Cálculo de primos em paralelo

Alunos: Igor Padilha, Osmar Souza

Descobrindo números primos

- Um método comum usado para determinar se um número é primo se baseia na divisão exaustiva desse número
- Para um número N ser primo, testamos N / x para todo N > x > 2

```
=== PRIMES ===
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197
199 211 223 227 229 233 239 241 251 257 263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 359 367 373 379 383 389 397 401 409 419 421 431
 433 439 443 449 457 461 463 467 479 487 491 499 503 509 521 523 541 547 557 563 569 571 577 587 593 599 601 607 613 617 619 631 641 643 647 653 659 66
1 673 677 683 691 701 709 719 727 733 739 743 751 757 761 769 773 787 797 809 811 821 823 827 829 839 853 857 859 863 877 881 883 887 907 911 919 929 9
37 941 947 953 967 971 977 983 991 997 1009 1013 1019 1021 1031 1033 1039 1049 1051 1061 1063 1069 1087 1091 1093 1097 1103 1109 1117 1123 1129 1151 11
53 1163 1171 1181 1187 1193 1201 1213 1217 1223 1229 1231 1237 1249 1259 1277 1279 1283 1289 1291 1297 1301 1303 1307 1319 1321 1327 1361 1367 1373 138
1 1399 1409 1423 1427 1429 1433 1439 1447 1451 1453 1459 1471 1481 1483 1487 1489 1493 1499 1511 1523 1531 1543 1549 1553 1559 1567 1571 1579 1583 1597
 1601 1607 1609 1613 1619 1621 1627 1637 1657 1663 1667 1669 1693 1697 1699 1709 1721 1723 1733 1741 1747 1753 1759 1777 1783 1787 1789 1801 1811 1823
1831 1847 1861 1867 1871 1873 1877 1879 1889 1901 1907 1913 1931 1933 1949 1951 1973 1979 1987 1993 1997 1999 2003 2011 2017 2027 2029 2039 2053 2063 2
069 2081 2083 2087 2089 2099 2111 2113 2129 2131 2137 2141 2143 2153 2161 2179 2203 2207 2213 2221 2237 2239 2243 2251 2267 2269 2273 2281 2287 2293 22
97 2309 2311 2333 2339 2341 2347 2351 2357 2371 2377 2381 2383 2389 2393 2399 2411 2417 2423 2437 2441 2447 2459 2467 2473 2477 2503 2521 2531 2539 254
3 2549 2551 2557 2579 2591 2593 2609 2617 2621 2633 2647 2657 2659 2663 2671 2677 2683 2687 2689 2693 2699 2707 2711 2713 2719 2729 2731 2741 2749 2753
```

Otimizações: Pares de divisores

- Divisores vêm em pares
  - 6/2 = 3
  - Logo 6 é divisivel por 3
  - 6/3 = 2
- O maior divisor mediano de um número é a raíz dele mesmo
  - O divisor mediano de 36 é 6 (sqrt(36) = 6)
  - 2, 3, **6**, 18, 12
- Portanto, para descobrir de um N é primo, precisamos testar apenas N / x para todo sqrt(N) > x > 2

Implementação

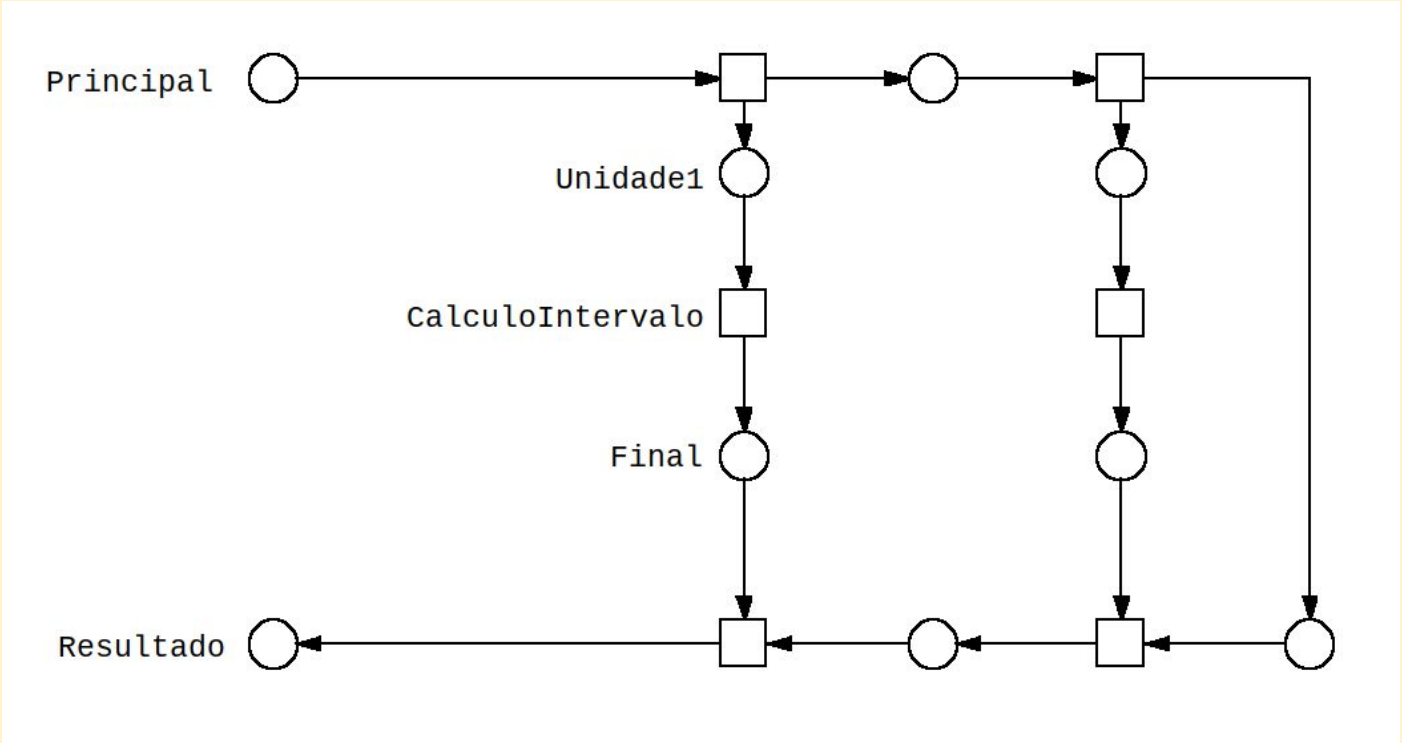
Testa números até sqrt(N)

```
IntBuffer search_primes(int start, int end) {
    IntBuffer buffer = {0};

    if (start < 2) {
        start = 2;
    }

    for (int number = start; number <= end; number++) {
        bool prime = true;

        for (int test = 2; test * test <= number; test++) {
            if (number % test == 0) {
                prime = false;
                break;
            }
        }

        if (prime) {
            bf_append(buffer, number);
        }
    }
    return buffer;
}
```

Utilizando processamento paralelo

- Para listar todos os primos entre 0 e N, dividimos o intervalo entre 1000 threads para o processamento
  - A primeira thread processa o intervalo de 0 a 1000
  - A segunda thread processa o intervalo de 1000 a 2000
  - e assim por diante
- Ao final a thread principal coleta todos os primos processados e exibe na saída do programa

Rede de petri



Gerenciamento de recurso

- As threads não compartilham recursos durante a sua execução
- A thread principal espera cada thread finalizar o seu processamento para coletar os primos processados