

GCI Impact

David Ferreira Quaresma (david.quaresma@ccc.ufcg.edu.br)

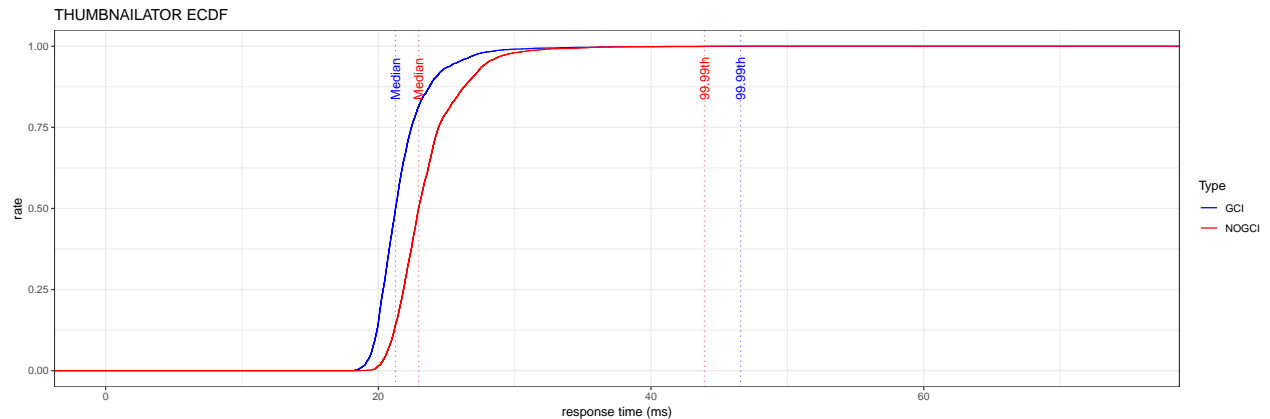
october, 2019

Runtime experiments: GCI vs NOGCI

Descrição

- **script de execução dos experimentos.**
- Número de requisições: 10 mil, **código.**
- Lógica de negócio da função: **Redimensionamento de uma imagem.**
- Escala de redimensionamento: 0.1.
- Tamanho da imagem: 131kb.
- Versão Java 1.8
- Tamanho do Heap: 256MB
- Flags de GC: -XX:+UnlockExperimentalVMOptions -XX:+UseG1GC -XX:G1NewSizePercent=90 -XX:G1MaxNewSizePercent=90
- Configuração do Proxy: -ygen=209715200

ECDF for response time



Neste ECDF é possível notar que o GCI está perdendo na cauda. Acredito que isto deve ser algum impacto externo que com o aumento do tamanho da imagem e copiando a imagem a cada requisição venha a desaparecer.

summary table of confidence interval for response time

##	stats	gci.cii	gci.cis	nogci.cii	nogci.cis
## 1	avg	22.00	22.00	23.00	24.00
## 2	p50	21.22	21.32	22.89	23.02
## 3	p95	25.44	26.08	27.84	28.21
## 4	p99	28.70	31.16	31.31	33.24
## 5	p999	37.25	46.32	36.87	43.47
## 6	p9999	45.20	47.38	42.30	45.32
## 7	p99999	46.94	47.02	44.57	44.70
## 8	dist	25.73	25.70	21.68	21.68

Os valores em colunas com .cii são referentes à valores limite inferior do intervalo de confiança. O valores em colunas com .cis são referentes à valores limite superior do intervalo de confiança. Tal qual como no gráfico com ECDF, GCI tá perdendo na cauda da latência.

GCI: comparing response time vs service time, both in milliseconds

##	stats	gci.st.cii	gci.st.cis	gci.rs.cii	gci.rs.cis
## 1	avg	20.00	20.00	22.00	22.00
## 2	p50	19.20	19.30	21.22	21.32
## 3	p95	23.10	23.71	25.44	26.08
## 4	p99	26.22	27.65	28.70	31.16
## 5	p999	34.77	40.40	37.25	46.32
## 6	p9999	39.85	42.31	45.20	47.38
## 7	p99999	41.52	41.66	46.94	47.02
## 8	dist	22.31	22.36	25.73	25.70

Os valores em colunas com .st são referentes à valores de tempo de serviço. O valores em colunas com .rt são referentes à valores de tempo de serviço. Como esperado, rt é ou maior que st ou não diferente estatisticamente.

NOGCI: comparing response time vs service time, both in milliseconds

##	stats	nogci.st.cii	nogci.st.cis	nogci.rt.cii	nogci.rt.cis
## 1	avg	21.00	21.00	22.00	22.00
## 2	p50	20.11	20.20	21.22	21.32
## 3	p95	24.60	25.03	25.44	26.08
## 4	p99	27.60	29.19	28.70	31.16
## 5	p999	33.90	39.18	37.25	46.32
## 6	p9999	38.35	42.16	45.20	47.38
## 7	p99999	40.94	41.15	46.94	47.02
## 8	dist	20.83	20.95	25.73	25.70

Os valores em colunas com .st são referentes à valores de tempo de serviço. O valores em colunas com .rt são referentes à valores de tempo de serviço. Como esperado, rt é ou maior que st ou não diferente estatisticamente.

checking errors

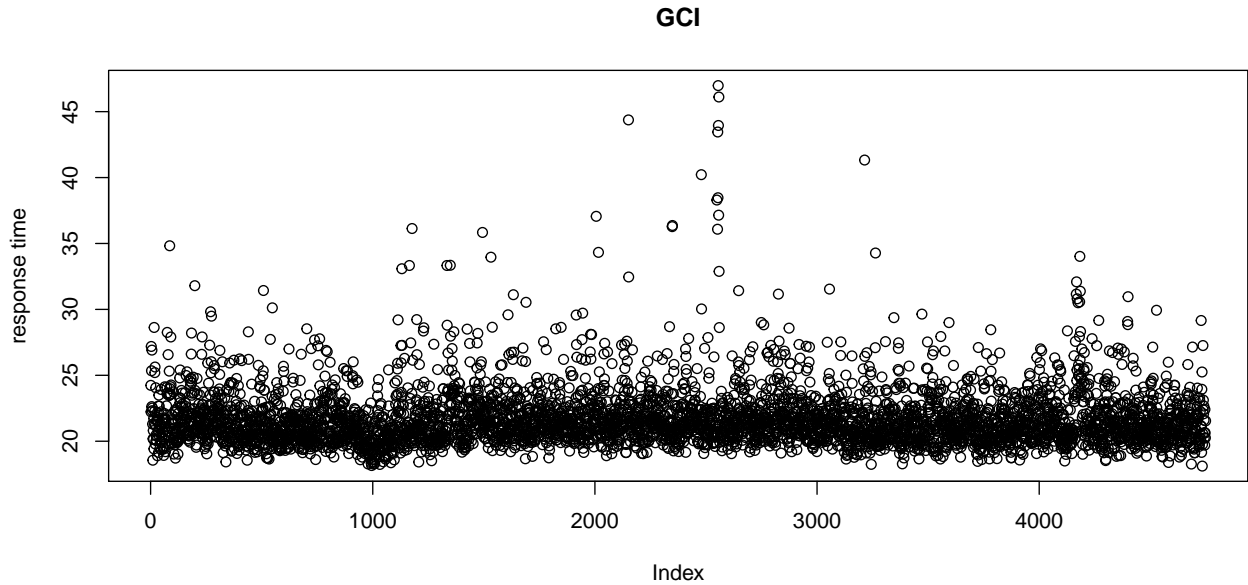
```
## GCI - number of response time values greater then its service time values: 0
## NOGCI - number of response time values greater then its service time values: 0
```

comparing difference between response time and service time for GCI vs NOGCI

##	stats	gci.st.cii	gci.st.cis	nogci.rt.cii	nogci.rt.cis
## 1	avg	2.000	2.100	2.800	2.900
## 2	p50	1.838	1.859	2.558	2.605
## 3	p95	3.221	3.394	4.419	4.659
## 4	p99	4.651	5.320	6.223	6.873
## 5	p999	6.982	8.969	9.528	13.200
## 6	p9999	8.677	9.758	12.750	14.370
## 7	p99999	9.432	9.489	13.890	13.970
## 8	dist	7.593	7.630	11.330	11.370

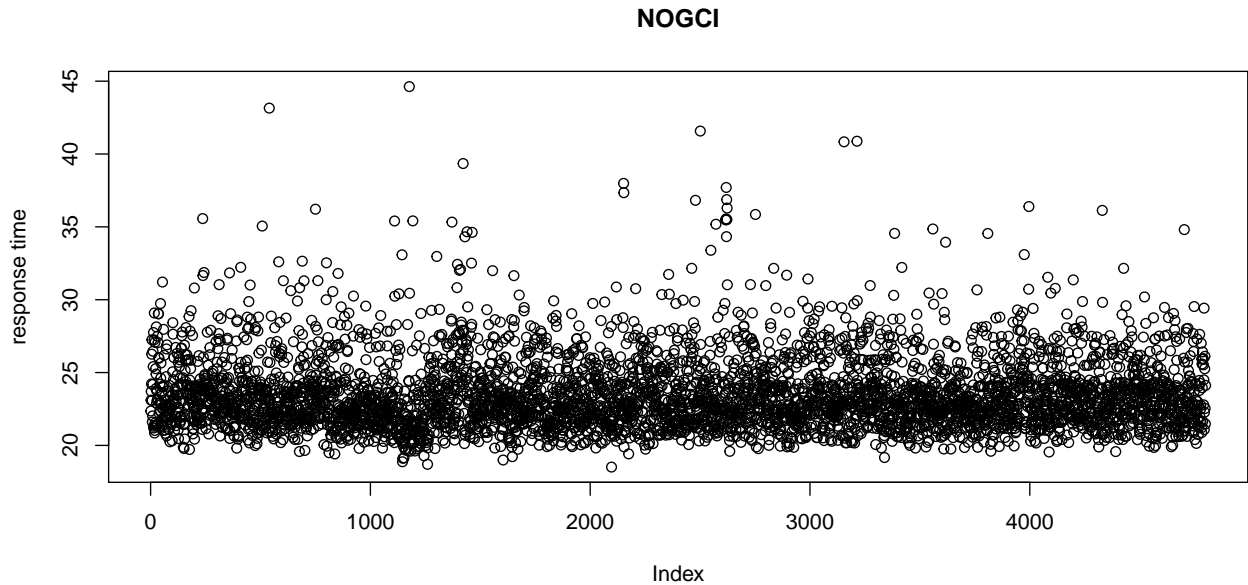
Essas valores de cauda da diferença entre o tempo de resposta e o tempo de serviço me deixou com pulgas atrás da orelha. Não pretendo perder tempo nisso agora, pois vou tocar os passos já combinados. Tenho suspeitas que possa ser alguma coisa envolvendo a lib http do Go somado a alguma outra fonte de impacto.

GCI: plot response time



Este plot é apenas para evidenciar que não há mais as anomalias que antes identificamos.

NOGCI: plot response time



Este plot é apenas para evidenciar que não há mais as anomalias que antes identificamos.