Parallel Prime Number Generation Using The Sieve of Eratosthenes

Adrian Tudor Pănescu
adrian.panescu@epfl.ch

Problem

- The Sieve of Eratosthenes is a classic algorithm for finding all the prime numbers up to a specified integer (*limit*);
- Marks out composite numbers - the multiples of each prime starting with 2 (or 3).

Sequential algorithm

$Eratosthenes$(limit)

$list \Leftarrow [2, 3, ..., limit]$

for $i = 2, 3, ..., \frac{limit}{2}$ **do**

  if $list[i]$ **then**

    for $j = 2i, 3i..., ki \mid j \leq n$ **do**

      $list[j] = nil$ {we will call this operation a "cross-out" (of a composite number)}

    end for

  end if

end for

Parallelization Strategy

- Each process will be responsible (i.e. remove composites) for a part of the array representing a contiguous segment of natural numbers;
- The first process (*master*) will loop through its assigned segment, find the primes and broadcast them to the other processes that will cross-out the composites in their lists.

Computation versus Communication

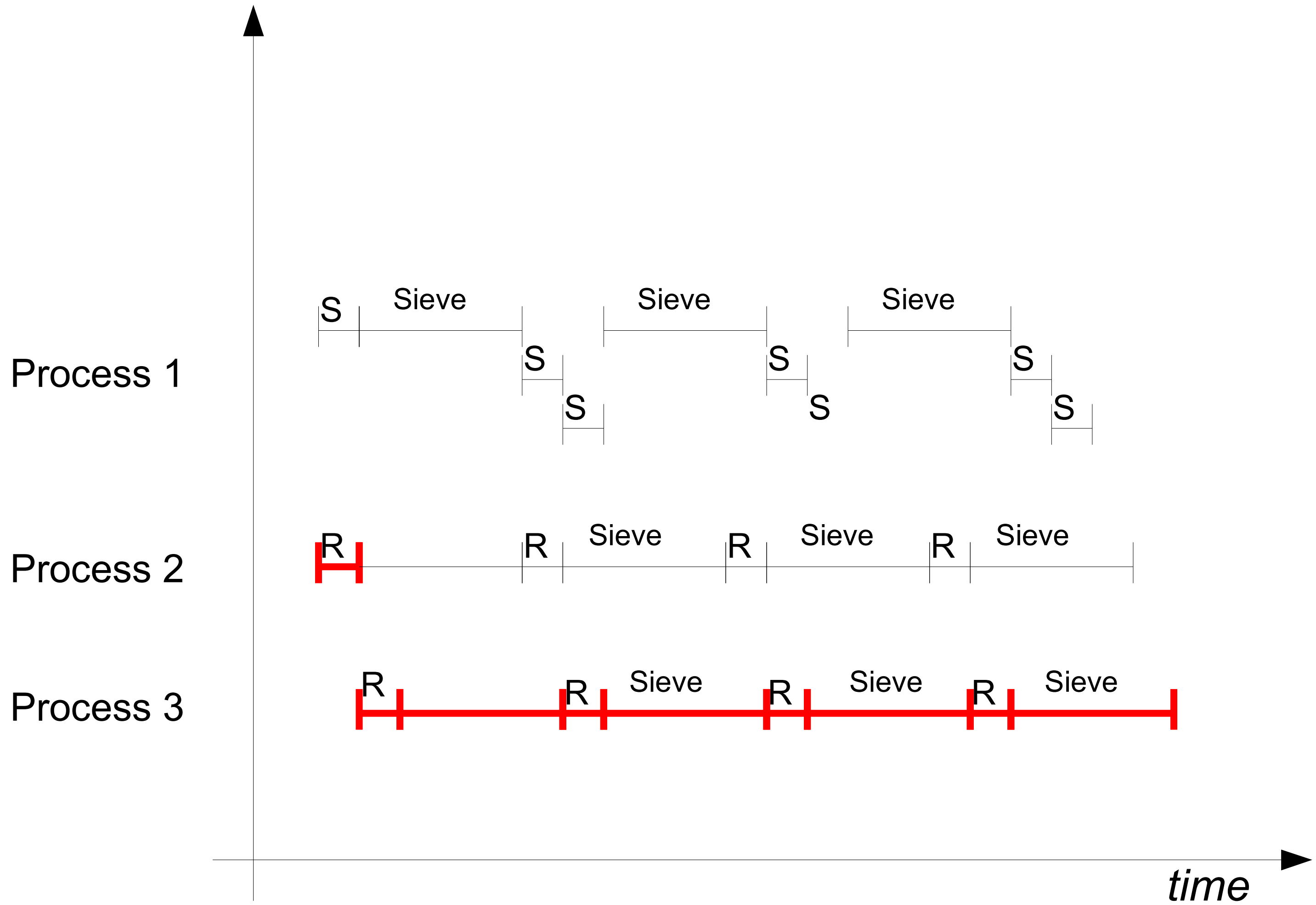
- If we have p processes and k primes, each process will have assigned ~(limit/p) numbers => k loops of length ~(limit/p) per process;
- The master process will send (k*p) messages;
- k = O(*limit* log *limit*).

Amdahl's Law

$$
\begin{aligned}
speedup &= \frac{t_{serial}}{t_{parallel}} \\
&\leq \frac{1}{f+\frac{(1-f)}{p}} \\
&= \frac{1}{\frac{1}{l+1}+\frac{1-\frac{1}{l+1}}{p}} \\
&\simeq \frac{1}{\frac{1}{\sqrt{limit}}+\frac{\sqrt{limit}}{p\cdot(\sqrt{limit}+1)}}
\end{aligned}
$$

- Serial part of the algorithm: one loop to initialize the array;
- Total: (l+*1*) loops, where $l^2 \approx$ *limit*, as the algorithm will stop sieving upon reaching a prime larger than the square root of *limit*;

Timing Diagram



Theoretical speedup analysis

- Let *t* be the time spent crossing-out a single composite;
- Total time spent by **one** process sieving is bounded by:

$$t \cdot (\lceil \frac{\lceil \frac{limit}{p} \rceil}{2} \rceil + \lceil \frac{\lceil \frac{limit}{p} \rceil}{3} \rceil + ... + \lceil \frac{\lceil \frac{limit}{p} \rceil}{prime_k} \rceil)$$

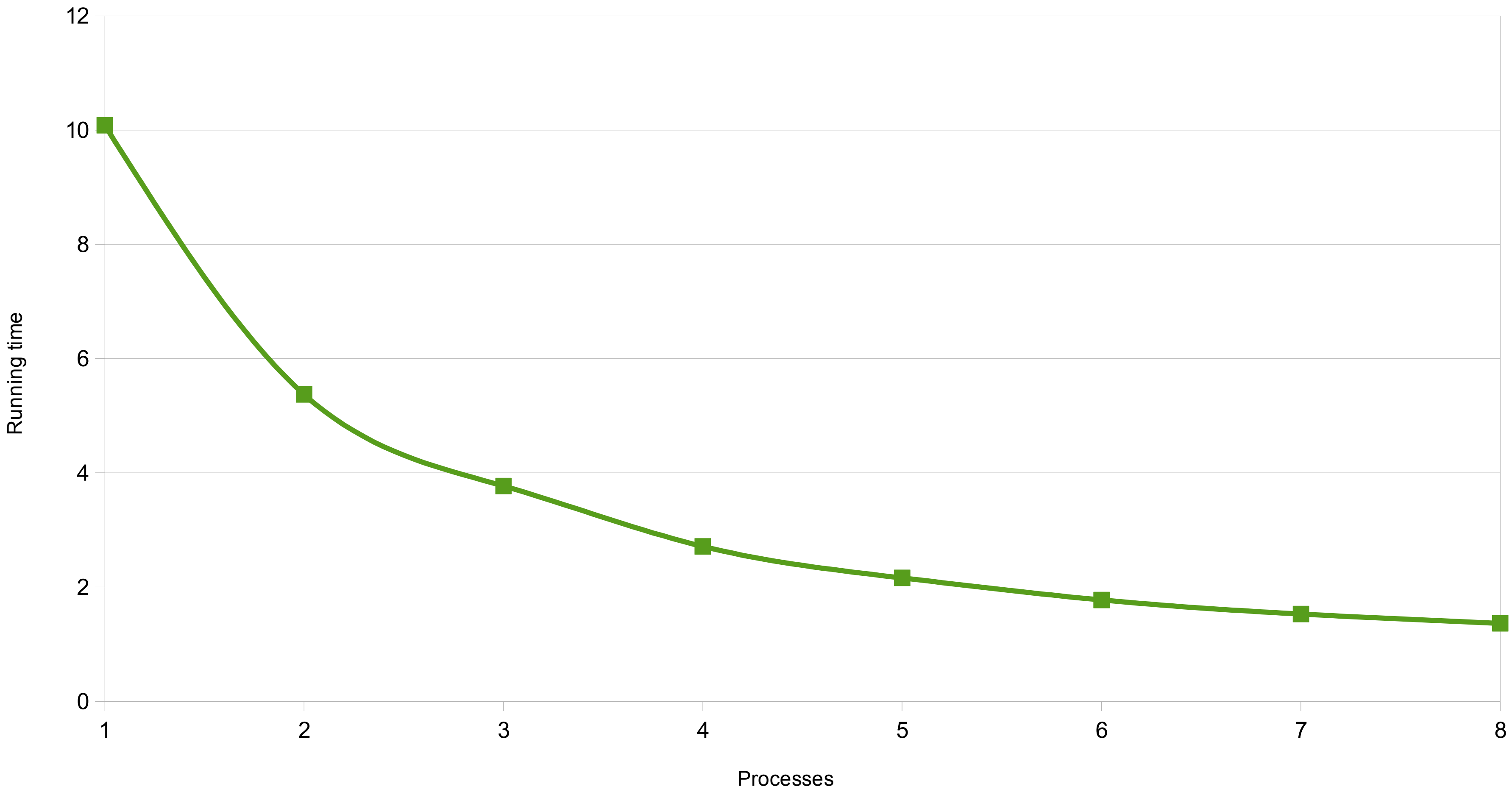
- If *d* is the time spent sending a prime from the master to another process, the **total** communication time will be: *d * k * (p-1)*

Execution times I

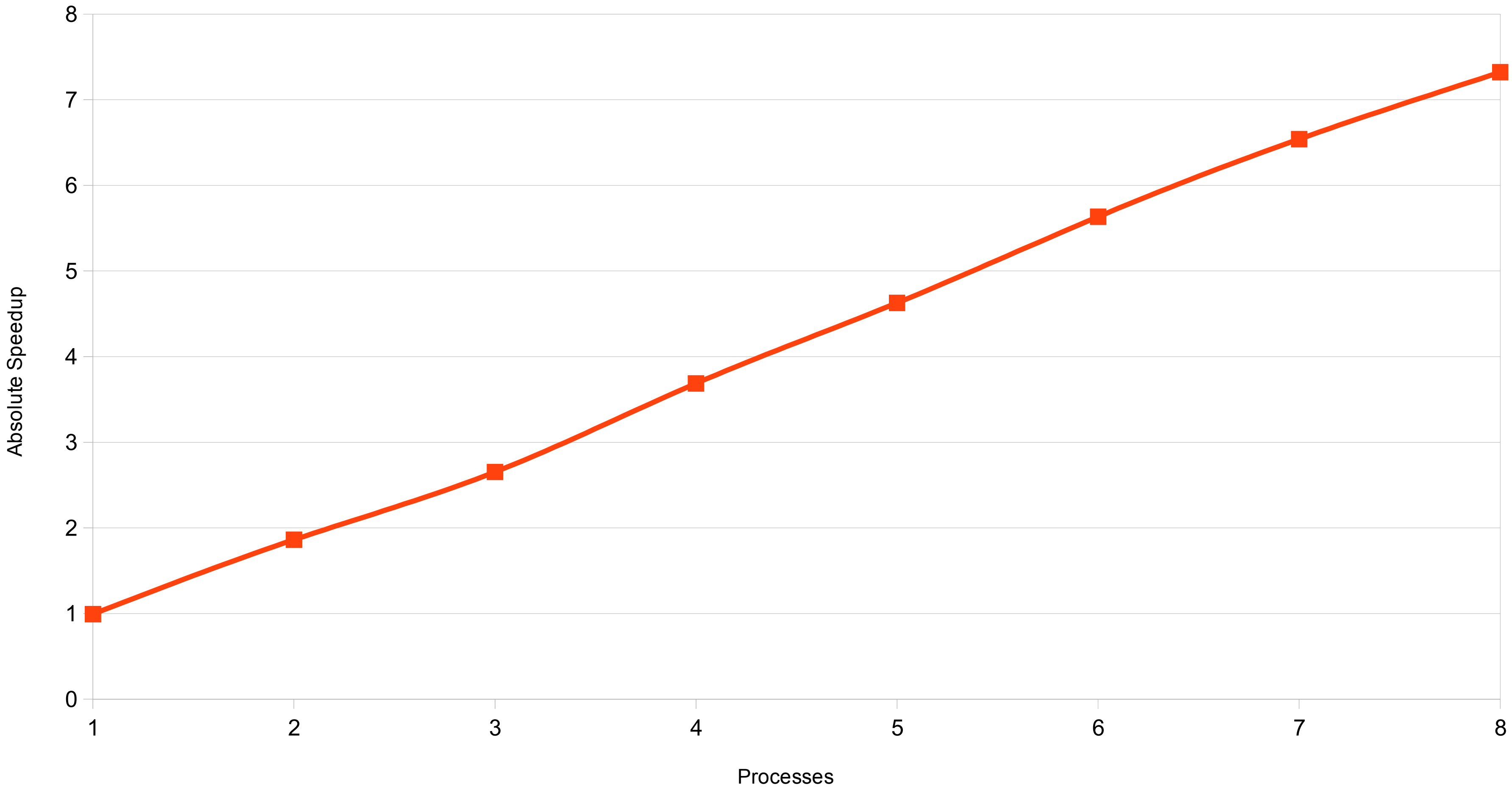
- *limit* = 2^{28};

Number of processes	Execution time (seconds)
1 (sequential)	~10
1 (parallel)	10.085
2	5.372
3	3.770
4	2.712
5	2.161
6	1.775
7	1.529
8	1.366

Execution times II



Absolute Speedup

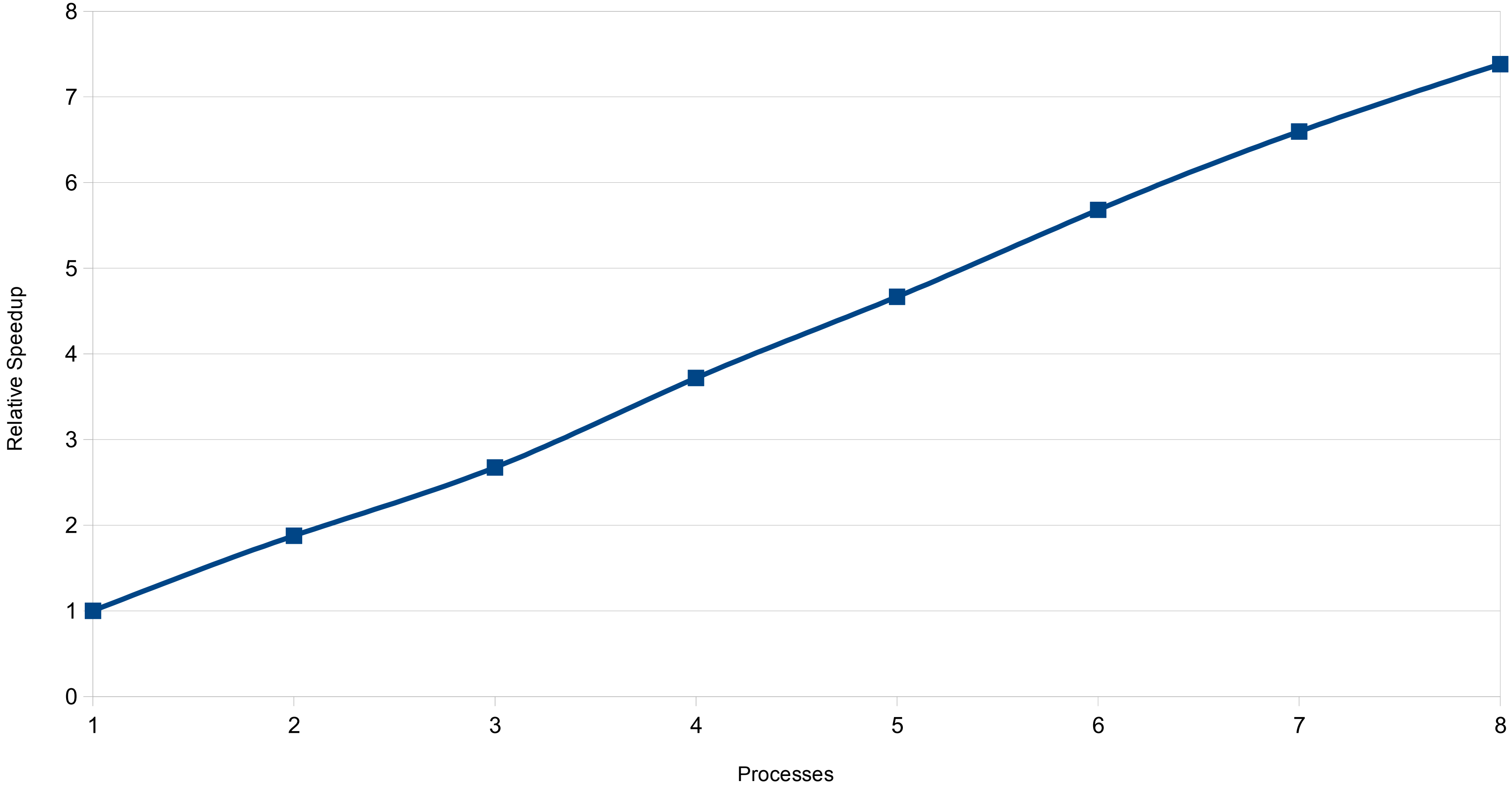


Amdahl's Law revisited

$$
\begin{aligned}
speedup &= \frac{t_{serial}}{t_{parallel}} \\
&\leq \frac{1}{f+\frac{(1-f)}{p}} \\
&= \frac{1}{\frac{1}{l+1}+\frac{1-\frac{1}{l+1}}{p}} \\
&\simeq \frac{1}{\frac{1}{\sqrt{limit}}+\frac{\sqrt{limit}}{p\cdot(\sqrt{limit}+1)}}
\end{aligned}
$$

For $l = 2^{28}$ and $p = 8$ we obtain that *speedup* ≤ 7.996

Relative Speedup



Project Planning

- Possible optimizations:
  - Eliminate communication by running an initial sieving loop on every process;
  - OpenMP;
- Re-run experiments for a larger limit and on a bigger cluster ($p > 8$);

Conclusions

- The Sieve of Eratosthenes is an algorithm that can benefit from parallelization;
- Unlike other applications, communication time is reduced and may even be eliminated.