Servidores e bases de dados: :boas práticas

10 mandamentos:

USE a memória RAM de forma eficiente
Carregue apenas as variáveis e observações realmente necessárias. Limpe periodicamente o ambiente usando o coletor de lixo (gc()). Ao trabalhar com grandes volumes de dados, teste suas rotinas completas primeiro com pequenas amostras.

NÃO GUARDE cópias desnecessárias de bases de dados
Cuidado com redundâncias, especialmente em bases de dados grandes e de acesso restrito, como a RAIS e o CadÚnico.

EVITE SALVAR na área de trabalho dos servidores
A área de trabalho é armazenada no disco "C:", que é compartilhado. Se o espaço for esgotado, pode causar travamento do servidor para todos os usuários. Prefira utilizar diretórios na rede e/ou repositórios de código para guardar scripts, resultados e dados (estes exclusivamente na rede).

NÃO UTILIZE seu PC para processar e armazenar dados
Computadores pessoais não oferecem backup, redundância elétrica e isolamento físico. Opte pelos servidores estatísticos.

NÃO ARMAZENE dados restritos em pastas compartilhadas
Assegure-se de que apenas usuários autorizados tenham acesso ao diretório onde estão salvos dados restritos (identificados).

NÃO RETIRE dados restritos da rede do Ipea
Acesse-os unicamente nos servidores/computadores internos, evitando cópias para dispositivos externos.

ESCOLHA o servidor menos sobrecarregado
Identifique os usuários que mais consomem os recursos: Task Manager>Detalhes Adicionais>Usuários e pesquise pelo nome de usuário (R*, B* ou T*) no Webmail ou no Teams.

USE os servidores somente para análise e modelagem de dados
Para internet, intranet, IpeaProjetos use seu PC ou desktop virtual. Evite usos não institucionais, como treinar algoritmos de ML para trabalhos acadêmicos.

NÃO TRANSFIRA bases de dados entre o Rio e Brasília
Evite acessar dados em storages de Brasília a partir de servidores do Rio e vice-versa para otimizar o tráfego de rede.

PROCESSE dados com eficiência
Para linguagem R, recomendamos o uso de data.table, arrow, DuckDB ou SGBD-SQL, conforme o benchmark ná página seguinte.

***O descumprimento dos mandamentos 2, 4 e 5 pode acarretar consequências legais, previstas na Lei Geral de Proteção de Dados (LGPD).**

Servidores estatísticos

Servidores de alta capacidade com softwares para processamento de dados:

NOME	MEMÓRIA (GB)	CPU (GHZ)	SOFTWARES ESTATÍSTICOS
bsb_stat1	512	2,30	r, python e stata
bsb_stat2	512	2,30	r, stata e debeaver
bsb_stat3	512	2,30	r, stata e dbeaver
bsb_stat4	512	3,80	r, python
rio_stat1	256	3,80	r, stata e dbeaver

Como acessar? 1) Solicitar acesso por e-pedidos de TI; 2) Estar na rede-Ipea (PC ou conexaoVPN); 3) Acesso remoto ao servidor.

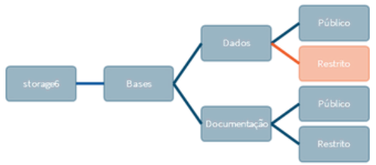
Bases de dados

Quais são as bases disponíveis? Veja o catálogo na Intranet.

Storage	Localização*	Porta
Storage6	Brasília	-
BSB_MSSQL (SQL)	Brasília	1433
SRJN4	Rio de Janeiro	-

*Os *storages* devem ser acessados por meio de servidores do mesmo local (Brasília ou Rio).

Arquitetura das pastas de dados (storage6)
As versões originais das bases de dados ficam na pasta storage6/dados. Para dados sigilosos, explore a documentação mesmo antes de solicitar acesso (Saiba como solicitar acesso a bases restritas).



Manipulação e modelagem: :pacotes recomendados

Benchmark:

Simulação com dados da RAIS vínculos 2004 (44 milhões de linhas): leitura, tabulação de empregados por setor e UF, e estimação de um modelo de regressão minceriano (ols, iv, e fe).

Leitura e tabulação por setor e UF (44 milhões de observações)*

pacote	dados	Mínimo (s)	Mediana (s)
arrow	parquet	18.78	19.52
duckdb	parquet	18.43	20.06
sgbd	mssql	20.78	21.04
data.table	csv	46.58	58.87
dplyr	csv	123	153

*100 iterações: leitura, tabulações e estimação de um modelo de regressão.

Estimação de equação minceriana (400 mil observações)

pacote	Padrão (s)	efeito fixo * (s)	IV (s)
fixest	1.24	1.26	4.02
lfe	2.81	4.07	6.03
lm (base r)	2.82	305	--

*Efeitos fixos para 561 CNAEs.

Os resultados completos estão no GIT do IpeaDATA-lab.

- Sintaxe do dplyr.
- Não é necessário subir bases completas para a memoria.

Saiba mais:
arrow.apache.org/docs/r

Ex. Calculando o número de vínculos a RAIS por UF:

```
library(tidyverse)
library(arrow)

#Leitura dos dados "fora da memória"
 dados <- open_dataset(PATH_RAIS_PARQUET)

#Número de empregados por UF
 tab_uf <- dados |>
  count(uf, name = "num_empregados")

#Retornar resultado
 tab_uf <- tab_uf |> collect()
```

- DuckDB + dplyr
- Sintaxe do dplyr ou SQL
- Não é necessário subir bases completas para a memoria.

Saiba mais:
github.com/tidyverse/duckplyr

Ex. Computando vínculos formais da RAIS por CNAE:

```
library(tidyverse)
library(duckplyr)

#Leitura dos dados "fora da memória"
 dados <- duckplyr_df_from_parquet(PATH_RAIS_PARQUET)

#Número de empregados por CNAE
 tab_cnae <- dados |>
 count(clas_cnae10, name = "num_empregados")

#Retornar resultado
 tab_cnae <- tab_cnae |> collect()
```

- **MS SQL Server + dplyr**
- **Sintaxe do dplyr ou SQL**

Saiba mais:
github.com/tidyverse/dbplyr

Ex. Computando renda média por UF:

```
library(DBI)

# Conectar ao servidor
con_mssql <- dbConnect(odbc::odbc(),
                       .connection_string = "driver={SQL Server}",
                       Server   = "bsb_mssql", # nome do servidor
                       Trusted_Connection = "Yes",
                       database = "RAIS" # base de dados)
```

```
library(tidyverse)
library(dbplyr)

tab_rem_uf <- tbl(con_mssql, "tb_vinculos_2021") |>
 group_by(uf) |>
 summarise(rem_uf = mean(rem_med_r)) |>  collect()
```