

# GCI Impact Simulated

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

*october, 2019*

## Pergunta geral: qual o impacto do GCI quando utilizado no contexto de computação serverless?

### Descrição das simulações:

- Tempo de experimento simulado: 2 horas e 30 minutos
- Número de repetições: 5
- Tamanho da amostra por simulação: aproximadamente 2.250.000 requisições
- Tamanho da amostra unindo as simulações: aproximadamente 11.250.000 requisições
- Warmup aplicado aos arquivos de entrada da simulação: 500
- Lambda da distribuição Poisson: 20

**Em relação ao tempo de resposta e throughput: a fim de evitar o impacto do coletor de lixo, há melhoria no tempo de resposta mesmo ao submeter uma requisição a possíveis vários redirecionamentos e um cold-start?**

Apesar de não se ter notado impacto no throughput durante as simulações (que pode ser observado na tabela de visualização presente na resposta da segunda pergunta), notou-se um impacto positivo do GCI, contribuindo para um melhor valor de tempo de resposta nos percentis não afetados por coldstart.

### Analizando estatisticamente o tempo de resposta dos resultados

Na análise estatística utilizando intervalo de confiança, temos o objetivo de identificar diferenças estatísticas em medidas da amostra (tais como média, mediana e percentis). Uma diferença estatística pode ser identificada ao notar a inexistência de interseções nos intervalos de confiança de uma dada medida. Caso haja uma interseção entre os intervalos comparados, não podemos afirmar que há diferença estatística entre os dois valores observados.

## [1] "TABELA 1: CONFIDENCE INTERVAL FOR RESPONSE TIME IN MS"

##	stats	nsc.gci.cis	nsc.nogci.cii	opsc.gci.cis	opsc.nogci.cii
## 1	avg	23.00	24.00	22.00	23.00
## 2	p50	20.97	22.52	20.98	22.48
## 3	p95	26.05	28.01	26.10	27.98
## 4	p99	30.05	31.84	30.03	31.59
## 5	p999	40.24	43.93	38.29	40.56
## 6	p9999	3030.00	2989.00	51.26	62.54
## 7	p99999	3034.00	3493.00	2983.00	68.21
## 8	dist	3013.00	3470.00	2962.00	45.72
##	opgsc.gci.cis	opgsc.nogci.cii			
## 1		22.00	23.00		
## 2		20.97	22.48		
## 3		26.07	27.98		

## 4	29.99	31.59
## 5	38.03	40.56
## 6	48.15	62.54
## 7	67.25	68.21
## 8	46.28	45.72

##	stats	nsc.gci.cii	nsc.gci.cis	nsc.nogci.cii	nsc.nogci.cis
## 1	avg	22.00	23.00	24.00	24.00
## 2	p50	20.97	20.97	22.52	22.53
## 3	p95	26.03	26.05	28.01	28.03
## 4	p99	29.96	30.05	31.84	31.90
## 5	p999	39.80	40.24	43.93	44.63
## 6	p9999	2982.00	3030.00	2989.00	2989.00
## 7	p99999	3034.00	3034.00	3493.00	3493.00
## 8	dist	3013.00	3013.00	3470.00	3470.00

##	stats	opsc.gci.cii	opsc.gci.cis	opsc.nogci.cii	opsc.nogci.cis
## 1	avg	22.00	22.00	23.00	23.00
## 2	p50	20.98	20.98	22.48	22.49
## 3	p95	26.08	26.10	27.98	28.00
## 4	p99	29.94	30.03	31.59	31.69
## 5	p999	37.94	38.29	40.56	41.14
## 6	p9999	49.02	51.26	62.54	63.40
## 7	p99999	2964.00	2983.00	68.21	68.25
## 8	dist	2943.00	2962.00	45.72	45.76

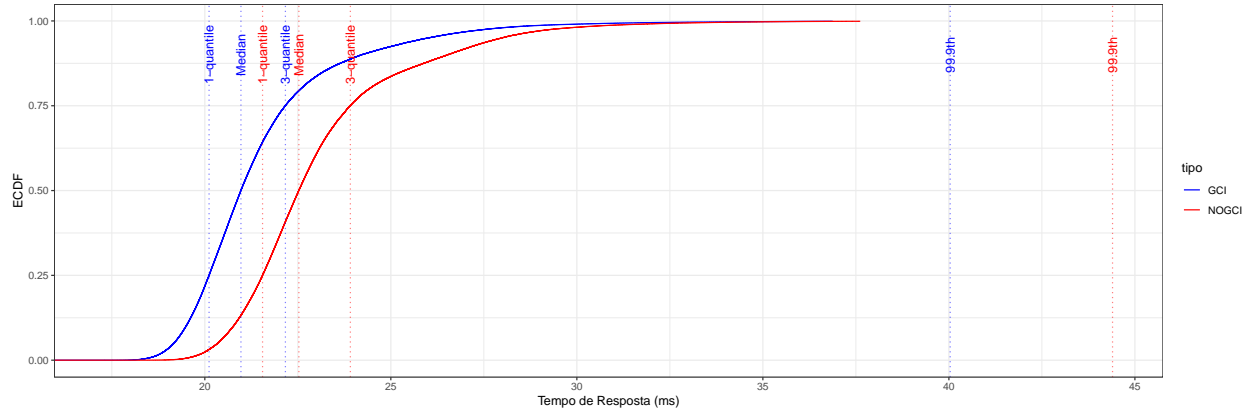
##	stats	opgsc.gci.cii	opgsc.gci.cis	opgsc.nogci.cii	opgsc.nogci.cis
## 1	avg	22.00	22.00	23.00	23.00
## 2	p50	20.97	20.97	22.48	22.49
## 3	p95	26.05	26.07	27.98	28.00
## 4	p99	29.91	29.99	31.59	31.69
## 5	p999	37.71	38.03	40.56	41.14
## 6	p9999	47.29	48.15	62.54	63.40
## 7	p99999	59.66	67.25	68.21	68.25
## 8	dist	38.70	46.28	45.72	45.76

Na tabela 1 referente ao cenário com o escalonador normal (normsched.gci.cis normsched.nogci.cii), é possível notar que temos diferença estatística até o percentil 99.9. Os demais percentis (acima de 99.9) não serão considerados pois sofreram impacto do coldstart. Também na tabela 1, o cenário relacionado com o escalonador ótimo que não sabe do GCI (opsched.gci.cis opmsched.nogci.cii), é possível notar que temos diferença estatística até o percentil 99.99. O percentil 99.999 não será considerado pois sofreu impacto do coldstart. Por fim, ainda na tabela 1 e referente ao cenário com o escalonador ótimo que sabe da presença do GCI (opgcsched.gci.cis opgcimsched.nogci.cii), é possível notar que temos diferença estatística até o percentil 99.999. Assim, podemos estudar todo o espectro de percentis da distribuição do tempo de serviço para este caso.

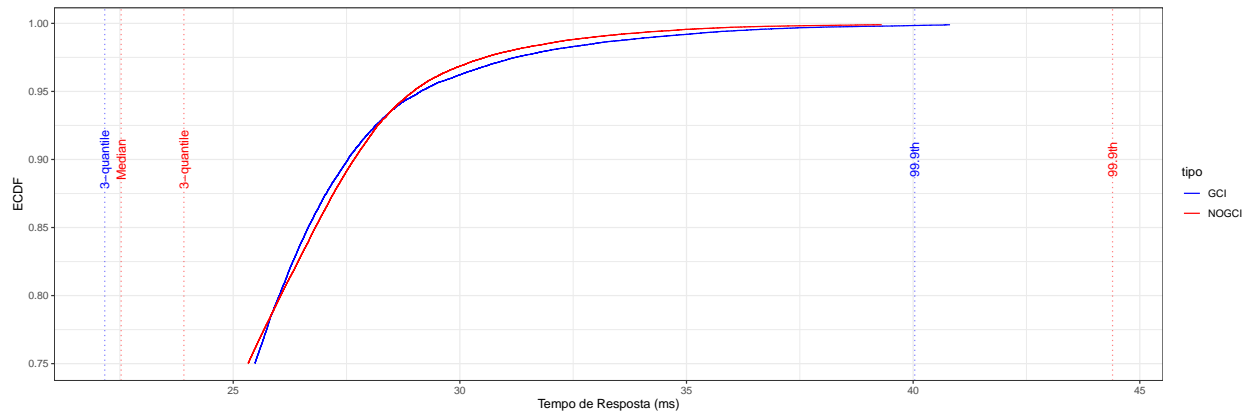
## Análise do tempo de resposta utilizando ECDFs

É possível notar que o GCI obteve melhor desempenho através de toda a distribuição do tempo de serviço de sua função ECDF. A seguir, as ECDFs para cada configuração de escalonador, comparando o cenário com GCI (intitulado GCI) e o sem GCI (intitulado NOGCI).

```
## [1] "FIGURA 1: ECDF RESPONSE TIME - NORM SCHEDULER - 0-4 quantile"
```

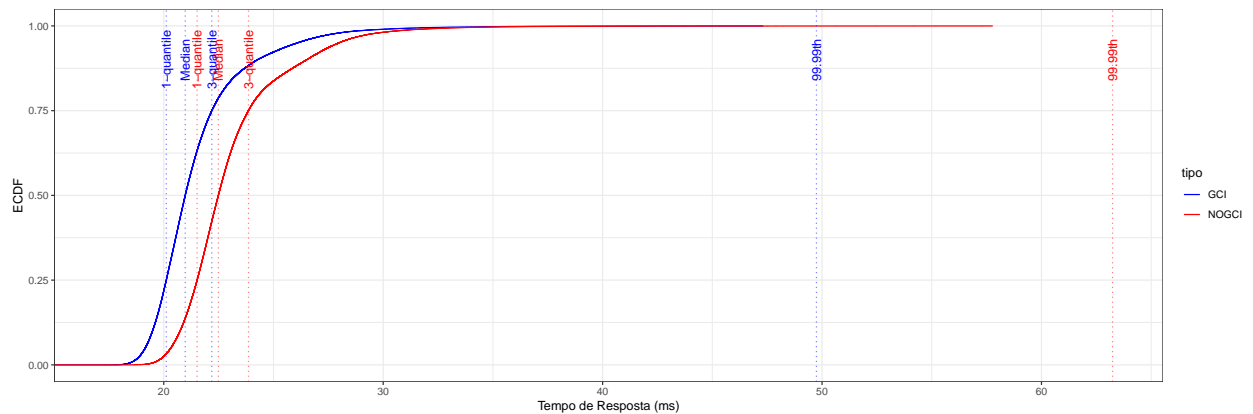


## [1] "FIGURA 1: ECDF RESPONSE TIME - NORM SCHEDULER - 3-4 quantile"

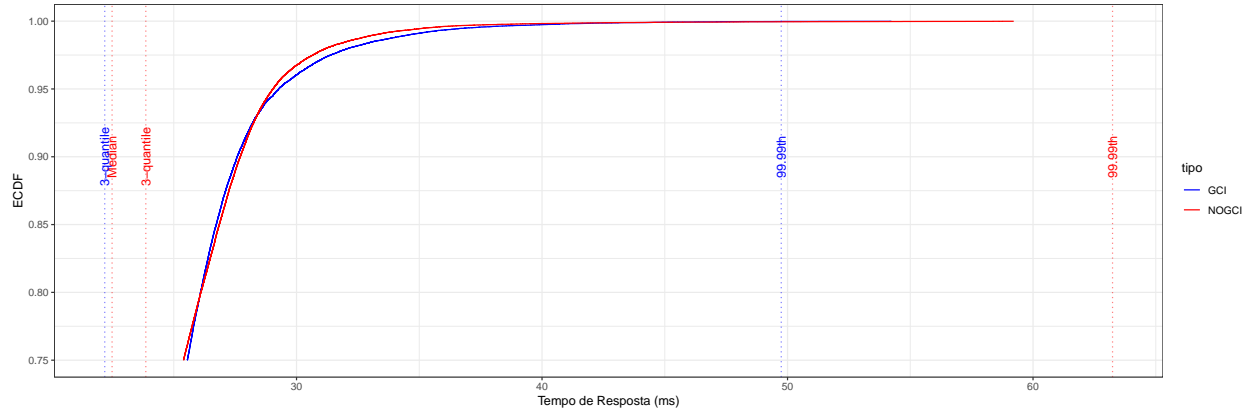


Na Figura 1, não podemos observar os percentils 99.99th pois estes foram impactados pelo coldstart e, por isto, tornaram-se outliers.

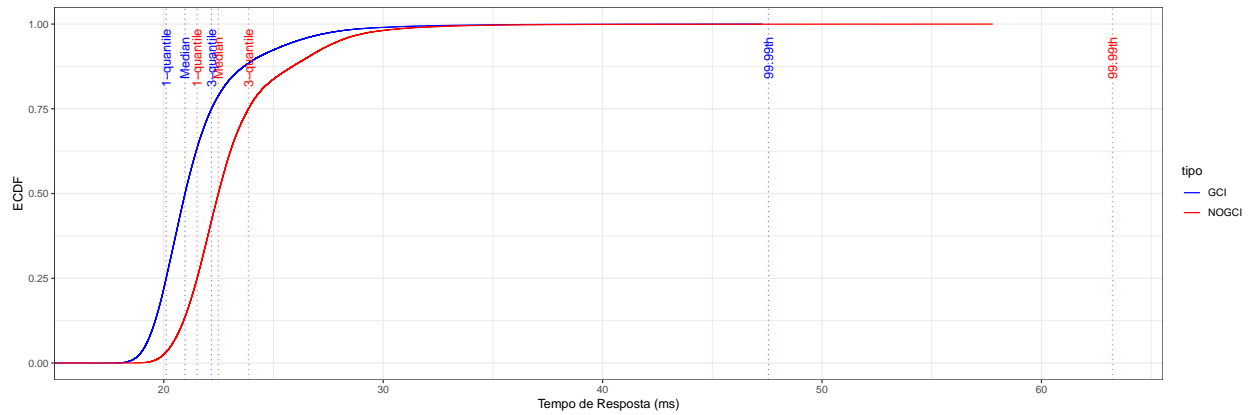
## [1] "FIGURA 2: ECDF RESPONSE TIME - OP SCHEDULER - 0-4 quantile"



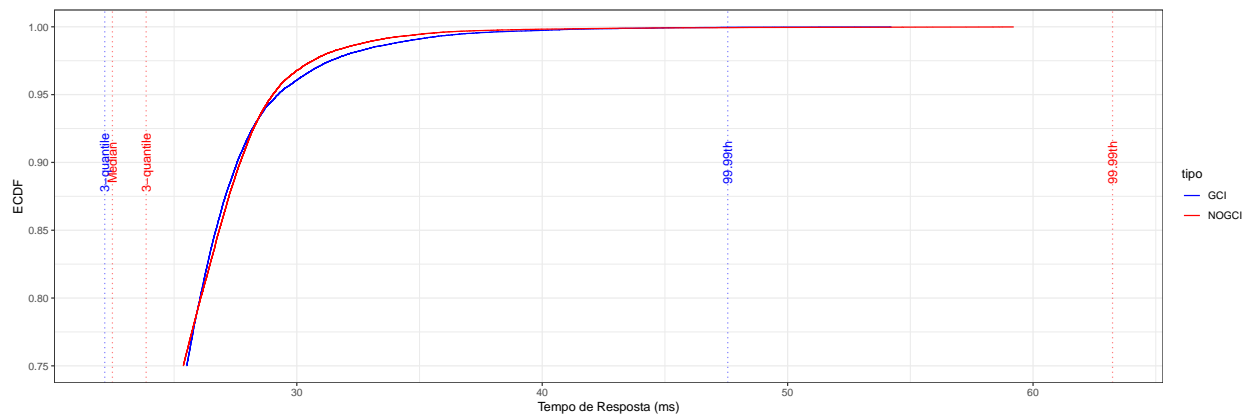
## [1] "FIGURA 2: ECDF RESPONSE TIME - OP SCHEDULER - 3-4 quantile"



## [1] "FIGURA 3: ECDF RESPONSE TIME - OP GCI SCHEDULER - 0-4 quantile"



## [1] "FIGURA 3: ECDF RESPONSE TIME - OP GCI SCHEDULER - 3-4 quantile"



Já nas Figuras 2 e 3, quando observamos o GCI, notamos que o percentil 99.99th diminui quando o escalonador ótimo tem ciência do GCI. Por fim, de modo geral, podemos afirmar que a partir das ECDFs o GCI diminui o tempo de serviço ao longo de toda a ECDF, quando comparado ao NOGCI.

### Análise do tempo de resposta através de tabelas de comparação

No caso do escalonador normal, a redução do tempo de serviço quando utilizado o GCI chega até 10% no percentil 99.9. No caso do escalonador ótimo para provisionamento não ciente do GCI, utilizar o GCI diminui

o tempo de resposta em até 22% no percentil 99.99. Por fim, no caso do escalonador ótimo considerando a presença do GCI, fazer uso do GCI diminui o tempo de resposta em até 32% no percentil 99.99. A tabela a seguir apresenta a comparação entre GCI e NOGCI através da razão entre o tempo de resposta do caso NOGCI e GCI, isto é, o valor na coluna comparison representa o valor do tempo de resposta do NOGCI dividido pelo tempo de resposta do GCI.

## [1] "TABELA 2: COMPARISON TABLE FOR RESPONSE TIME"

##	normsched	opsched	opgcisched
## 0%	1.053091	1.06025419	1.0602542
## 25%	1.071492	1.06979866	1.0701179
## 50%	1.074245	1.07193250	1.0723926
## 75%	1.078769	1.07540112	1.0760801
## 90%	1.095946	1.09256130	1.0940943
## 95%	1.075877	1.07274818	1.0740243
## 99%	1.062304	1.05503669	1.0562346
## 99.9%	1.109005	1.07532440	1.0794421
## 99.99%	1.002487	1.27128900	1.3296882
## 99.999%	1.151376	0.02298431	1.0527836
## 100%	1.150991	0.02427659	0.9592586

Na tabela 2, é importante ressaltar que o impacto do coldstart no cenário normsched já se apresenta a partir do percentil 99.99 e, por isto, conduzimos nossas análises do tempo de resposta para o escalonador normal apenas até o percentil 99.9. Também tabela 2, no cenário opsched observamos que o impacto do coldstart já não se faz presente para o caso NOGCI, conforme esperado. Contudo, a partir do percentil 99.999 o GCI sofre impacto de coldstart e, por isto, conduzimos as análises relacionadas ao escalonador ótimo (não considerando o GCI) até o percentil 99.99. Por último, na tabela 3 atendo-se para o cenário opgcisched, uma vez que por construção o modelo de escalonador utilizado não gera impacto do coletor de lixo, podemos utilizar todos os percentils para conduzir a análise.

### Em relação à eficiência: a fim de evitar o impacto do coletor de lixo, há prejuízo no uso eficiente dos recursos computacionais ao levantar novas instâncias para dar suporte às redistribuições de requisições?

Como esperado, o cenário com escalonador normal, quando comparado aos demais escalonadores, possui o maior custo e menor eficiência dos recursos, tanto para com GCI quanto sem ele. Se compararmos para este escalonador a configuração com GCI e sem GCI, notamos que o GCI aumenta o custo e em 4% e aumenta a eficiência em aproximadamente 13%. Com relação ao cenário utilizando GCI e os escalonadores ótimos, não conseguimos observar variação significativa com relação ao custo e eficiência. Contudo, Podemos notar que houve um impacto negativo do GCI quando comparado ao caso ótimo sem GCI, onde o aumento no custo foi de aproximadamente 12% e a redução da eficiência foi de aproximadamente 26%. Abaixo, uma tabela para visualização das métricas extraídas das simulações. Os valores presentes nesta tabela é uma média dos valores extraídos das simulações.

##	scenario	scheduler_name	throughput	instances_cost	instances_efficiency
## 1	gci	normscheduler	49.9931112	74330.886996	0.0300054484
## 2	gci	opscheduler	50.0049332	30944.083496	0.27586019634
## 3	gci	opgcischeduler	50.0060222	31134.03881	0.27672200802
## 4	nogci	normscheduler	50.0071776	71713.105654	0.02624214418
## 5	nogci	opscheduler	50.0035554	27535.794948	0.37304896054