CS200 LECTURE 9

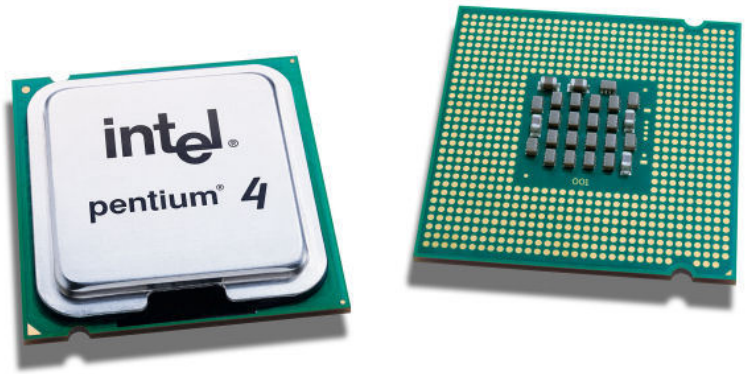
DISTRIBUTED ALGORITHMS

Benjamin Li

April 1, 2022

Why Distributed Algorithms?

Historical Context



So far, sequential algorithms

CPUs used to be single core, not anymore!

Distributed (parallel) algorithms make better use of hardware

Speedup

How much of a speedup can we actually get?

Depends on code itself

Amdahl's Law -

$$\frac{1}{1 \quad p + \frac{p}{s}}$$

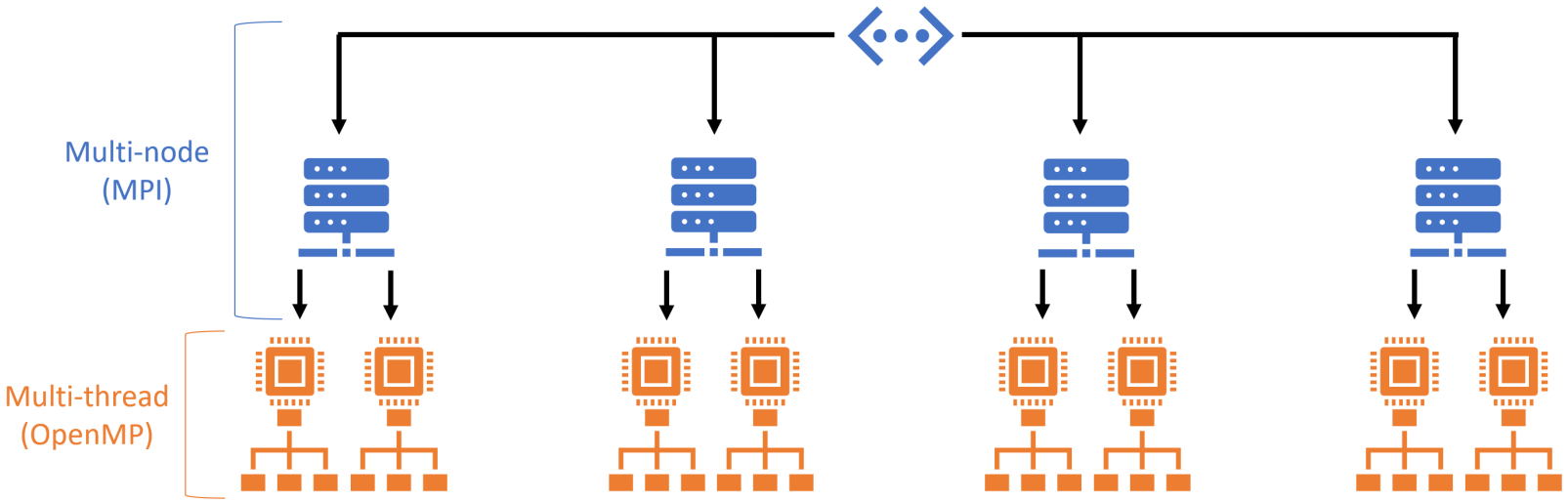
p - fraction of original execution time that can be optimized

s - speedup of optimized code over original code

This throws practical applications in the face of theory

distinction of $O(2n)$ and $O(n)$

Levels of Parallelism



Message Passing Interface (MPI)

Message Passing Interface (MPI)

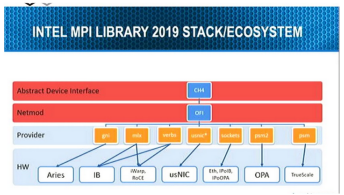
MPI is a standard designed to allow for parallel functionality in a cluster

Every implementation has to follow this standard

Looks like function calls in code (library)

Pass data between processes as messages

- 1:1 transmit and receive



Hello World Example

MPI has a whole slew of new concepts

process - instance of the program being run

rank - unique identi er for a process (usually related to thread count)

communicators - groups together MPI processes
(Ex. `MPI_COMM_WORLD`)

nalization - clean up performed at end of MPI program

Does MPI Work?

Let's take a look at the example of bitonic sort.

Bitonic sorter

From Wikipedia, the free encyclopedia

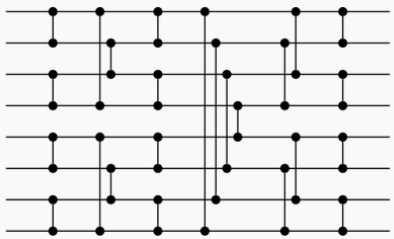
This article **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.
Find sources: "Bitonic sorter" – news · newspapers · books · scholar · JSTOR *(October 2017)* *(Learn how and when to remove this template message)*

Bitonic mergesort is a parallel algorithm for sorting. It is also used as a construction method for building a sorting network. The algorithm was devised by Ken Batcher. The resulting sorting networks consist of $O(n \log^2(n))$ comparators and have a delay of $O(\log^2(n))$, where n is the number of items to be sorted.[1]

A sorted sequence is a monotonically non-decreasing (or non-increasing) sequence. A *bitonic* sequence is a sequence with $x_0 \leq \cdots \leq x_k \geq \cdots \geq x_{n-1}$ for some $k, 0 \leq k < n$, or a circular shift of such a sequence.

Contents [hide]
1 Complexity
2 How the algorithm works
2.1 Alternative representation
3 Example code
4 See also
5 References
6 External links

Bitonic sorter



Bitonic sort network with eight inputs.

Class	Sorting algorithm
Data structure	Array
Worst-case performance	$O(\log^2(n))$ parallel time
Best-case performance	$O(\log^2(n))$ parallel time
Average performance	$O(\log^2(n))$ parallel time
Worst-case space complexity	$O(n \log^2(n))$ non-parallel time

Complexity [edit]

Let $p = \lfloor \log_2 n \rfloor$ and $q = \lceil \log_2 n \rceil$.

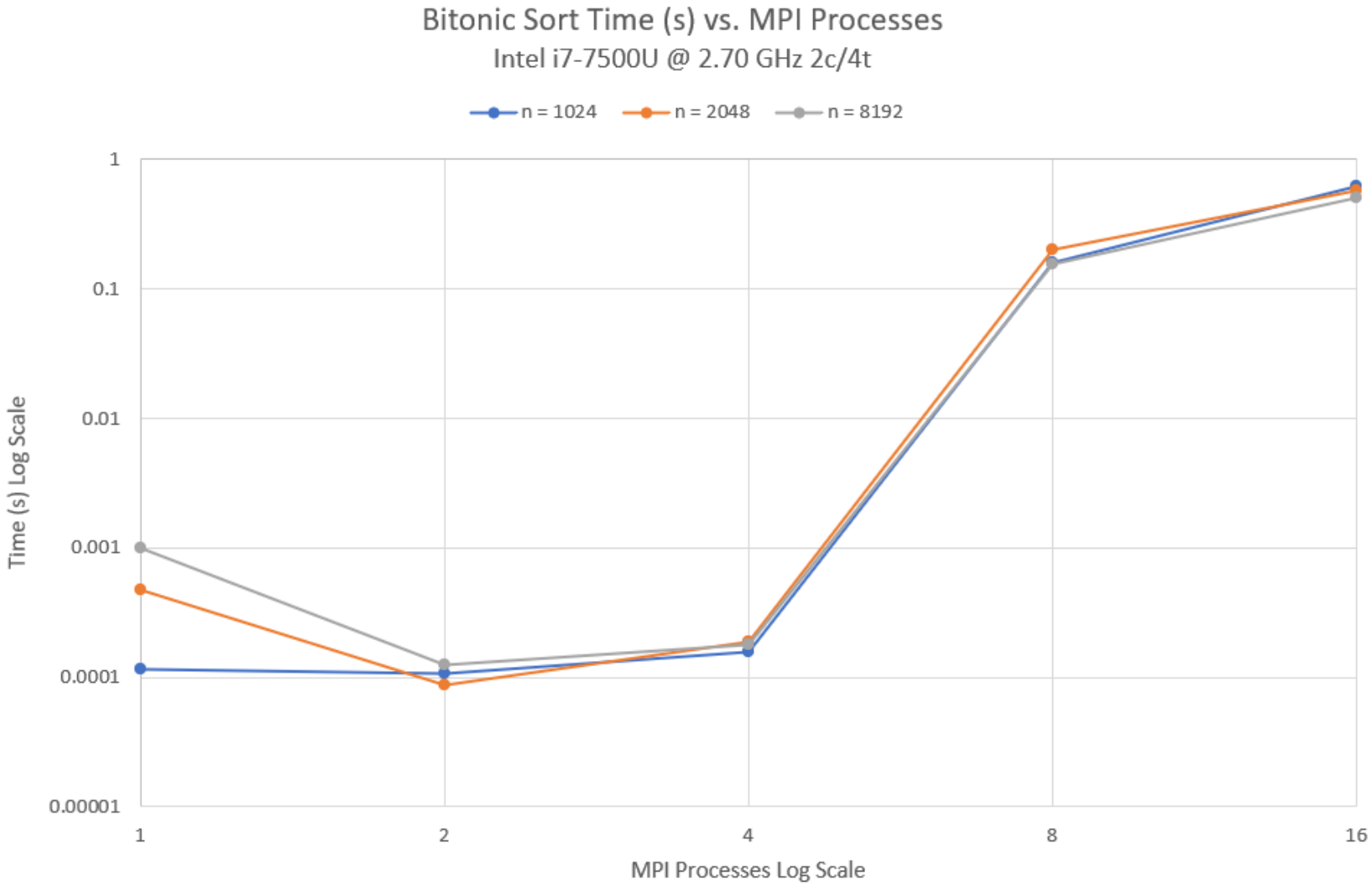
It is obvious from the construction algorithm that the number of rounds of parallel comparisons is given by $q(q+1)/2$.

It follows that the number of comparators c is bounded $2^{p-1} \cdot p(p+1)/2 \leq c \leq \lfloor n/2 \rfloor \cdot q(q+1)/2$ (which establishes an exact value for c when n is a power of 2).

Merge Sort Runtime: $O(n \log n)$

Parallel Bitonic Sort Runtime: $O(\log^2 n)$

Data for Bitonic Sort



Calculating Pi Example

Mathematicians are very concerned (perhaps to a worrying degree) about nding all the digits of .

To date, 62.8 trillion digits of have been calculated.

Leibniz formula:

$$1 \quad \frac{1}{3} + \frac{1}{5} \quad \frac{1}{7} + \frac{1}{9} \quad \cdots = \frac{}{4}$$

We can calculate this in a distributed fashion!