# Go Gopher
## NPB - EP em Golang

TEC VII
Andrei Majada e Caroline Evangelista

# Estratégias de paralelização

**Goroutines** - Funciona como um thread onde você pode gerar uma nova goroutine para executar o código simultaneamente usando a palavra-chave go.

**WaitGroup** - Aguarde a conclusão de várias goroutines. Possui três funções: Add(), Done() e Wait().

**Channels** - Permite que as goroutines sejam sincronizadas assim que todas as goroutines concluírem sua computação. Obriga a thread principal a aguardar a goroutine.

# EP

O **kernel EP** consiste na geração de um número determinado de pares de desvios gaussianos aleatórios que pode ser feita de forma independente.

Ao fim do laço, para cada processo, as variáveis sx e sy contém a soma dos desvios gaussianos gerados, e o vetor q contém a quantidade de pares dentro do square annulus.

É feita então a soma das variáveis e do vetor entre todos os processos para obter os valores finais, que são validados pelo processo raiz.

# Trechos de código

```
// Channels sends and receives block until the other side is ready.
// The main thread can terminate even before executing our goroutine.
// Buffered channels accept a limited number of values without a corresponding receiver for those values.
operationResult := make(chan Operations,np)

//To wait for multiple goroutines to finish, we can use a wait group.
var awaitOP sync.WaitGroup

//Time package contains both a wall clock reading and a monotonic clock reading
//The general rule is that the wall clock is for telling time and the monotonic clock is for measuring time.
initialTime := time.Now()

//Add np number to the WaitGroup counter. If the counter becomes zero, all goroutines blocked on Wait are released.
awaitOP.Add(np)

// Each instance of this loop may be performed independently. We compute
// the k separately to take into account the fact that some nodes
// have more numbers to generate than others
for k := 1; k <= np; k++{
  //Goroutine is like a thread where you can spawn a new goroutine to run code simultaneously using the go keyword:
  go func(k int){
    //Each Done() function decreases the counter by 1.
    defer awaitOP.Done()
    var SX, SY float64
    var t1,t2,t3,t4,x1,x2 float64
```

# Trechos de código

```
//A sender can close a channel to indicate that no more values will be sent.
close(operationResult)
//The Wait() function blocks the code and it will be released until the counter is zero.
awaitOP.Wait()
```

# Resultados

**Configurações da máquina de execução:**

- **Processador:** intel i7-8565U CPU / 1.80GHz / 4 cores / 8 threads
- **RAM:** 16 GB DDR4 3200mhz

# Resultados

**Mops:**

- **Operações de memória por segundo, expressando a performance da memória.**
- **Utilizado para determinar a eficiência da RAM.**
- **Pode ser afetado por tudo que estiver acessando a RAM no momento.**

$$Mop/s = 2^{M+1} / tSeconds / 1000000.0$$

```
Mops = pow(2.0, M+1)/tm/1000000.0;
```

# Resultados - S

```
NAS Parallel Benchmark Parallel GO version
Number of random numbers generated: 3.3554432e+07

Benchmark EP Results:
CPU Time = 616.375816ms
N = 24
No. Gaussian Pairs = 1.3176389e+07
Sums = -3247.8346520346163  -6958.407078382828
Counts:
0 - 6.140517e+06
1 - 5.8653e+06
2 - 1.100361e+06
3 - 68546
4 - 1648
5 - 17
6 - 0
7 - 0
8 - 0
Class NPB = S
Total threads = 8
Mop/s total = 54.438268226928614
Operation type = EP - Generation of Random Numbers
Verification = SUCCESSFUL
Compiler Version = go1.18
```

```
Number of random numbers generated:        33554432

EP Benchmark Results:

CPU Time =    1.1801
N = 2^   24
No. Gaussian Pairs =        13176389
Sums =    -3.247834652034551e+03    -6.958407078382874e+03
Counts:
 0        6140517
 1        5865300
 2        1100361
 3          68546
 4           1648
 5             17
 6              0
 7              0
 8              0


EP Benchmark Completed
class_npb       =                         S
Size            =                  33554432
Total threads   =                         1
Iterations      =                         0
Time in seconds =                      1.18
Mop/s total     =                     28.43
Operation type  = Random numbers generated
Verification    =                SUCCESSFUL
Version         =                       4.1
```

Podemos identificar que todos os valores gerados estão dentro da faixa de representação esperada em comparação com o NPB-CPP, porém o código em CPP executa a metade do número de Mops e tem o melhor desempenho.

# Resultados - A

```
...
NAS Parallel Benchmark Parallel GO version
Number of random numbers generated: 5.36870912e+08

Benchmark EP Results:
CPU Time = 8.459190978s
N = 28
No. Gaussian Pairs = 2.10832767e+08
Sums = -4295.875165632603  -15807.32573678602
Counts:
0 - 9.8257395e+07
1 - 9.3827014e+07
2 - 1.7611549e+07
3 - 1.110028e+06
4 - 26536
5 - 245
6 - 0
7 - 0
8 - 0
Class NPB = A
Total threads = 8
Mop/s total = 63.4659878700282
Operation type = EP - Generation of Random Numbers
Verification = SUCCESSFUL
Compiler Version = go1.18
```

```
Number of random numbers generated:       536870912

EP Benchmark Results:

CPU Time =   23.7007
N = 2^   28
No. Gaussian Pairs =       210832767
Sums =     -4.295875165632459e+03    -1.580732573678569e+04
Counts:
 0       98257395
 1       93827014
 2       17611549
 3        1110028
 4          26536
 5            245
 6              0
 7              0
 8              0


EP Benchmark Completed
class_npb       =                         A
Size            =                 536870912
Total threads   =                         1
Iterations      =                         0
Time in seconds =                     23.70
Mop/s total     =                     22.65
Operation type  = Random numbers generated
Verification    =                SUCCESSFUL
Version         =                       4.1
```

Idem do anterior com a faixa de valores do NPB-CPP e o um menor número de Mops.

# Resultados - B

```
...
NAS Parallel Benchmark Parallel GO version
Number of random numbers generated: 2.147483648e+09

Benchmark EP Results:
CPU Time = 35.751348007s
N = 30
No. Gaussian Pairs = 8.43345606e+08
Sums = 40338.15542441945  -26606.69192810556
Counts:
0 - 3.9305847e+08
1 - 3.75280898e+08
2 - 7.0460742e+07
3 - 4.438852e+06
4 - 105691
5 - 948
6 - 5
7 - 0
8 - 0
Class NPB = B
Total threads = 8
Mop/s total = 60.06720774778981
Operation type = EP - Generation of Random Numbers
Verification = SUCCESSFUL
Compiler Version = go1.18
```

```
Number of random numbers generated:      2147483648

EP Benchmark Results:

CPU Time =   94.3142
N = 2^   30
No. Gaussian Pairs =       843345606
Sums =     4.0338155424414O2e+04    -2.660669192810772e+04
Counts:
 0      393058470
 1      375280898
 2       70460742
 3        4438852
 4         105691
 5            948
 6              5
 7              0
 8              0


EP Benchmark Completed
class_npb       =                          B
Size            =                 2147483648
Total threads   =                          1
Iterations      =                          0
Time in seconds =                      94.31
Mop/s total     =                      22.77
Operation type  = Random numbers generated
Verification    =                 SUCCESSFUL
Version         =                        4.1
```

Idem do anterior com a faixa de valores do NPB-CPP e o um menor número de Mops.

# Resultados - C

```
NAS Parallel Benchmark Parallel GO version
Number of random numbers generated: 8.589934592e+09

Benchmark EP Results:
CPU Time = 3m44.977706937s
N = 32
No. Gaussian Pairs = 3.373275903e+09
Sums = 47643.679279943426  -80840.72988043955
Counts:
0 - 1.572172634e+09
1 - 1.501108549e+09
2 - 2.81805648e+08
3 - 1.7761221e+07
4 - 424017
5 - 3821
6 - 13
7 - 0
8 - 0
Class NPB = C
Total threads = 8
Mop/s total = 38.18127008648648
Operation type = EP - Generation of Random Numbers
Verification = SUCCESSFUL
Compiler Version = go1.18      You, 4 seconds ago • Uncommitted cha
```

```
Number of random numbers generated:      8589934592

EP Benchmark Results:

CPU Time =  358.5132
N = 2^   32
No. Gaussian Pairs =      3373275903
Sums =     4.764367927990943e+04    -8.084072988044894e+04
Counts:
 0     1572172634
 1     1501108549
 2      281805648
 3       17761221
 4         424017
 5           3821
 6             13
 7              0
 8              0


EP Benchmark Completed
class_npb       =                        C
Size            =               8589934592
Total threads   =                        1
Iterations      =                        0
Time in seconds =                   358.51
Mop/s total     =                    23.96
Operation type  = Random numbers generated
Verification    =                SUCCESSFUL
Version         =                      4.1
```

Idem do anterior com a faixa de valores do NPB-CPP e o um menor número de Mops.