FRANCES ALLEN

Frances Allen was an American **computer scientist** at IBM from 1957-2002, that specialized in compiler optimization and parallel computing. She was the **first woman** to become an IBM Fellow, and the **first woman** to win the Turing Award in 2006! Allen received a Bachelors degree in mathematics from The New York State College for Teachers, and a Masters degree from the University of Michigan in 1957. Frances Allen notably led IBM's parallel computing research and worked on the Blue Gene Project: a $100 million initiative to build a state of the art massively parallel computer. Blue gene was used to study protein folding and gene development and won the 2009 National Medal of Technology and Innovation.

"Computer science is a field that came out of the problems we solved and the ones we are still trying to solve. It's in my nature to find the frontiers."

Parallel Computing

Parallel Computing is used to speed-up computing problems that can utilize multiple processors. One can compare this to a construction job; how long would it take to build a skyscraper with only a single construction worker? Hundreds of years? It is much faster to employ as many workers as possible to the project that do not interfere with each other! The same is true of many computing problems vital to scientific achievement, human longevity, and national security. Message Passing Interface (MPI) is a library that will be utilized in this challenge to allow multiple CPUs to communicate with each other. They enable multiple cores on the same machine to communicate, or completely different computers to communicate with each other! State of the art parallel computing is achieved with Hardware Accelerators such as Graphics Processing Units (GPUs), but this challenge does not involve GPUs.

More information

Frances Allen

Parallel Computing   Message Passing Interface (MPI)   Computer Science

Experiment: Your first Parallel Computing Program

What you need

A computer capable of running C or Python

Procedure

This procedure assumes the use of Python, although it is possible to perform this experiment with C:

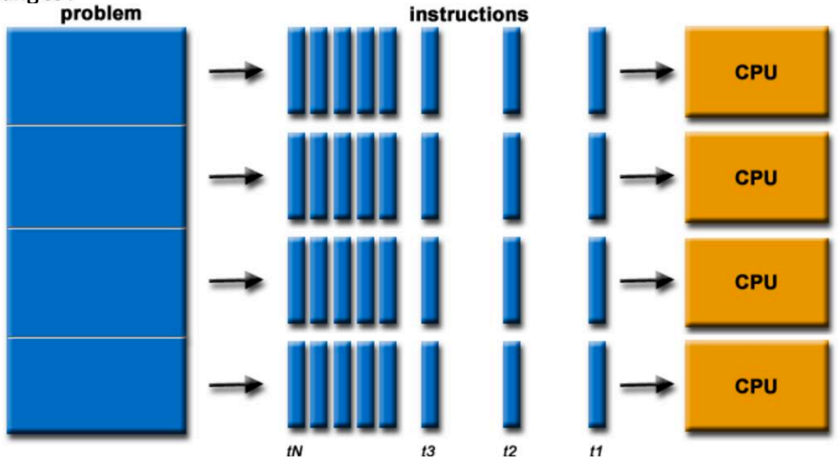
Download openMPI, Python, and the mpi4py and numpy Python packages

Create a numpy array of size 10^7 containing numbers 1,2,3,... 10^7

Calculate the sum of these numbers i) directly ii) using MPI, and measure how long it takes to do each of these operations. DO NOT count the time it takes to send the data to each processor.

Calculate the speedup: $S = \frac{t_{single}}{t_{MPI}}$ the ratio of the time it takes a single processor, divided by the time it takes multiple processors to do the same calculation.

Try experimenting with different numbers and present your findings. What happens to the speedup as the size of the numpy array changes?


problem
instructions
CPU
CPU
CPU
CPU
tN
t3
t2
t1