

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?

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, notou-se um impacto positivo do GCI, contribuindo para um melhor valor de tempo de resposta nos percentis não afetados por coldstart. 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.

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%.

Simulation results: GCI vs NOGCI

```
## Warning: package 'dplyr' was built under R version 3.6.1
```

```
## Warning: package 'ggplot2' was built under R version 3.6.1
```

ECDF for response time of all simulation

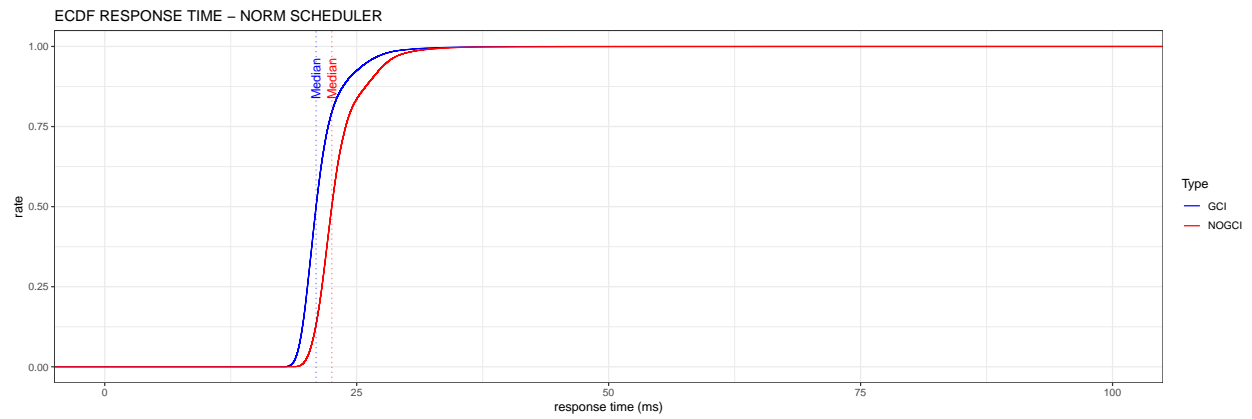
```
## Warning: Removed 1534 rows containing non-finite values (stat_ecdf).
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```

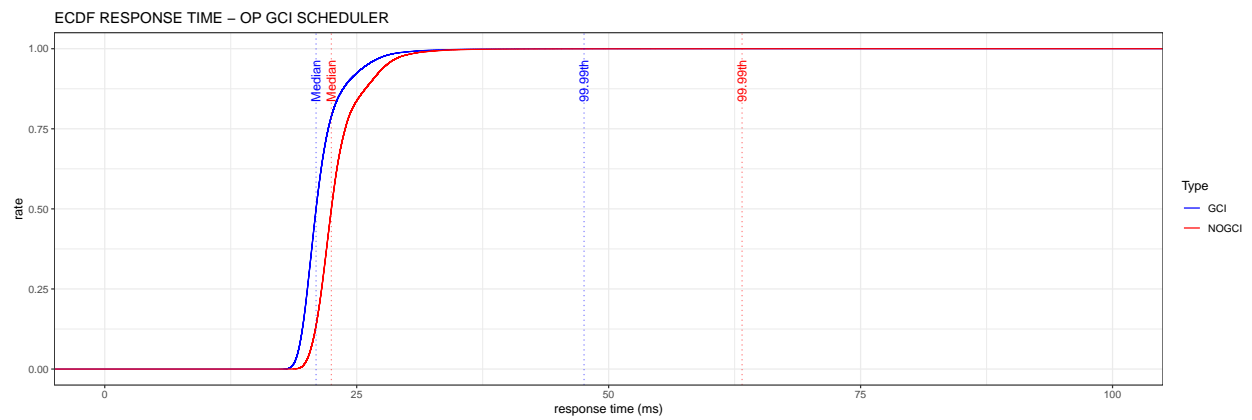
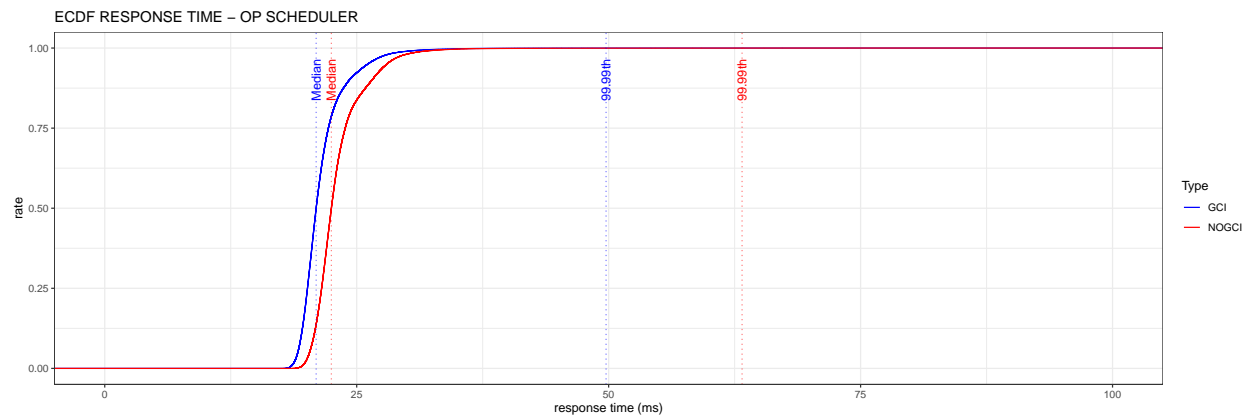
```
## Warning: Removed 1 rows containing missing values (geom_vline).
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```

```
## Warning: Removed 1 rows containing missing values (geom_vline).
```



```
## Warning: Removed 50 rows containing non-finite values (stat_ecdf).
```



summary table of confidence interval for response time, both in milliseconds, of all simulation

```
##      stats gci.cii gci.cis nogci.cii 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	gci.cii	gci.cis	nogci.cii	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	gci.cii	gci.cis	nogci.cii	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

comparison table for response time, of all simulation

##		nogci	gci	comparison
## 0%		18.229	17.310	1.053091
## 25%		21.552	20.114	1.071492
## 50%		22.528	20.971	1.074245
## 75%		23.912	22.166	1.078769
## 90%		26.546	24.222	1.095946
## 95%		28.018	26.042	1.075877
## 99%		31.884	30.014	1.062304
## 99.9%		44.399	40.035	1.109005
## 99.99%		2989.443	2982.027	1.002487
## 99.999%		3492.909	3033.681	1.151376
## 100%		3492.909	3034.696	1.150991

##		nogci	gci	comparison
## 0%		18.353	17.31000	1.06025419
## 25%		21.519	20.11500	1.06979866
## 50%		22.487	20.97800	1.07193250
## 75%		23.861	22.18800	1.07540112
## 90%		26.511	24.26500	1.09256130
## 95%		27.988	26.09000	1.07274818
## 99%		31.630	29.98000	1.05503669
## 99.9%		40.940	38.07223	1.07532440
## 99.99%		63.239	49.74400	1.27128900
## 99.999%		68.246	2969.24261	0.02298431
## 100%		73.696	3035.68200	0.02427659

	nogci	gci	comparison
## 0%	18.353	17.31000	1.0602542
## 25%	21.519	20.10900	1.0701179
## 50%	22.487	20.96900	1.0723926
## 75%	23.861	22.17400	1.0760801
## 90%	26.511	24.23100	1.0940943
## 95%	27.988	26.05900	1.0740243
## 99%	31.630	29.94600	1.0562346
## 99.9%	40.940	37.92700	1.0794421
## 99.99%	63.239	47.55927	1.3296882
## 99.999%	68.246	64.82434	1.0527836
## 100%	73.696	76.82600	0.9592586

Visualizing metrics

	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
##	simulation_exec_time				
## 1			8		
## 2			4		
## 3			4		
## 4			8		
## 5			4		