

PROGRAMAÇÃO PARALELA PROJETO DE EXPERIMENTOS

Marco A. Zanata Alves

O QUE VAMOS VER?

Checklist de experimentos em programação paralela

Como obter resultados significativos

O QUE NÃO VAMOS VER?

Como escolher as melhores métricas

Como analisar resultados

PROGRAMAÇÃO PARALELA

Grande maioria dos trabalhos envolve **experimentos computacionais**

- Utilizam pelo menos uma máquina
- Podem envolver vários computadores

Análise de desempenho por vezes é feita de maneira **superficial**

- As conclusões não são necessariamente portáteis no tempo

OBJETIVO

Tornar os experimentos mais confiáveis

Obter a maior quantidade de informação rodando a menor quantidade possível de experimentos

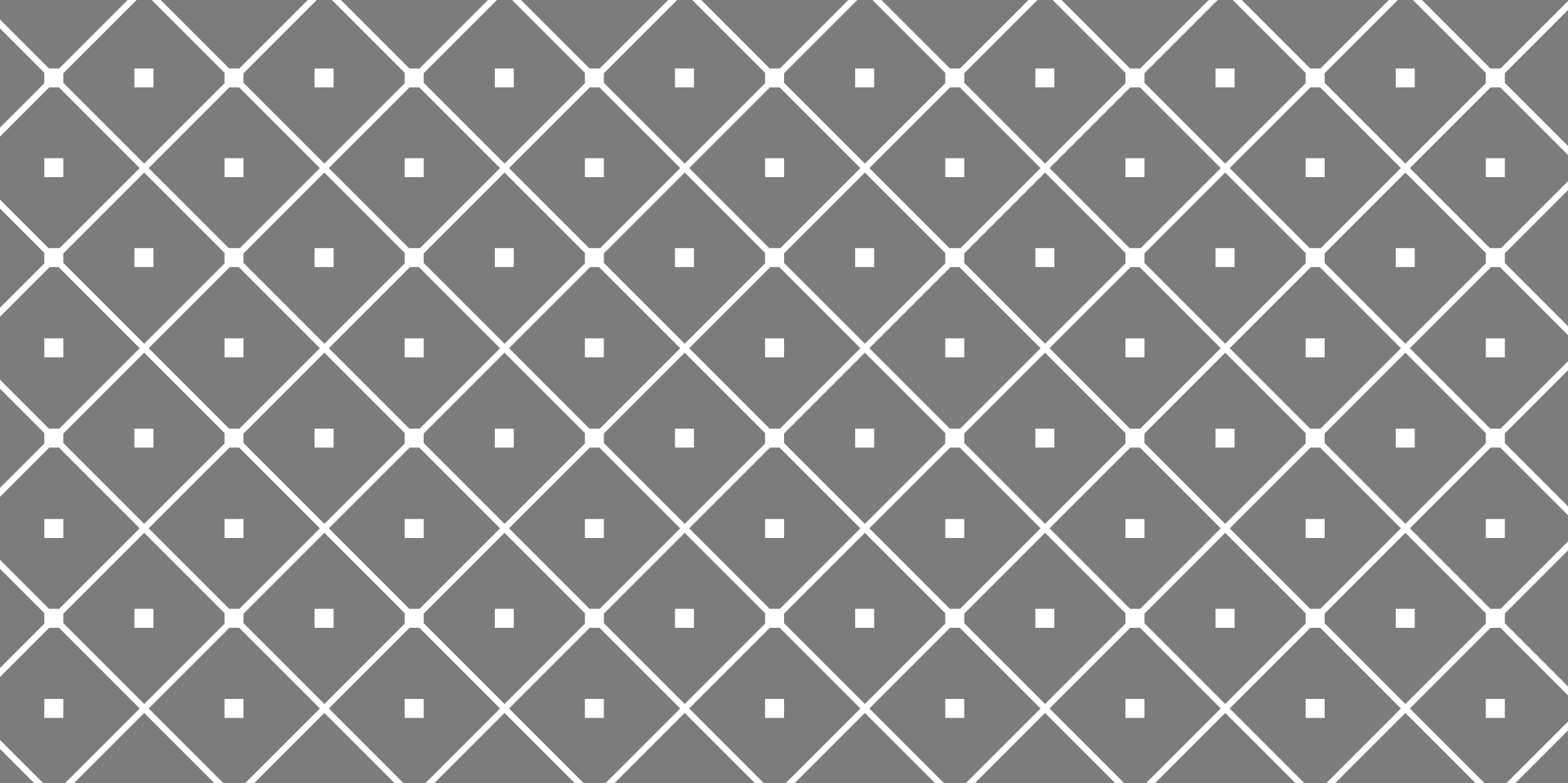
EXPERIMENTO CONFIÁVEL

Confiabilidade → experimentos reprodutíveis

EXPERIMENTO CONFIÁVEL

Para ter confiabilidade:

1. Exercer um controle sobre as variáveis controláveis
2. Registrar o valor das variáveis não controladas (contexto)



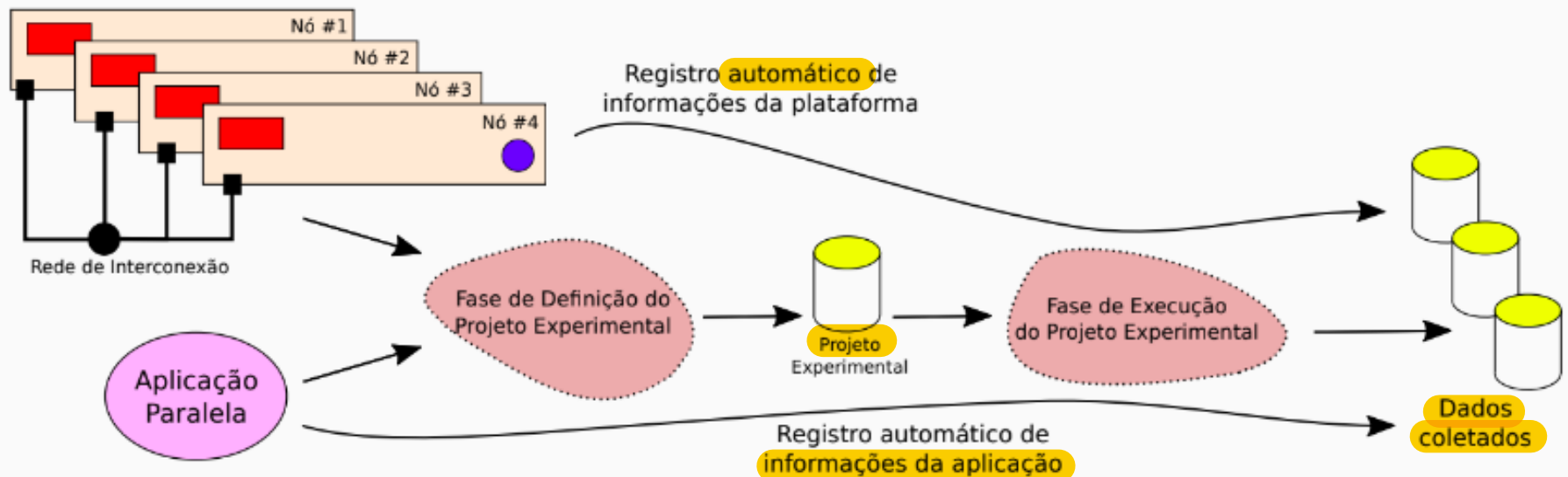
FASES DE UM EXPERIMENTO

VISÃO GERAL

1. Controle e Coleta
2. Análise de Dados

FASE 1 - CONTROLE E COLETA

Mecanismos automáticos guiados por um projeto experimental



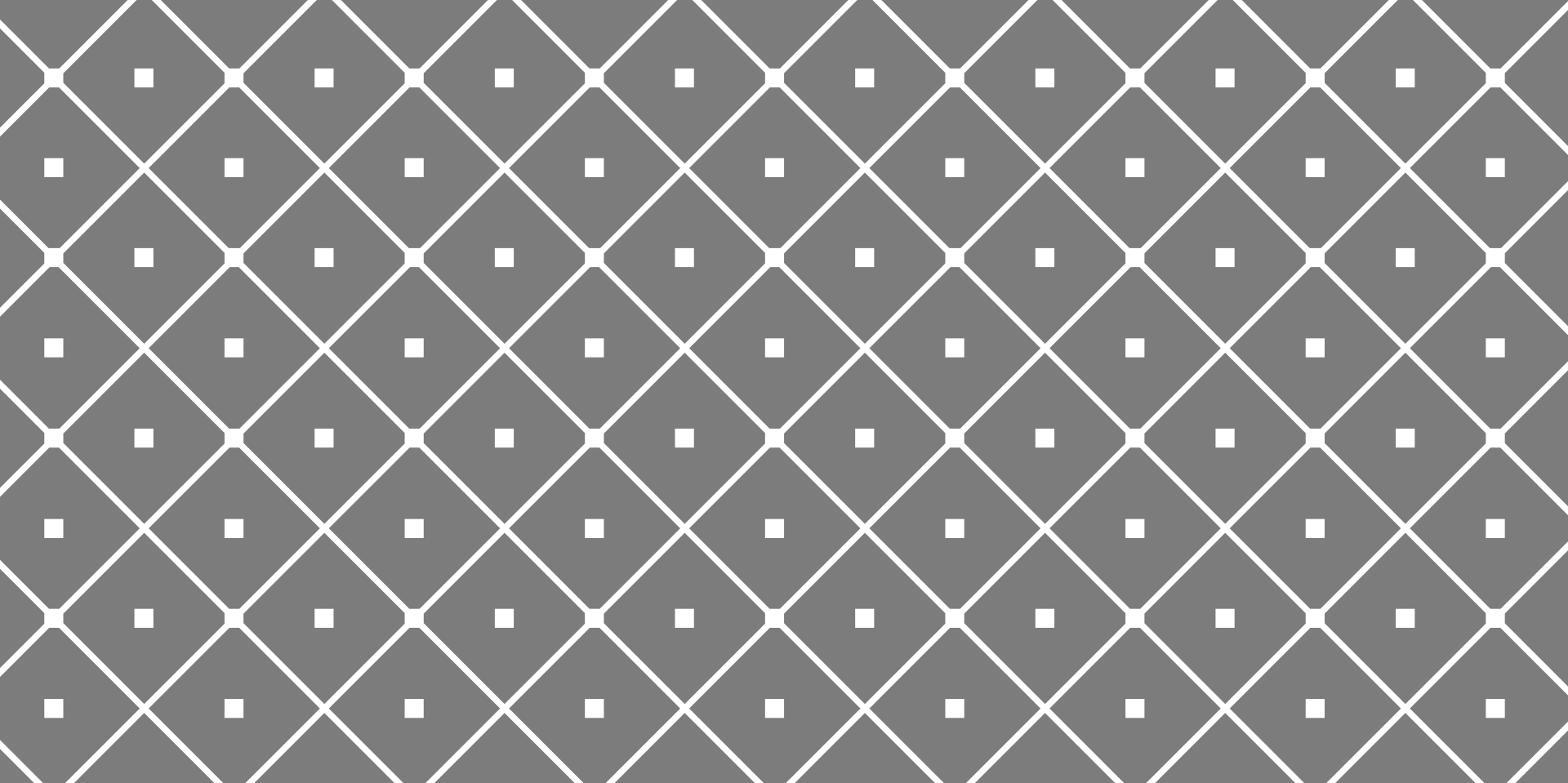
FASE 2 - ANÁLISE

Mecanismos automáticos de tratamento dos dados

- Interpretação dos dados é feita à posteriori

Métricas de desempenho

- Overhead, speedup, eficiência, escalabilidade



FATORES, NÍVEIS E VARIÁVEIS DE RESPOSTA

TERMINOLOGIA DO PROJETO EXPERIMENTAL

Fatores (variáveis de controle)

- tamanho de cache
- número de threads
- número de nós
- latência de rede

TERMINOLOGIA DO PROJETO EXPERIMENTAL

Fatores (variáveis de controle)

Níveis (seus valores)

- tamanho de cache [10M, 20M, 2GB]
- número de threads [2, 4, 8, 16]
- número de nós [2, 4, 8, 16]
- latência de rede [2.67 ms, 2.06 ms, 2.70 ms]



TERMINOLOGIA DO PROJETO EXPERIMENTAL

Fatores (variáveis de controle)

Níveis (seus valores)

- tamanho de cache [10M, 20M, 2GB]
- número de threads [2, 4, 8, 16]
- número de nós [2, 4, 8, 16]
- latência de rede [2.67 ms, 2.06 ms, 2.70 ms]

Variáveis de resposta (observações medidas)

TERMINOLOGIA DO PROJETO EXPERIMENTAL

Fatores (variáveis de controle)

Níveis (seus valores)

- tamanho de cache [10M, 20M, 2GB]
- número de threads [2, 4, 8, 16]
- número de nós [2, 4, 8, 16]
- latência de rede [2.67 ms, 2.06 ms, 2.70 ms]

Variáveis de resposta (observações medidas)

- tempo de resposta
- energia
- número clocks
- quantidade de cache miss

TERMINOLOGIA DO PROJETO EXPERIMENTAL

Exemplo:

Para uma aplicação paralela

Variáveis de resposta: tempo de execução (s)

Fatores: número de processos

Níveis: 1, 2, 4, 8

TERMINOLOGIA DO PROJETO EXPERIMENTAL

Exemplo:

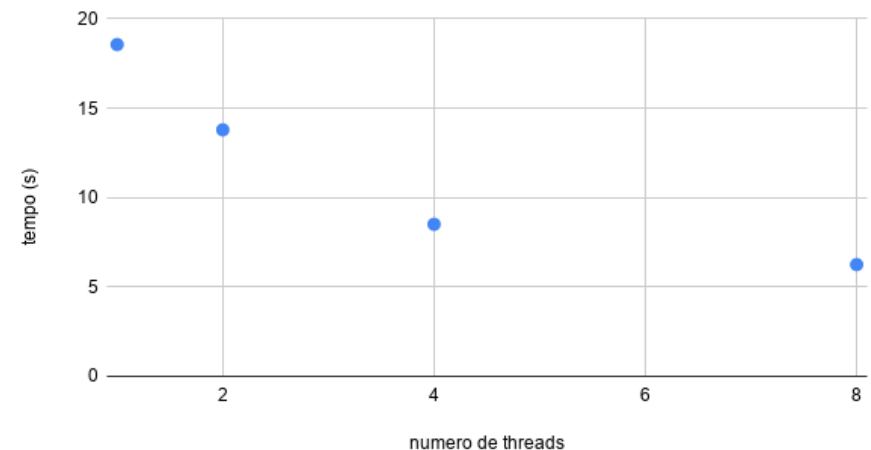
Para uma aplicação paralela

Variáveis de resposta: tempo de execução (s)

Fatores: número de processos

Níveis: 1, 2, 4, 8

kmeans x numero de threads



METODOLOGIA - EXEMPLO

Processor Cores: 8 cores @ 2.0 GHz, 32 nm; 4-wide out-of-order; 16 stages 16 B fetch size; 18-entry fetch buffer, 28-entry decode buffer; 168-entry ROB; MOB entries: 64-read, 36-write; 3-alu, 1-mul. and 1-div. int. units (1-3-32 cycle); 1-alu, 1-mul. and 1-div. fp. units (3-5-10 cycle); 1-load and 1-store units (1-1 cycle); Branch Predictor: 1 branch per fetch; 4 K-entry 4-way set-associ., LRU policy BTB; 48-entry BOB; Two-Level GAs 2-bits; 16 K-entry PBHT; 256 lines, 2048 sets SPHT;

L1 Data + Inst. Cache: 32 KB, 8-way LRU, 64 B line size; 2-cycle; MSHR: 8-request, 10-write-back, 1-prefetch; Stride prefetch: 1-degree, 16-strides;

L2 Cache: Private 256 KB, 8-way LRU, 64 B line size; 4-cycle; MSHR: 4-request, 6-write-back, 2-prefetch; Stream prefetch: 2-degree, 256-streams;

L3 Cache: Shared 20 MB (8-banks), 20-way LRU; 64 B line size; 6-cycle; Bi-directional ring; MOESI coherence protocol; MSHR: 8-request, 12-write-back;

DDR3-1333 Modules: On-core ctrl.; 4 GB; DRAM@166 MHz; 8 KB row buffer 8 DRAM banks; 4-channels; 8 B burst width at 2:1 core-to-bus freq. ratio; Open-row policy; DRAM: CAS, RP, RCD, RAS and CWD latency 9-9-9-24-7 cycles;

gcc: 5.5.0 20171010

flag gcc: -O2

COMO MEDIR

Informações do processador

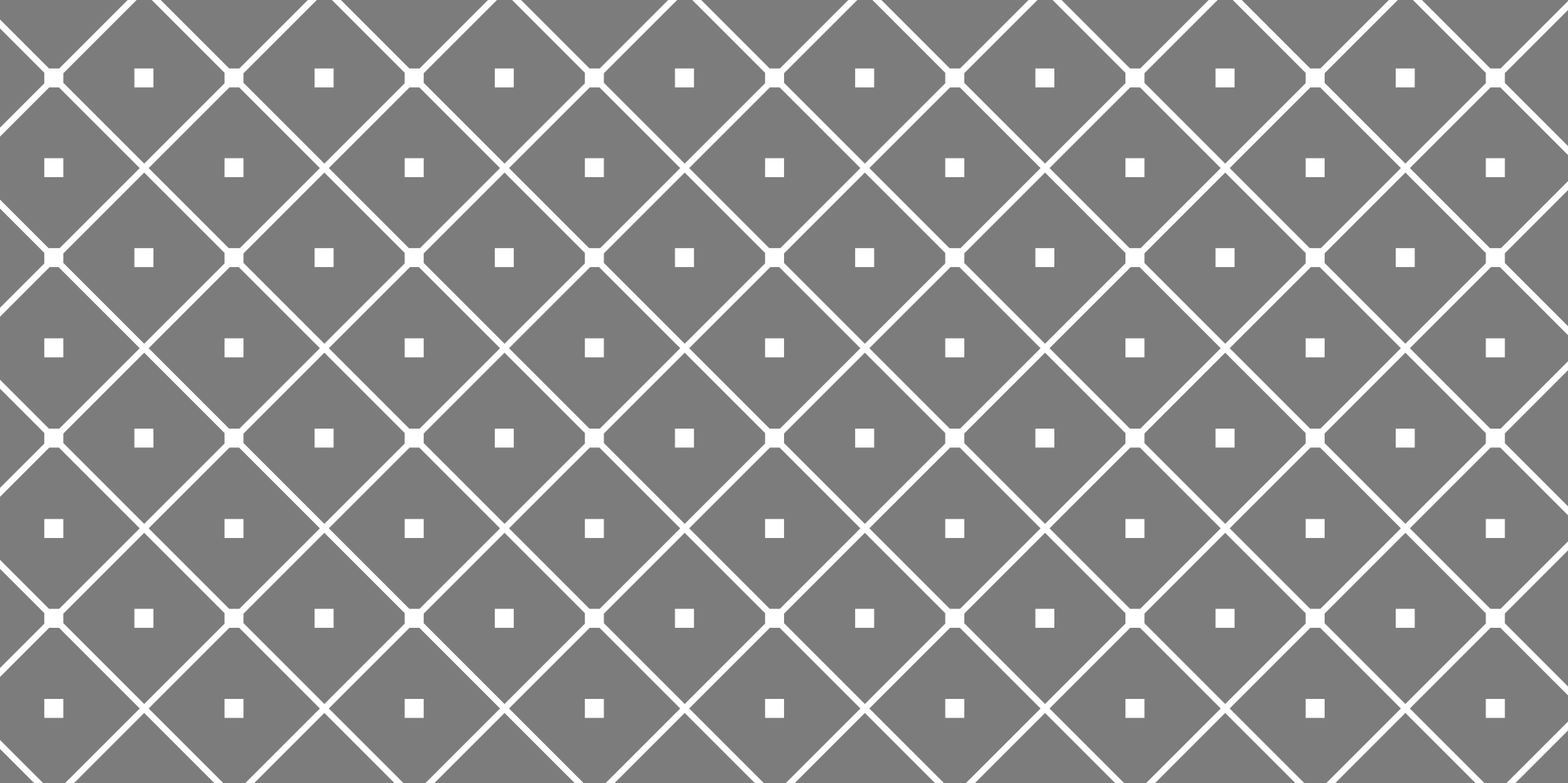
- `lshw`
- `lspci`
- `cat /proc/cpuinfo`
- `lscpu`

gcc:

- `gcc --version`

Flags gcc:

- `-O1, -O2, -O3, -O0, -Ofast`

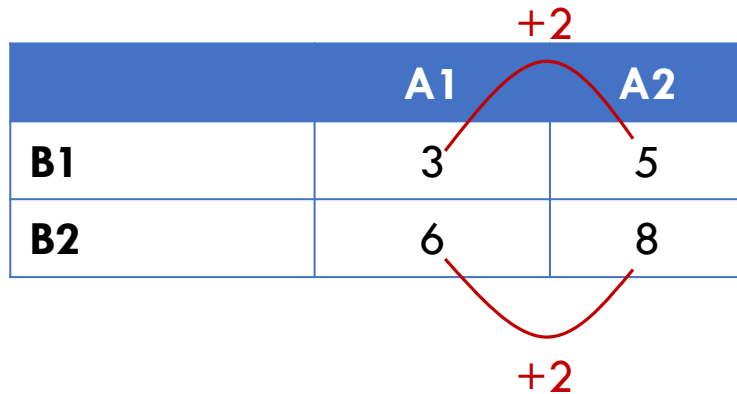


DESIGN DE EXPERIMENTO

INTERAÇÃO ENTRE FATORES

	A1	A2
B1	3	5
B2	6	8

INTERAÇÃO ENTRE FATORES



	A1	A2
B1	3	5
B2	6	8

Sem interação entre os fatores

O Fator A tem sempre o mesmo impacto no resultado

INTERAÇÃO ENTRE FATORES

	A1	A2
B1	3	5
B2	6	8

Sem interação entre os fatores

	A1	A2
B1	3	5
B2	6	9

+2

+3

INTERAÇÃO ENTRE FATORES

	A1	A2
B1	3	5
B2	6	8

Sem interação entre os fatores

	A1	A2
B1	3	5
B2	6	9

Com interação entre os fatores

O fator A tem impacto variável
no resultado, dependendo de B

DESIGN DE EXPERIMENTOS

Design Simples

Design fatorial completo

Design fatorial fracionário

DESIGN SIMPLES

Variação de apenas um fator por execução

Pode levar a conclusões erradas se houver interação entre os fatores

Número de experimentos baixo

DESIGN FATORIAL COMPLETO

Todas as combinações possíveis

Consegue encontrar os efeitos de cada fator

Design desejado

Muito custoso computacionalmente

DESIGN FATORIAL FRACIONÁRIO

Em casos onde as interações negligenciáveis são conhecidos

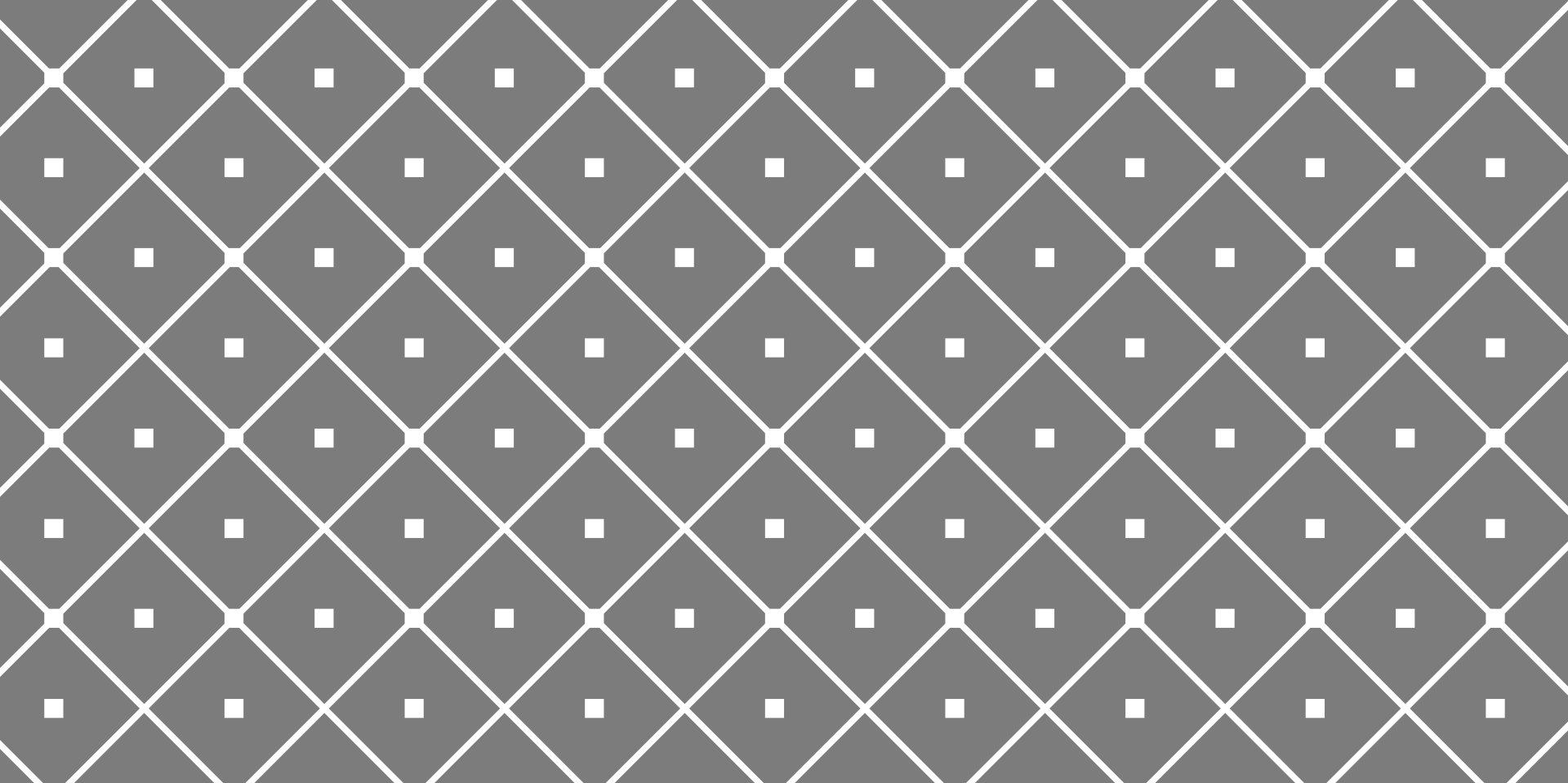
Economiza tempo e recursos

Não obtém todas as interações



Menos informação

Praticável



CHECKLIST PARA TESTES

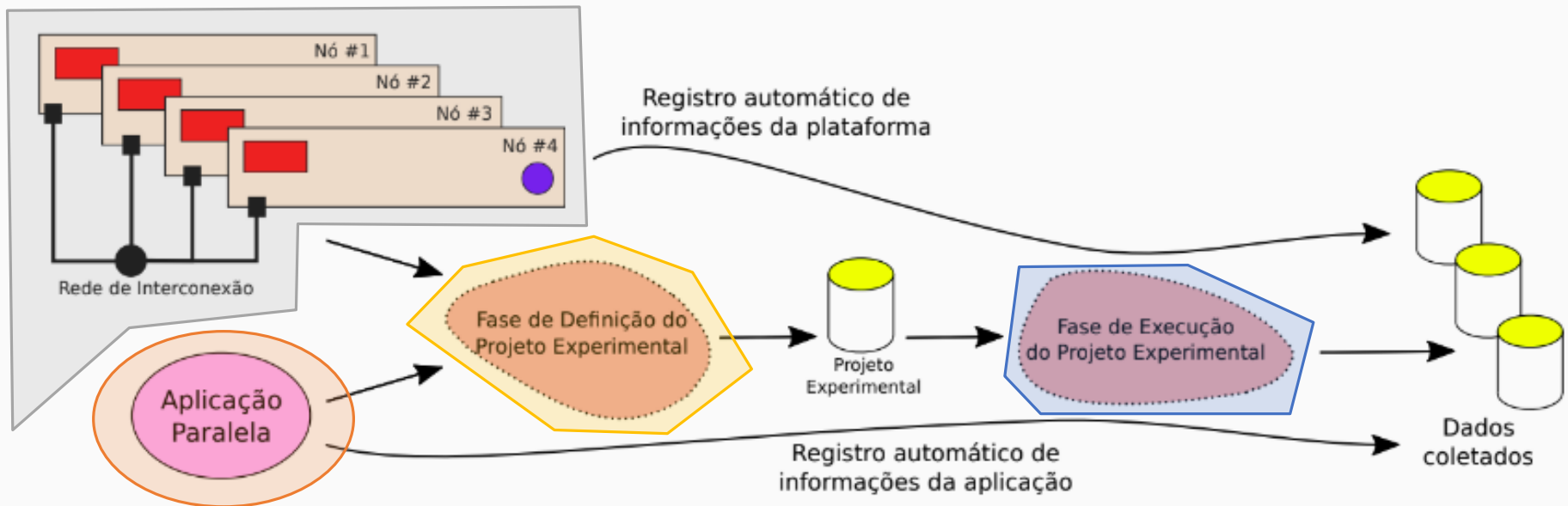
CHECKLIST PARA TESTES

1. Aplicação Paralela

3. Definição do Projeto

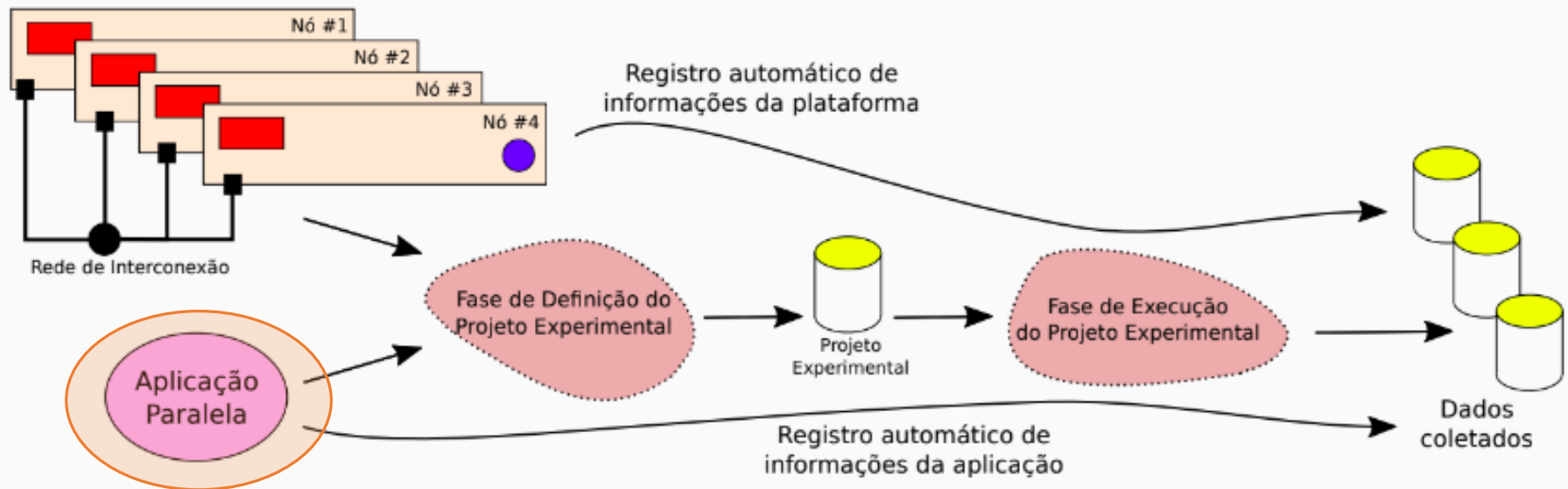
2. Máquinas e Rede

4. Execução dos Experimentos



CHECKLIST PARA TESTES

1. APLICAÇÕES PARALELAS



APLICAÇÃO PARALELA

Para assegurar a corretude da implementação paralela, deve-se verificar se os resultados paralelos batem com os sequenciais.

Crie tamanhos variados para a entrada

- Teste de escalabilidade

Garanta que a menor entrada rode em um tempo significativo

- Ex: maior que 10s

Não crie entradas aleatórias

- Usa a mesma entrada para variações nos fatores
- Melhor x pior caso, heurística e busca em espaço de projeto

EXEMPLO: INSERTION SORT



EXEMPLO: INSERTION SORT



EXEMPLO: INSERTION SORT



SWAP



OK

EXEMPLO: INSERTION SORT



EXEMPLO: INSERTION SORT



SWAP



OK

EXEMPLO: INSERTION SORT



EXEMPLO: INSERTION SORT



SWAP



OK

EXEMPLO: INSERTION SORT



EXEMPLO: INSERTION SORT



$O(N^2)$ Pior caso



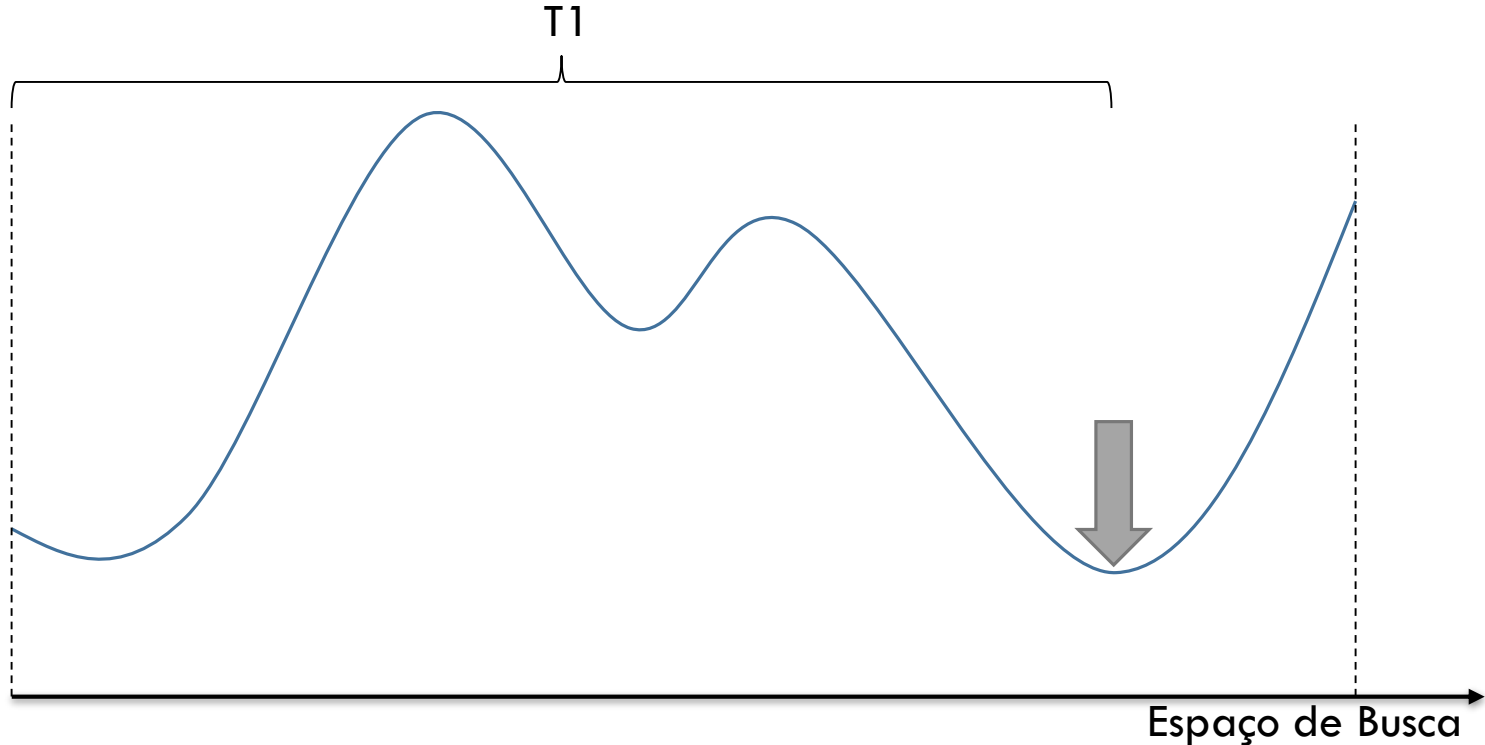
$O(N)$ Melhor caso

Dependendo da entrada,
teremos um desempenho
totalmente diferente!

Devemos utilizar o pior caso,
e sempre a mesma entrada
em todos os experimentos

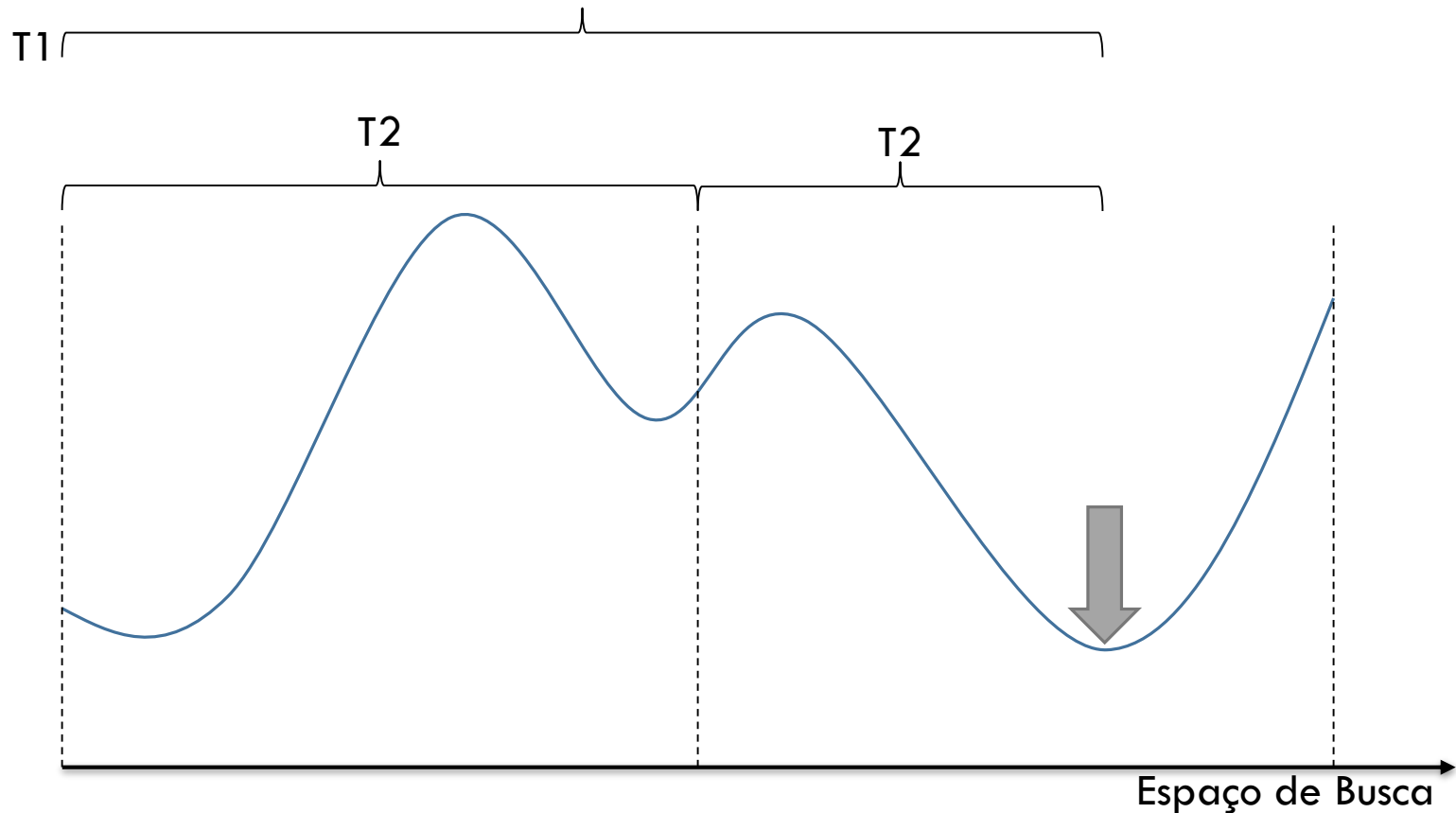
APLICAÇÃO PARALELA

EX: BUSCA PARA ENCONTRAR O MIN.



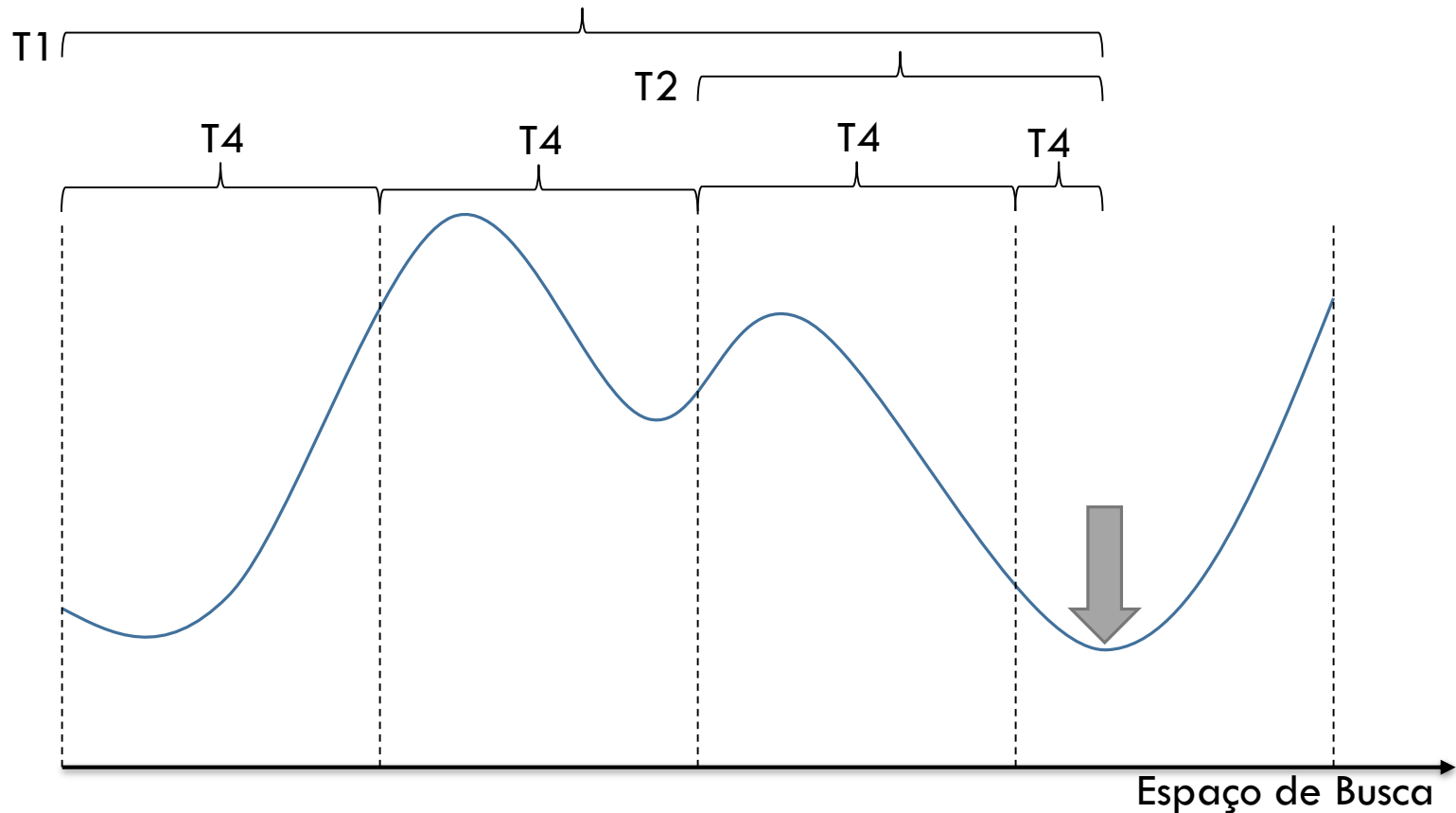
APLICAÇÃO PARALELA

EX: BUSCA PARA ENCONTRAR O MIN.



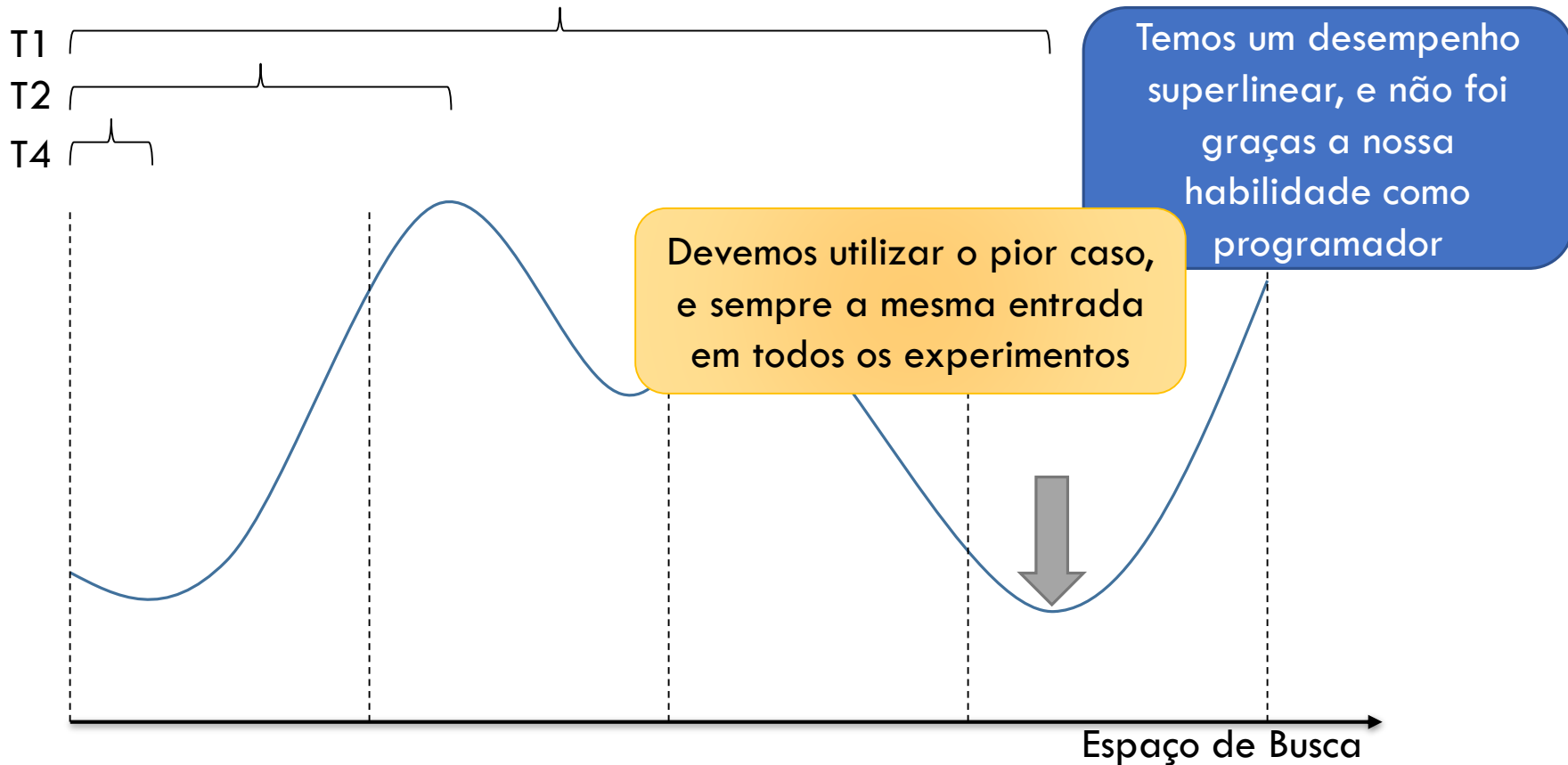
APLICAÇÃO PARALELA

EX: BUSCA PARA ENCONTRAR O MIN.



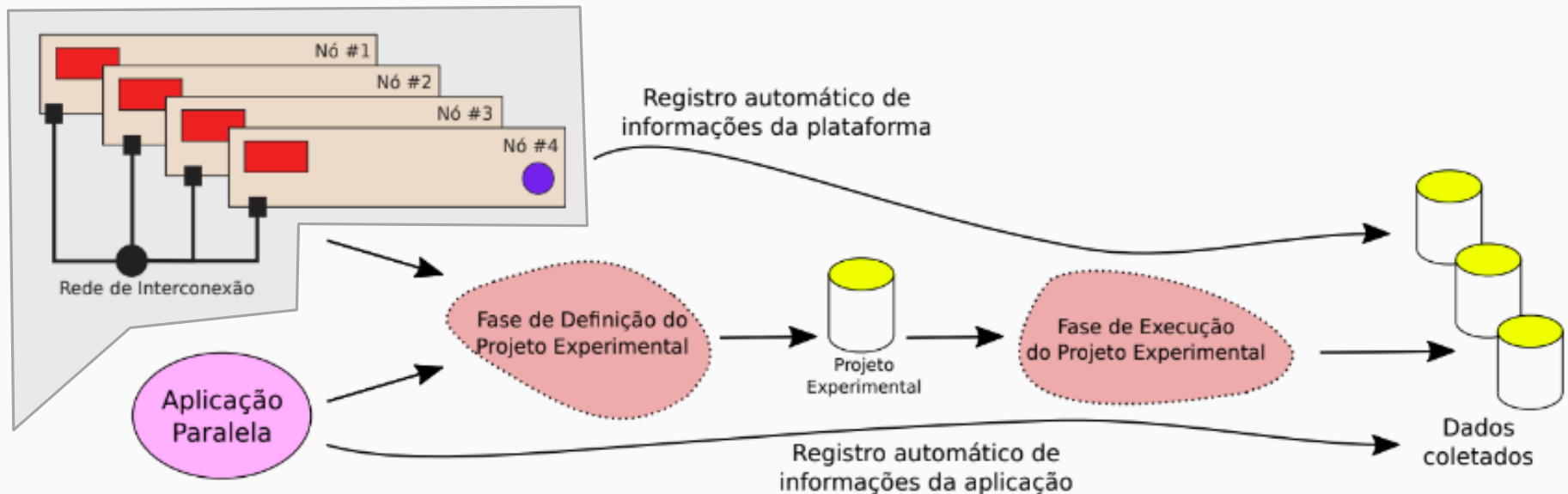
APLICAÇÃO PARALELA

EX: BUSCA PARA ENCONTRAR O MIN.



CHECKLIST PARA TESTES

2. MÁQUINAS E REDE



VARIABILIDADE

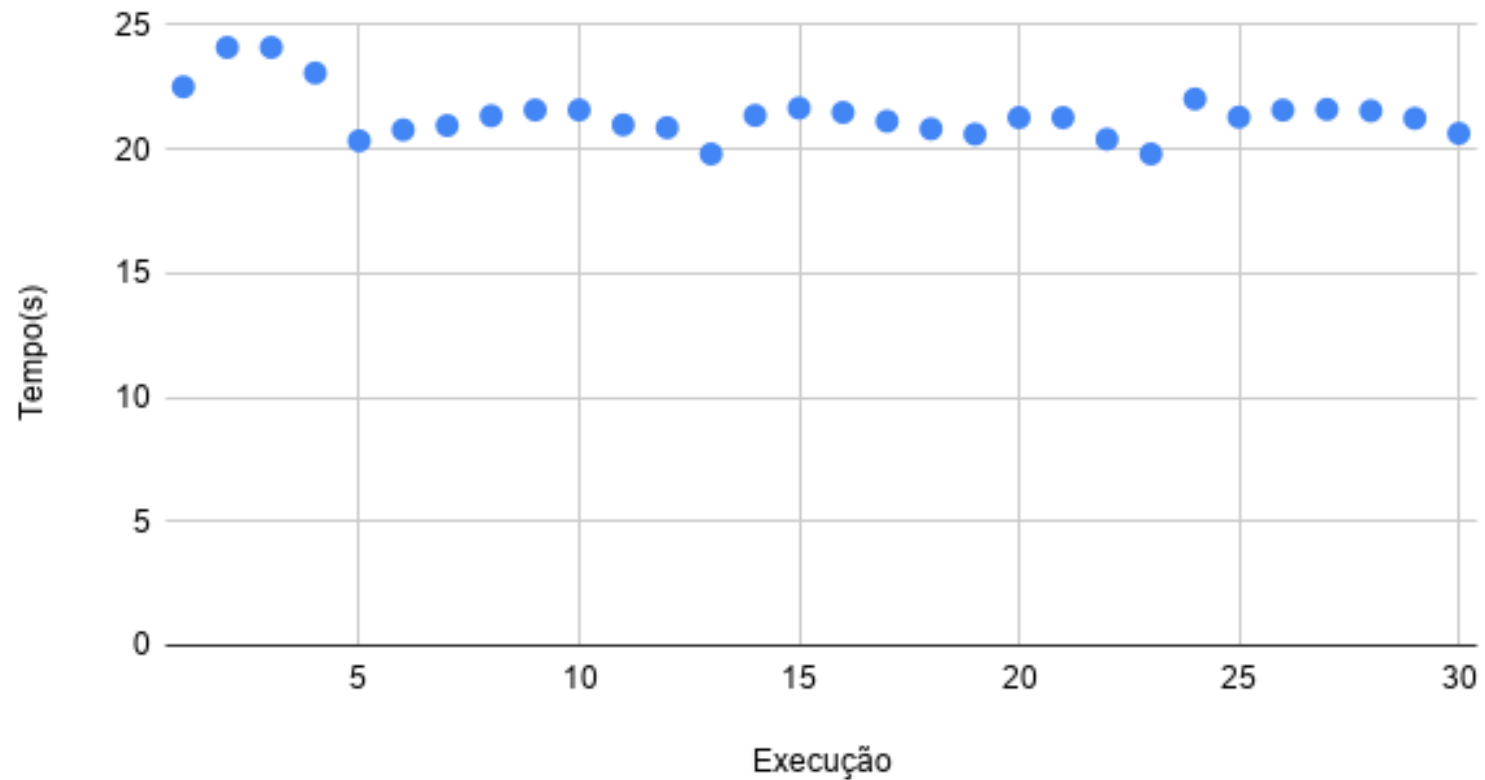
Indeterminismo da execução paralela (pela concorrência)

Aparição de anomalias durante a execução

Complexidade do sistema computacional

VARIABILIDADE

Tempo(s) vs. Execução



VARIABILIDADE

Tasks: 159, 705 thr; 2 running
Load average: 0.72 0.88 0.95
Uptime: 01:14:24

```
1 [|||||] 0.6% Tasks: 159, 705 thr; 2 running
2 [|||||] 5.3% Load average: 0.72 0.88 0.95
3 [|||||] 3.5% Uptime: 01:14:24
4 [|||||] 5.3%
Mem [|||||] 3.89G/7.78G
Swp [|||||] 0K/0K

PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
2537 marianaca  20    0 1454M 101M 60672 R   7.4  1.3  2:38.13 compiz
1801 root        20    0 569M  128M 111M  S   6.1  1.6  2:39.96 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/run/lightdm/root/:0 -nolisten
6019 marianaca  20    0 2937M 229M 111M  S   2.7  2.9  0:40.26 /snap/spotify/41/usr/share/spotify/spotify
2546 marianaca  9 -11  713M 17820 13700 S   2.7  0.2  1:05.05 /usr/bin/pulseaudio --start --log-target=syslog
6676 marianaca  20    0 27164 4960  3156 R   0.7  0.1  0:05.69 htop
5318 marianaca -6    0  713M 17820 13700 S   0.7  0.2  0:27.51 /usr/bin/pulseaudio --start --log-target=syslog
2891 marianaca  20    0 3359M 732M 170M  S   2.0  9.3  3:37.37 /usr/lib/chromium-browser/chromium-browser --enable-pinch
6217 marianaca  20    0 2937M 229M 111M  S   0.7  2.9  0:08.09 /snap/spotify/41/usr/share/spotify/spotify
2988 marianaca  20    0 1324M 90960 64236 S   0.7  1.1  0:59.61 /usr/lib/chromium-browser/chromium-browser --type=utility --utility-sub-type=net
2991 marianaca  20    0 1324M 90960 64236 S   0.7  1.1  0:52.50 /usr/lib/chromium-browser/chromium-browser --type=utility --utility-sub-type=net
6216 marianaca  20    0 2937M 229M 111M  S   1.3  2.9  0:11.07 /snap/spotify/41/usr/share/spotify/spotify
6149 marianaca  20    0 2937M 229M 111M  S   0.7  2.9  0:08.10 /snap/spotify/41/usr/share/spotify/spotify
3568 marianaca  20    0 658M  41516 30824 S   0.7  0.5  0:04.88 /usr/lib/gnome-terminal/gnome-terminal-server
2968 marianaca  20    0 3359M 732M 170M  S   0.7  9.3  0:45.47 /usr/lib/chromium-browser/chromium-browser --enable-pinch
6146 marianaca  20    0 2937M 229M 111M  S   0.0  2.9  0:07.49 /snap/spotify/41/usr/share/spotify/spotify
2706 marianaca  20    0 1245M 19916 14960 S   0.0  0.2  0:10.19 /usr/local/bin/warsaw/core
6667 marianaca  20    0 3359M 732M 170M  S   0.0  9.3  0:00.76 /usr/lib/chromium-browser/chromium-browser --enable-pinch
6302 marianaca  20    0 3359M 732M 170M  S   0.0  9.3  0:01.53 /usr/lib/chromium-browser/chromium-browser --enable-pinch
3414 marianaca  20    0 1245M 19916 14960 S   0.0  0.2  0:03.81 /usr/local/bin/warsaw/core
2463 marianaca  20    0 526M  29772 23320 S   0.0  0.4  0:00.34 /usr/lib/x86_64-linux-gnu/bamf/bamfdaemon
2409 marianaca  20    0 526M  29772 23320 S   0.0  0.4  0:02.05 /usr/lib/x86_64-linux-gnu/bamf/bamfdaemon
2607 marianaca  20    0 1454M 101M 60672 S   0.0  1.3  0:02.22 compiz
2667 marianaca  20    0 1506M 95524 50504 S   0.0  1.2  0:22.69 nautilus -n
933 root        20    0 4392  1284  1200 S   1.3  0.0  0:07.17 /usr/sbin/acpid
6240 marianaca  20    0 13.2G 169M 93368 S   1.3  2.1  0:22.96 /usr/lib/chromium-browser/chromium-browser --type=renderer --field-trial-handle=
1656 root        20    0 1147M 9692  8360 S   0.0  0.1  0:01.63 /opt/teamviewer/tv_bin/teamviewerd -d
3956 marianaca  20    0 3027M 348M 138M  S   0.0  4.4  0:00.23 /usr/lib/firefox/firefox
4148 marianaca  20    0 2552M 109M 88976 S   0.0  1.4  0:00.16 /usr/lib/firefox/firefox -contentproc -childID 3 -isForBrowser -prefsLen 6380 -p
4144 marianaca  20    0 2552M 109M 88976 S   0.0  1.4  0:01.59 /usr/lib/firefox/firefox -contentproc -childID 3 -isForBrowser -prefsLen 6380 -p
4821 marianaca  20    0 5585M 570M 183M  S   2.0  7.2  10:56.77 /usr/lib/chromium-browser/chromium-browser --type=renderer --field-trial-handle=
6159 marianaca  20    0 1553M 213M 95660 S   0.0  2.7  0:31.48 /snap/spotify/41/usr/share/spotify/spotify --type=renderer --no-sandbox --log-ft
5879 marianaca  20    0 5226M 127M 86080 S   0.7  1.6  0:05.07 /usr/lib/chromium-browser/chromium-browser --type=renderer --field-trial-handle=
4216 marianaca  20    0 5232M 136M 87716 S   0.0  1.7  0:04.68 /usr/lib/chromium-browser/chromium-browser --type=renderer --field-trial-handle=
6243 marianaca  20    0 13.2G 169M 93368 S   0.0  2.1  0:01.50 /usr/lib/chromium-browser/chromium-browser --type=renderer --field-trial-handle=
3409 root        20    0 1112M 30456 19752 S   0.0  0.4  0:01.31 /usr/local/bin/warsaw/core
1 root        20    0 181M  6084  3980 S   0.0  0.1  0:01.45 /sbin/init splash
926 root        20    0 162M  8652  7936 S   0.0  0.1  0:00.27 /usr/sbin/thermald --no-daemon --dbus-enable
6156 marianaca  20    0 2937M 229M 111M  S   0.0  2.9  0:00.85 /snap/spotify/41/usr/share/spotify/spotify
1341 mysql      20    0 660M  74712 16084 S   0.0  0.9  0:03.13 /usr/sbin/mysqld
1585 mysql      20    0 660M  74712 16084 S   0.0  0.9  0:00.21 /usr/sbin/mysqld
6688 marianaca  20    0 5585M 570M 183M  S   0.7  7.2  0:04.65 /usr/lib/chromium-browser/chromium-browser --type=renderer --field-trial-handle=
5745 marianaca  20    0 5585M 570M 183M  S   0.0  7.2  0:19.62 /usr/lib/chromium-browser/chromium-browser --type=renderer --field-trial-handle=
1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit
```

MÁQUINAS

Único computador: software e hardware

Cluster: todos os nós e a rede de interconexão

MÁQUINAS — ÚNICO COMPUTADOR

Não executar em servidores virtualizados

Não execute em servidores com outros usuários logados

Desligue quaisquer fontes de interferência

Use o modo usuário

MÁQUINAS - CLUSTER

Não executar em servidores virtualizados

Não execute em servidores com outros usuários logados

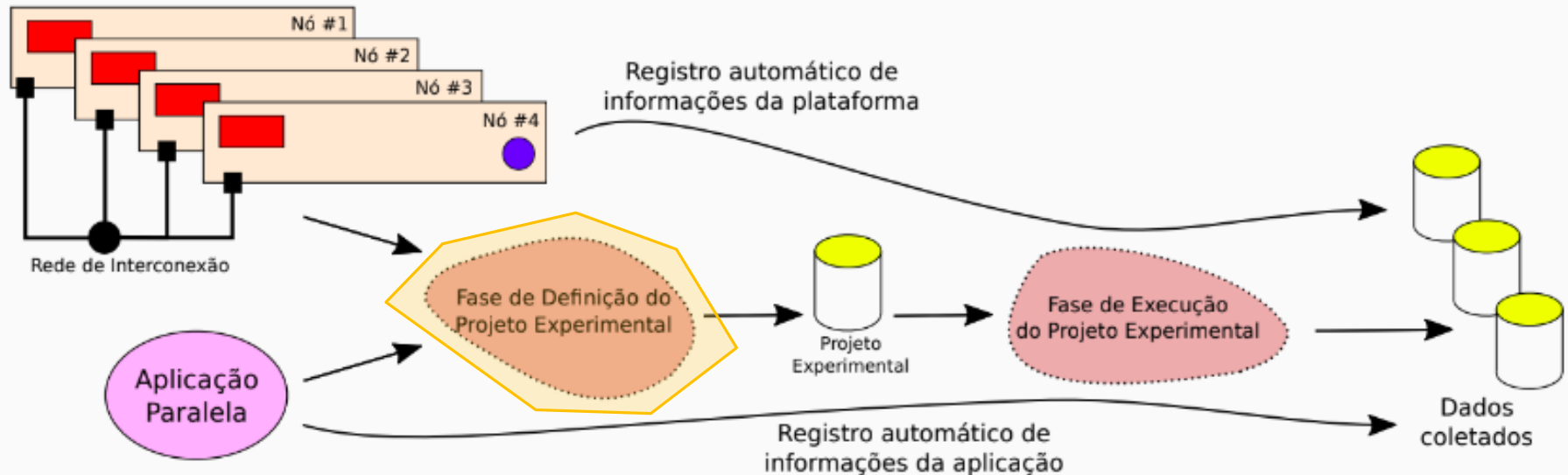
Desligue quaisquer fontes de interferência

Use o modo usuário

Escolha a máquina a ser usada e idealmente utilize máquinas iguais

CHECKLIST PARA TESTES

3. DEFINIÇÃO DO PROJETO



PROJETO EXPERIMENTAL

Escolha as medições que deseja fazer

- Ex: escalabilidade forte ou fraca

PROJETO EXPERIMENTAL

Escolha as medições que deseja fazer

- Ex: escalabilidade forte ou fraca

Defina as variáveis de resposta que precisa

PROJETO EXPERIMENTAL

Escolha as medições que deseja fazer

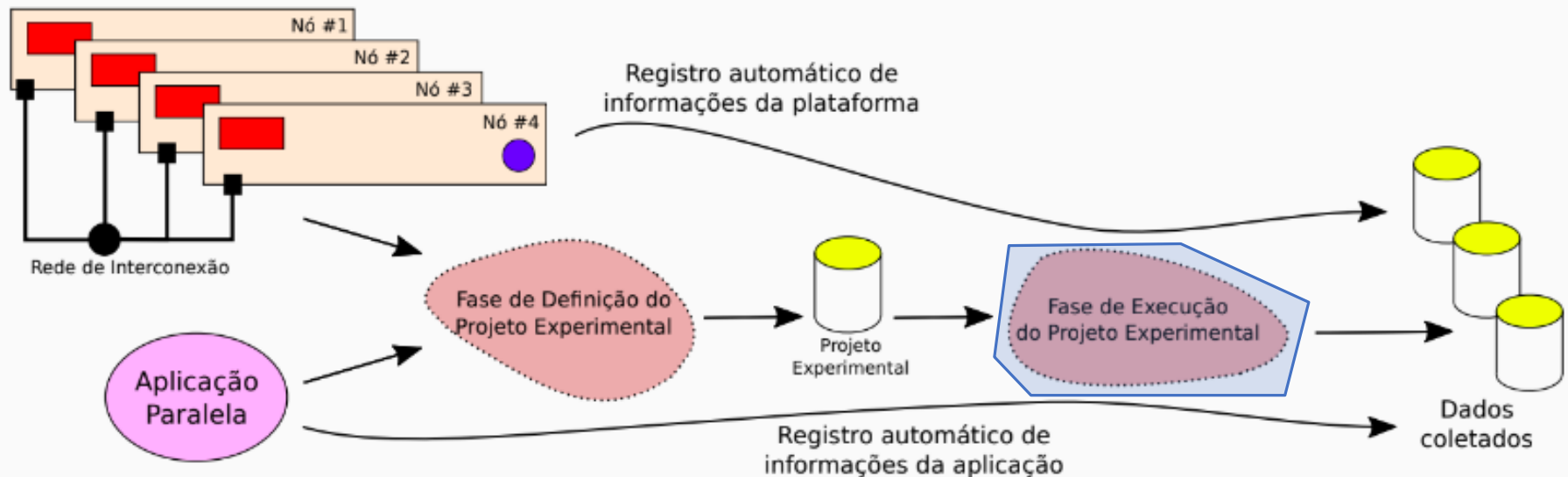
- Ex: escalabilidade forte ou fraca

Defina as variáveis de resposta que precisa

Defina os fatores que deseja alterar e quais os valores possíveis

CHECKLIST PARA TESTES

4. EXECUÇÃO DOS EXPERIMENTOS



EXECUÇÃO

Defina a quantidade de execuções significativo para a sua medição ,
quanto mais melhor

- Lembre-se que sempre que alterar um fator deve executar xN
- Criação de scripts

Lembre-se de calcular o speedup encima da média das execuções.

CONTROLE EXPERIMENTAL

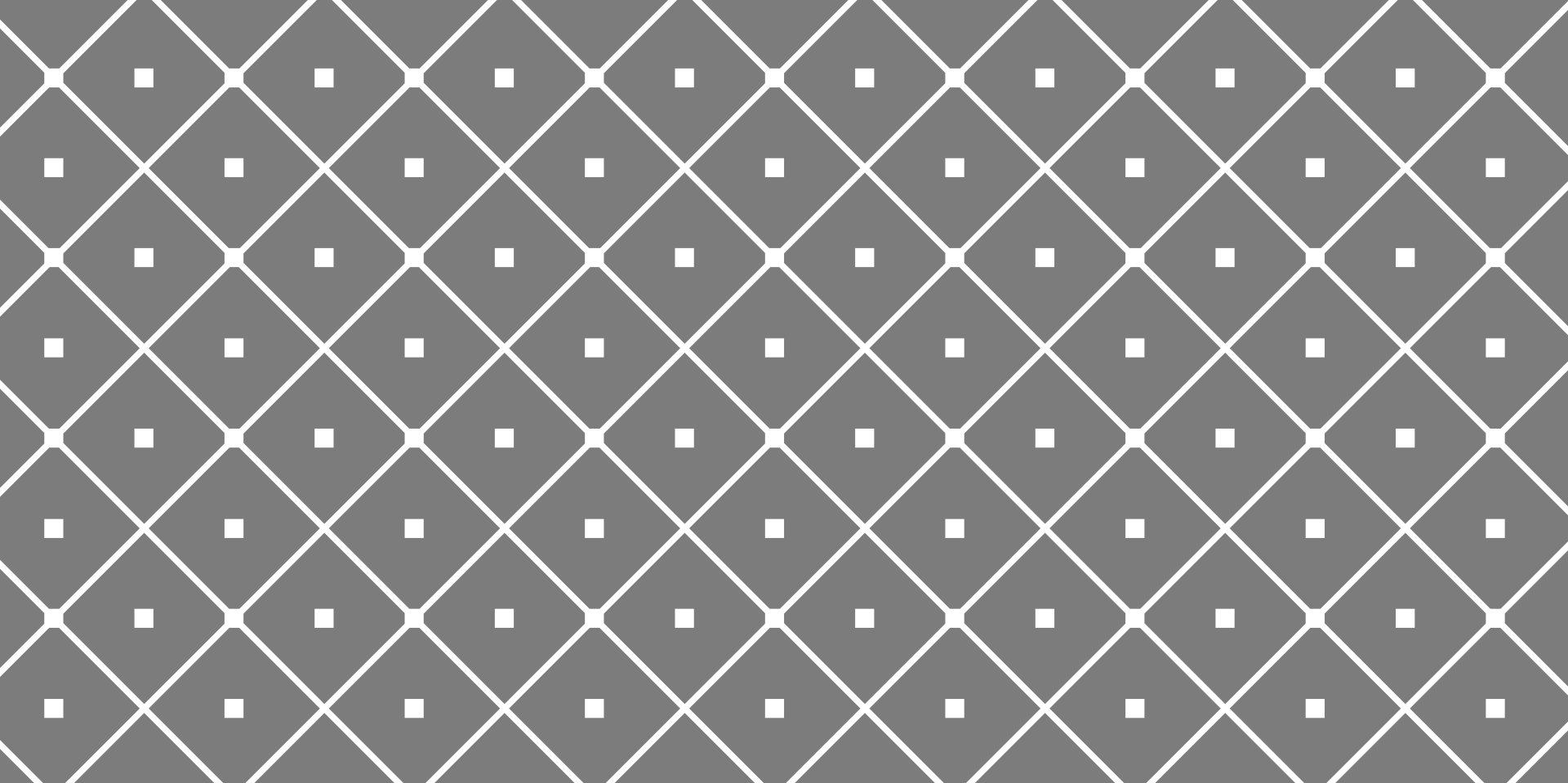
Desvantagens

- Experimentos se tornam mais burocráticos
- Cuidado maior no antes, durante e depois dos experimentos
- Processo investigativo pode ser tornar mais lento

CONTROLE EXPERIMENTAL

Vantagens

- Conclusões delineadas sejam mais perenes, significativas
- Relato facilitado (pois há substrato para derivar conclusões)
- **Facilita a reprodutibilidade**



CHECK LIST FINAL

CHECK LIST

Aplicação paralela

- Está correta? Algumas vezes retorna segfault?

Entrada

- Tenho um N que roda em pelo menos 10s?
- Tenho tamanhos variados? (2N, 4N, 6N)
- Usei a mesma semente? (caso da entrada gerada aleatoriamente)

Máquina

- É um servidor virtualizado? Nem pense em rodar na amazon cloud.
- Estou usando modo usuário? Grub.

CHECK LIST

Máquina

- Consigo usar a mesma máquina do começo ao fim dos experimentos?
- É um servidor virtualizado? Nem pense em rodar na amazon cloud.
- Estou usando modo usuário? Grub.
- Isolei minha máquina para que nenhum outro usuário utilize?
- Desliguei fontes de interferência?
 - Interface gráfica?
 - Rede?

CHECK LIST

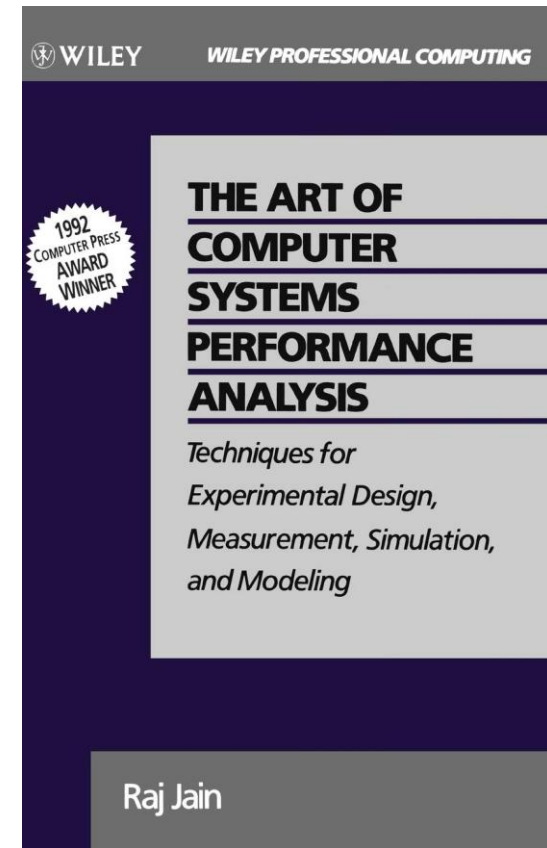
Projeto experimental

- Defini quantas vezes preciso rodar?
 - Desvio padrão pode ajudar a definir o número ideal.
- Estou rodando outros processos enquanto faço meus experimentos?
 - Nada de música ou vídeos.
- Qual versão do gcc estou usando?
- Qual flag de compilação?
- Qual SO?

LITERATURA

Leitura fortemente recomendada

**The art of computer systems performance analyses:
Techniques for experimental design, measurement,
simulation, and modeling** by Raj Jain. Wiley, 1991.



LITERATURA

Leitura recomendada

Measuring Computer Performance by David
J. Lilja. 2000.

