

foreach vs for

Miguel Carvalho Nascimento

Comparação de Desempenho

Usando o pacote microbenchmark para comparar:

```
# Carregar pacotes necessários  
library(foreach)  
library(doParallel)  
library(microbenchmark)
```

Criando função para testes de desempenho nos loops:

```
# Função para ser usada nos loops  
sum_function <- function() {  
  sum(runif(1e6))  
}
```

Função com for

```
# Loop usando for  
for_loop <- function() {  
  result <- numeric(100)  
  for (i in 1:100) {  
    result[i] <- sum_function()  
  }  
  return(result)  
}
```

Loop com “foreach”

Configurando a paralelização

```
# Configurar paralelizacao
num_cores <- detectCores() - 1
cl <- makeCluster(num_cores)
registerDoParallel(cl)
```

foreach

```
foreach_loop <- function() {
  result <- foreach(i = 1:100, .combine = c,
    .export=c("for_loop","sum_function")) %dopar% {
    sum_function()
  }
  return(result)
}
```

Comparação

```
# Comparação de desempenho usando microbenchmark
benchmark_results <- microbenchmark(
  for_loop = for_loop(),
  foreach_loop = foreach_loop(),
  times = 10
)

benchmark_results
```

```
Unit: milliseconds
      expr      min       lq     mean   median      uq      max  neval
for_loop 2140.5891 2193.5702 2215.734 2205.7704 2217.6906 2371.2547    100
foreach_loop 638.2154 698.3917 707.725 705.3725 721.6924 763.9755    100
cld
a
b
```

Encerrando a paralelização

```
stopCluster(cl)
```

Colunas Explicadas

- **Unit:** microseconds: A unidade de tempo usada para medir as expressões é microsegundos (1 microsegundo = 10^{-6} segundos).
- **expr:** A expressão sendo avaliada.
- **min:** O menor tempo de execução observado entre todas as execuções.
- **lq (lower quartile):** O valor do primeiro quartil, ou seja, 25% das execuções foram concluídas em menos tempo que este valor.
- **mean:** A média dos tempos de execução.
- **median:** O valor mediano dos tempos de execução, ou seja, 50% das execuções foram concluídas em menos tempo que este valor.
- **uq (upper quartile):** O valor do terceiro quartil, ou seja, 75% das execuções foram concluídas em menos tempo que este valor.
- **max:** O maior tempo de execução observado entre todas as execuções.
- **neval:** O número de execuções realizadas para cada expressão.