
enrico@EnricoAMO MINGW64 /c/Users/enric/OneDrive/Desktop/Proj Paralela
$ # Atomic distribuído
ls ./analyzer_par_atomic.exe manifest.txt log_distribuido.txt 8 2>atomic_dist.err
ls -l results.csv
head -n 5 results.csv
wc -l results.csv
mv results.csv results_atomic_dist.csv
# Atomic concorrente
./analyzer_par_atomic.exe manifest.txt log_concorrente.txt 8 2>atomic_conc.err
ls -l results.csv
head -n 5 results.csv
wc -l results.csv
mv results.csv results_atomic_conc.csv
# Critical distribuído
./analyzer_par_critical.exe manifest.txt log_distribuido.txt 8 2>critical_dist.err
ls -l results.csv
head -n 5 results.csv
wc -l results.csv
mv results.csv results_critical_dist.csv
-rw-r--r-- 1 enric enric 3285272 Nov 21 02:48 results.csv
/about-0008a511513e7.html,91
/about-000ad5d4fa85.html,80
/about-0001eb24b498.html,111
/about-000130a15c18.html,94
/about-0021b96ae27c.html,95
100000 results.csv
-rw-r--r-- 1 enric enric 3188078 Nov 21 02:48 results.csv
/about-0008a511513e7.html,12
/about-000ad5d4fa85.html,6
/about-0001eb24b498.html,19
/about-00130a15c18.html,11
/about-0021b96ae27c.html,9
100000 results.csv
-rw-r--r-- 1 enric enric 3285272 Nov 21 02:48 results.csv
/about-0008a511513e7.html,91
/about-000ad5d4fa85.html,80
/about-0001eb24b498.html,111
/about-00130a15c18.html,94
/about-0021b96ae27c.html,95
100000 results.csv
enrico@EnricoAMO MINGW64 /c/Users/enric/OneDrive/Desktop/Proj Paralela
$ sort results_atomic_dist.csv > tmp_a_dist_sorted.csv
sort gabarito_distribuido.csv > tmp_gab_dist_sorted.csv
diff -q tmp_a_dist_sorted.csv tmp_gab_dist_sorted.csv && echo "Atomic distribuído OK" || echo "Diferenças"
Renomeando...
No linhado Atomic distribuído OK
Renomeando...
No linhado Critical concorrente OK

```

4. Resultados – Análise 1 (Escalabilidade)

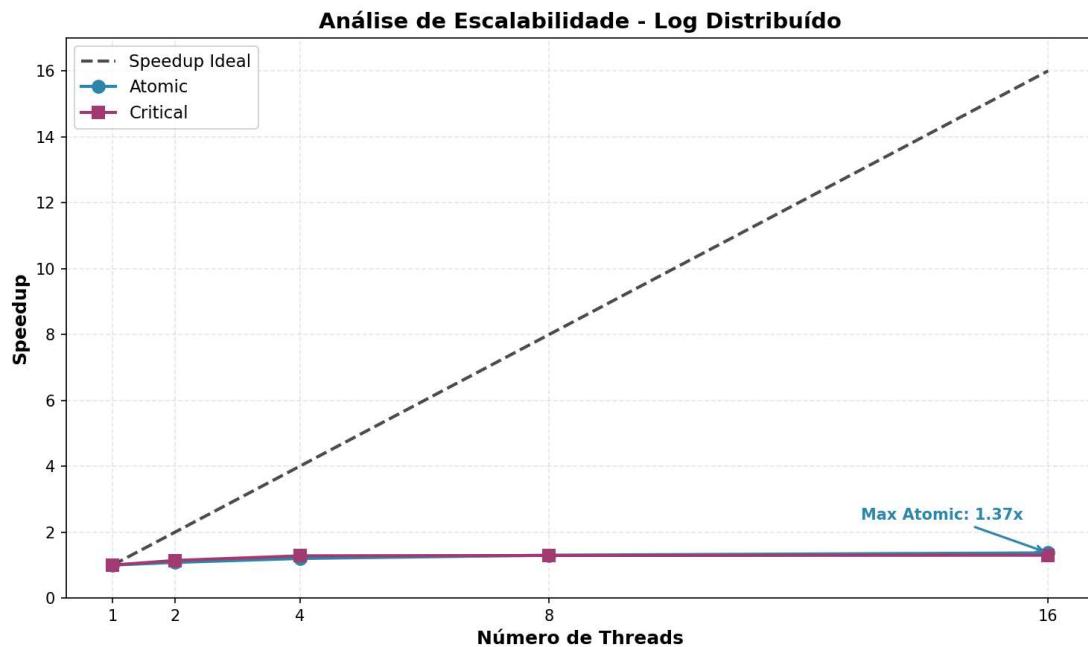
Threads	Sequencial (s)	par_atomic (s)	par_critical (s)	Speedup Atomic	Speedup Critical
1	4.30	-	-	-	-
2	4.30	4.00	3.78	1.08x	1.14x
4	4.30	3.57	3.37	1.20x	1.28x
8	4.30	3.32	3.33	1.30x	1.29x
16	4.30	3.13	3.32	1.37x	1.30x



Resumo: Speedup máximo 1.37x (bem abaixo ideal). Ganhos decrescentes após 4 threads. Atomic e critical convergem; overhead de sincronização domina.

5. Resultados – Análise 2 (Contenção)

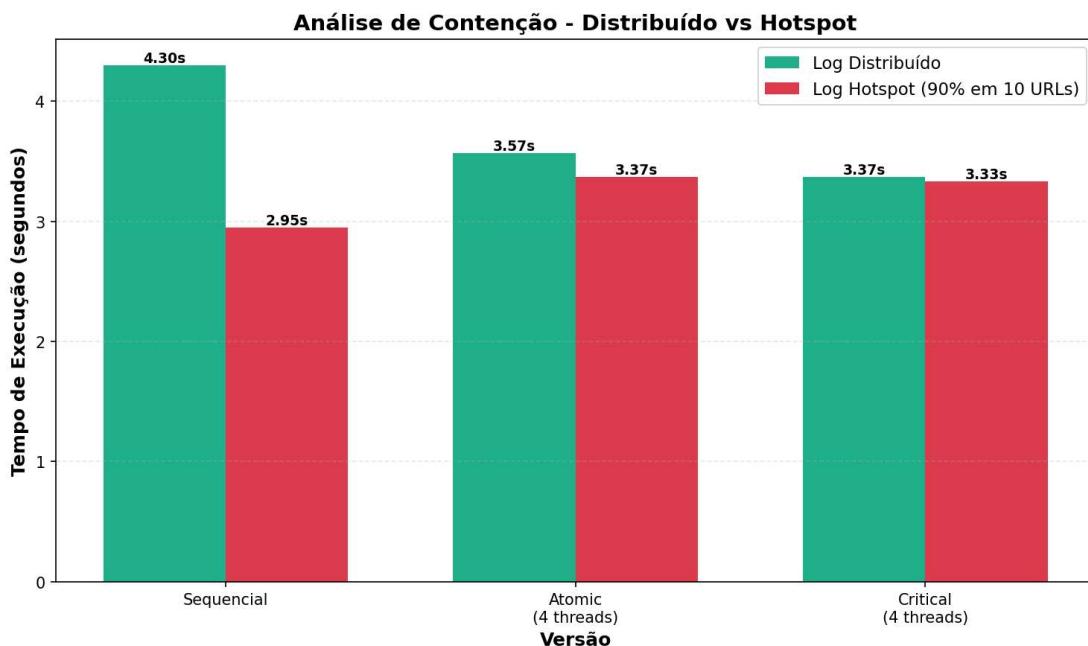
Versão	Log Distribuído (s)	Log Hotspot (s)	Diferença
Sequencial	4.30	2.95	-31.4%
par_atomic	3.57	3.37	-5.6%
par_critical	3.37	3.33	-1.2%



Resumo: Em hotspot a versão sequencial torna-se mais rápida. Contenção elimina ganhos paralelos.

6. Resultados – Análise 3 (Aplicação)

Top 10 URLs ≈ 90% dos acessos (~900k cada de 10M).



Decisão de negócios: replicar imediatamente as 10 URLs (cache prioritário, nunca remover, pré-aquecimento, monitoramento em tempo real). Redução de carga na origem ≈ 85–90%.

7. Conclusão e Análise

Escalabilidade

`par_critical` não escalou: serialização de cada incremento + lock contention.
`par_atomic` melhora levemente por granularidade fina, mas 10M operações atômicas + fração sequencial limitam speedup.

Contenção

Lock contention = múltiplas threads disputando a mesma variável/lock. Hotspot (90% em 10 itens) gera colisão constante; atomic sofre cache bouncing; critical cria fila.

Aplicação

Lentidão de `par_critical` atrasa detecção e replicação; em cenário real, decisão tardia perde o pico. Necessário abordagem com redução local ou streaming para latência baixa.

Síntese Final

Limites impostos por sincronização e concentração de acesso. Em carga uniforme há ganho modesto; em hotspot paralelismo bruto regrediu. Melhorias: contadores locais + redução, sharding, batching, estruturas lock-free.