

memcached analysis

Rafilx

2022-06-19

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

## Loading required package: gridExtra

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##   combine

## Loading required package: viridisLite

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##   date, intersect, setdiff, union
```

R Markdown

Na dissertação, foi relatado que

- 3,1% das requisições de Memcached eram variações de **stats**
- 91,6% das requisições eram **set** ou **get**
- 5,1% das requisições eram malformadas (e.g., requisições HTTP ou SSDP) ou eram **flush_all** (para limpar chaves do cache)

É possível fazer essa análise por trimestre, ou ao menos analisar a incidência de **stats** e **set+get**

```
db <- dbConnect(RSQLite::SQLite(), dbname="../db/database-2022-05-11/mix_protocol.sqlite")
```

```
data_unfetch <-dbSendQuery(db, "
  SELECT *, CAST(CAST(year AS text) || CAST(period AS text) as integer) as year_period, SUBSTR(payload,
    FROM (
      SELECT *, strftime(\"%Y\", tempo_inicio) as year, ((strftime(\"%m\", tempo_final) - 1) / 3) + 1
      FROM MEMCACHED_ANALYSIS
    )
  ")
```

```
data <- fetch(data_unfetch)
```

```
data_memcached_payload_types_unfetch <-dbSendQuery(db, "
  SELECT id, quantity, SUBSTR(payload,0,25) AS payload_limit
  FROM MEMCACHED_PAYLOAD_TYPES
  ")
```

```
## Warning: Closing open result set, pending rows
```

```
data_memcached_payload_types <- fetch(data_memcached_payload_types_unfetch)
```

```
dbDisconnect(db)
```

```
## Warning in connection_release(conn@ptr): There are 1 result in use. The
## connection will be released when they are closed
```

```
data['tempo_final_cast'] = as.POSIXct(data[['tempo_final']], format = "%Y-%m-%d %H:%M:%S")
data['tempo_inicio_cast'] = as.POSIXct(data[['tempo_inicio']], format = "%Y-%m-%d %H:%M:%S")
```

```
memcached_payload_types = data_memcached_payload_types %>%
  mutate(payload_str = toString(payload_limit)) %>%
  arrange(desc(quantity)) %>%
  select('quantity', 'payload_limit', 'id')
```

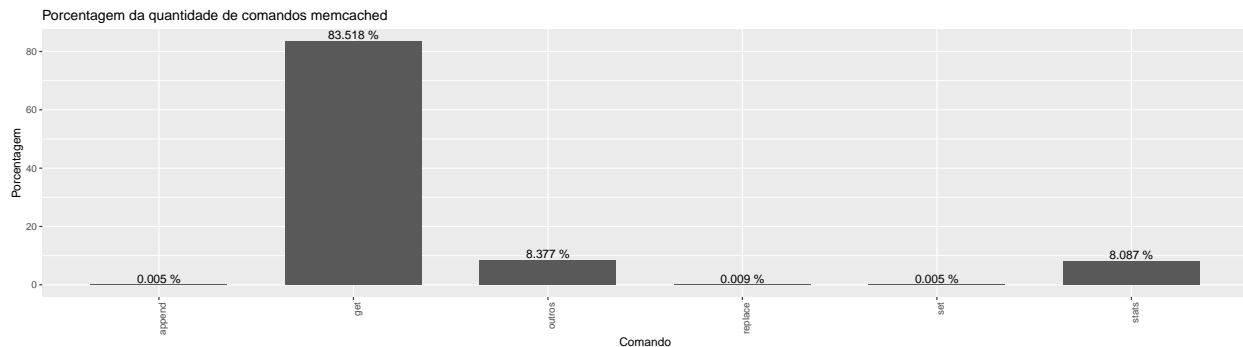
```
memcached_payload_types_quantity_percentage = memcached_payload_types %>%
  mutate(sum_quantity = sum(quantity)) %>%
  mutate(quantity_percentage = (quantity / sum_quantity) * 100)
```

```
memcached_payload_types_quantity_percentage %>%
  select('quantity', 'quantity_percentage', 'payload_limit') %>%
  arrange(desc(quantity)) %>%
  head(15)
```

```
##      quantity quantity_percentage payload_limit
## 1      17826          83.517616         get
## 2       1788           8.377061        outros
## 3       1726           8.086582         stats
## 4          2           0.009370        replace
## 5          1           0.004685          set
```

```
## 6      1      0.004685      append
## 7      0      0.000000      add
## 8      0      0.000000      cas
## 9      0      0.000000      prepend
## 10     0      0.000000      flush_all
```

```
memcached_payload_types_quantity_percentage %>%
  arrange(desc(quantity)) %>%
  filter(quantity > 0) %>%
  select('quantity_percentage', 'payload_limit') %>%
  ggplot(aes(x=payload_limit, y=quantity_percentage)) +
    geom_bar(stat="identity", width = 0.7, position="dodge") +
    geom_text(aes(label = paste(round(quantity_percentage, 3), "%"), vjust = -0.25)) +
    scale_fill_viridis(discrete=TRUE, direction = -1) +
    theme(axis.text.x = element_text(angle = 90, vjust = 1, hjust=1)) +
    ylab("Porcentagem") +
    xlab("Comando") +
    ggtitle("Porcentagem da quantidade de comandos memcached")
```



- Agrupamento realizado por período (trimestre) e “memcached_request_type” é o comando utilizado no ataque ["stats", "set", "get", "add", "cas", "replace", "append", "prepend", "flush_all", "outros"]
- Somando a quantidade de requisições utilizadas por cada comando e período

```
data_grouped_period_command = data %>%
  mutate(year_period_int = year_period,
         year_period = as.factor(year_period),
         command = as.factor(memcached_request_type)) %>%
  group_by(year_period, command) %>%
  summarise(sum_requests_per_attack = sum(requests_per_attack),
            number_of_attacks = n(),
            tempo_inicio=min(tempo_inicio_cast),
            tempo_final=max(tempo_final_cast))
```

```
## 'summarise()' has grouped output by 'year_period'. You can override using the
## '.groups' argument.
```

```
data_grouped_period_command_percentage = data_grouped_period_command %>%
  ungroup() %>%
  group_by(year_period) %>%
```

```

summarise(command = command,
          number_of_attacks = number_of_attacks,
          tempo_inicio = tempo_inicio,
          tempo_final = tempo_final,
          sum_period_number_of_attacks = sum(number_of_attacks),
          sum_period_requests_per_attack = sum(sum_requests_per_attack),
          sum_requests_per_attack = sum_requests_per_attack) %>%
mutate(number_of_attacks_percentage = (number_of_attacks / sum_period_number_of_attacks) * 100,
       number_of_requests_percentage = (sum_requests_per_attack / sum_period_requests_per_attack) * 100)

```

'summarise()' has grouped output by 'year_period'. You can override using the
'.groups' argument.

```

minimum_percentage_as_others = 1
decimals_digits = 1

data_grouped_period_command_others_percentage = data_grouped_period_command_percentage %>%
  mutate(
    command = case_when(
      number_of_requests_percentage < minimum_percentage_as_others ~ "OUTROS",
      TRUE ~ as.character(command)
    )
  ) %>%
  group_by(year_period, command) %>%
  summarise(number_of_requests_percentage = sum(number_of_requests_percentage))

```

'summarise()' has grouped output by 'year_period'. You can override using the
'.groups' argument.

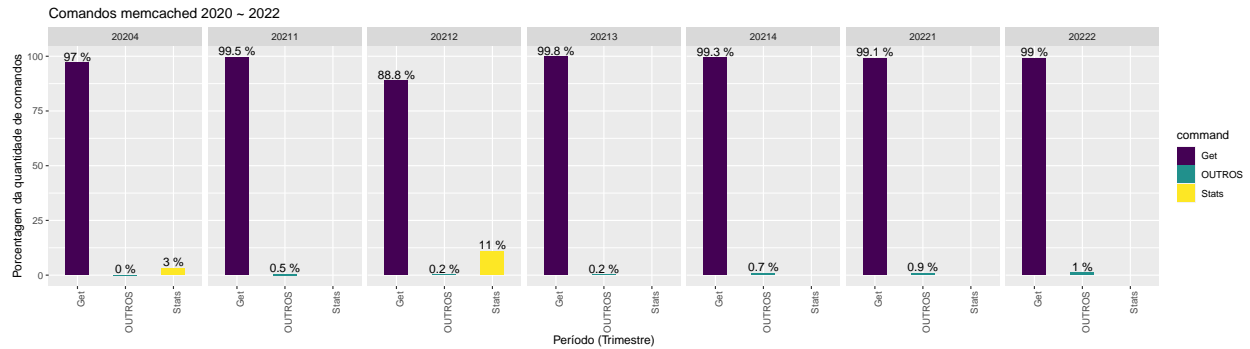
Porcentagem menores que 1 foram agrupadas como “OUTROS”

- Gráfico de barras 2020 ~ 2022

```

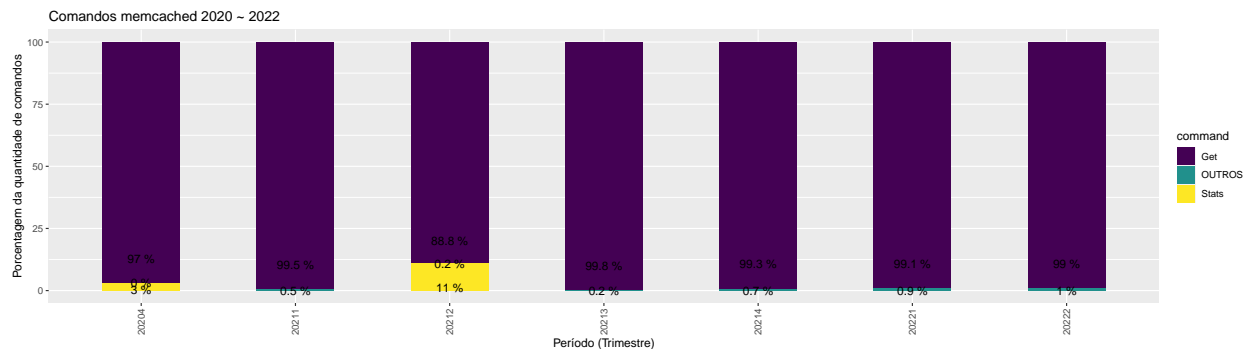
data_grouped_period_command_others_percentage %>%
  ggplot(aes(x=command, y=number_of_requests_percentage, fill=command)) +
  geom_bar(stat="identity", width = 0.5, position="dodge") +
  geom_text(aes(label = paste(round(number_of_requests_percentage, decimals_digits), "%"), vjust = -5)) +
  scale_fill_viridis(discrete=TRUE) +
  theme(axis.text.x = element_text(angle = 90, vjust = 1, hjust=1)) +
  facet_grid(~year_period) +
  ylab("Porcentagem da quantidade de comandos") +
  xlab("Período (Trimestre)") +
  ggtitle("Comandos memcached 2020 ~ 2022")

```



- Gráfico de barras empilhadas 2020 ~ 2022

```
data_grouped_period_command_others_percentage %>%
  ggplot( aes(x=year_period, y=number_of_requests_percentage, fill=command)) +
  geom_bar(stat="identity", width = 0.5) +
  geom_text(aes(label = paste(round(number_of_requests_percentage, decimals_digits), "%")), position = "bottom",
    scale_fill_viridis(discrete=TRUE) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  ylab("Porcentagem da quantidade de comandos") +
  xlab("Período (Trimestre)") +
  ggtitle("Comandos memcached 2020 ~ 2022")
```



- Gráfico de linhas 2020 ~ 2022

```
data_grouped_period_command_others_percentage %>%
  ggplot( aes(x=year_period, y=number_of_requests_percentage, group=command)) +
  geom_line(size=1.2, aes(color=command)) +
  geom_point(color="red", size=3, aes(color=command)) +
  geom_text(
    aes(label = paste(round(number_of_requests_percentage, decimals_digits), "%")),
    hjust = -0.03, nudge_x = 0.05, nudge_y = -1, angle = -10,
  ) +
  scale_fill_viridis(discrete=TRUE) +
  theme(
    axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
  ) +
  ylab("Porcentagem da quantidade de comandos") +
  xlab("Período (Trimestre)") +
  ggtitle("Comandos memcached 2020 ~ 2022")
```

