Parallel Programming Mini Project 01

Simulation of Stock Prices using Geometric Brownian Motion

Gaurav Tolani(201352021) Akhilesh Kumar(201351009)

Indian Institute of Information Technology, Vadodara

September, 2016

Outline

1. Problem
   - Problem Statement
   - Brownian Movement
   - Geometric Brownian Motion
   - Financial Modeling
2. Serial Code
   - Algorithm
   - Output Snapshots
   - Graphs
   - Memory Usage
3. Conclusion
   - Conclusion

Outline

1. Problem
   - Problem Statement
   - Brownian Movement
   - Geometric Brownian Motion
   - Financial Modeling
2. Serial Code
   - Algorithm
   - Output Snapshots
   - Graphs
   - Memory Usage
3. Conclusion
   - Conclusion

Problem Statement

- Doing stock prices simulation using Geometric Brownian Motion and comparing the efficiency and time of different algorithms and langauges for a number of simulations.

Outline

1. Problem
   - Problem Statement
   - Brownian Movement
   - Geometric Brownian Motion
   - Financial Modeling
2. Serial Code
   - Algorithm
   - Output Snapshots
   - Graphs
   - Memory Usage
3. Conclusion
   - Conclusion

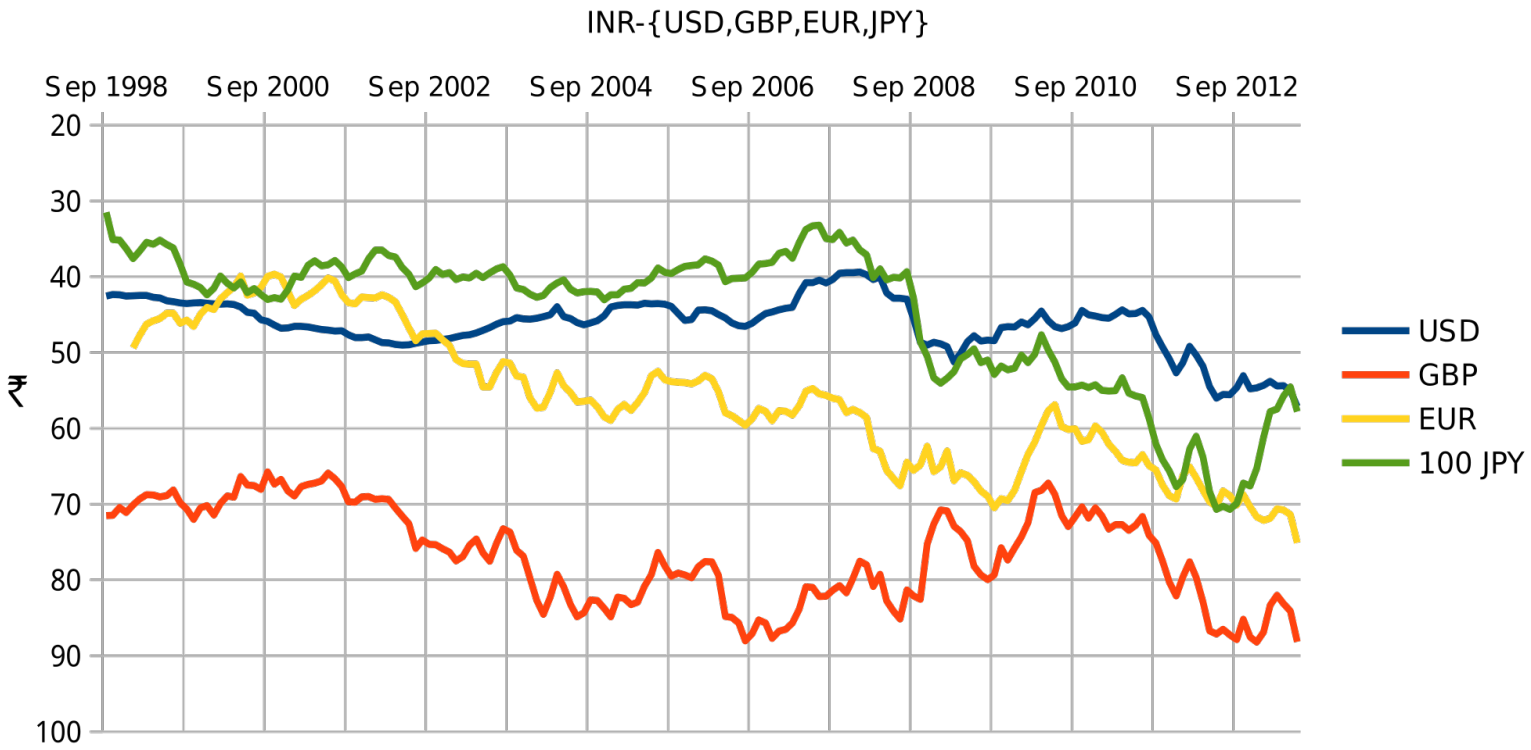
Brownian Movement

- Brownian motion is the random motion of particles suspended in a fluid (a liquid or a gas) resulting from their collision with the fast-moving atoms or molecules in the gas or liquid.

Outline

1. Problem
   - Problem Statement
   - Brownian Movement
   - Geometric Brownian Motion
   - Financial Modeling
2. Serial Code
   - Algorithm
   - Output Snapshots
   - Graphs
   - Memory Usage
3. Conclusion
   - Conclusion

Geometric Brownian Motion



- What? Simple Continuous time stochastic(probabilistic) processes.
- Why? Mostly to generate and simulate large amount of random data.
- Then Monte Carlo simulations are used to predict the future data many times over.

Outline

1. Problem
   - Problem Statement
   - Brownian Movement
   - Geometric Brownian Motion
   - Financial Modeling
2. Serial Code
   - Algorithm
   - Output Snapshots
   - Graphs
   - Memory Usage
3. Conclusion
   - Conclusion

Stock Prices Simulation

- Geometric Brownian Motion is used to generate random data over a number of stock prices.
- Given: Initial Stock Price
  % Increment in Prices Required
  % Variance
  Time
  Number of simulations
- $S_t = S_0 \times e^{\mu_t + \sigma_t z}$
  S_t is the Price of Stock at time T
  S_0 is the Initial Price of Stock
  μ and σ are the drift and volatility parameter respectively.

Outline

1. Problem
   - Problem Statement
   - Brownian Movement
   - Geometric Brownian Motion
   - Financial Modeling
2. Serial Code
   - Algorithm
   - Output Snapshots
   - Graphs
   - Memory Usage
3. Conclusion
   - Conclusion

Algorithm

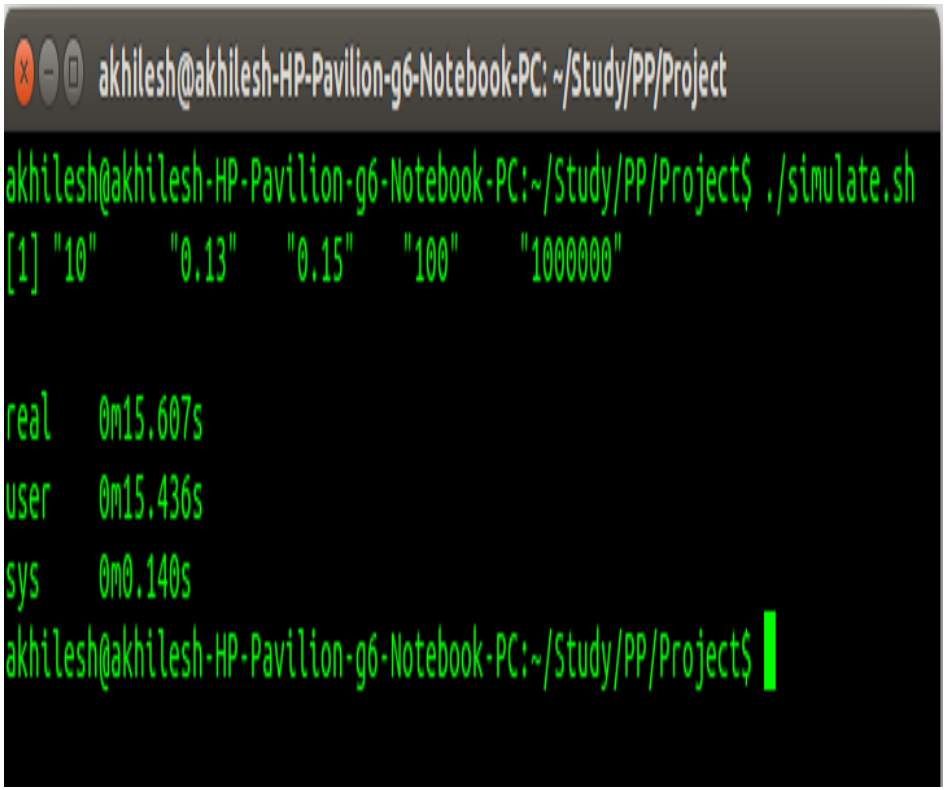
An algorithm for simulating the stock price at time t>0, given that current price at time t=0 is S_0 is as follows:

- Generate random variable z $\sim$ N(0,1)
- Set $\mu_t = (\mu - \sigma^2/2)t$ and $\sigma_t = \sigma t^{0.5}$
- Set $S_t = S_0 \times e^{\mu_t + \sigma_t z}$

Outline

1. Problem
   - Problem Statement
   - Brownian Movement
   - Geometric Brownian Motion
   - Financial Modeling
2. Serial Code
   - Algorithm
   - Output Snapshots
   - Graphs
   - Memory Usage
3. Conclusion
   - Conclusion

Output Snapshots for 'R' and 'C'



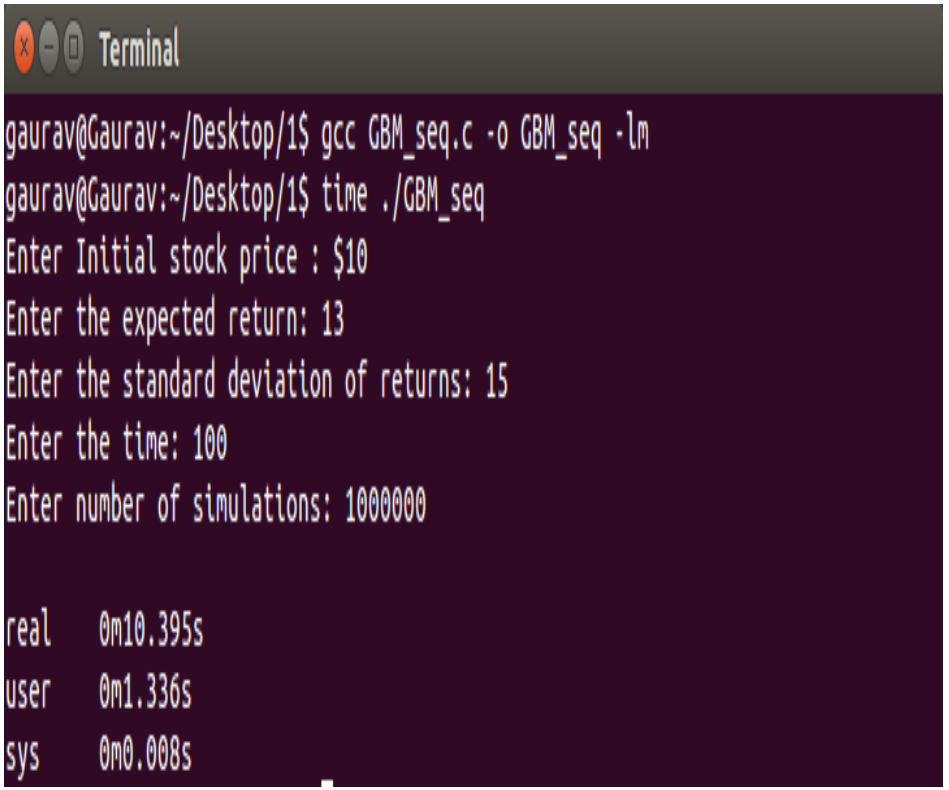


Figure : Brownian motion graph for 1000000 simulations

Outline

1. Problem
   - Problem Statement
   - Brownian Movement
   - Geometric Brownian Motion
   - Financial Modeling
2. Serial Code
   - Algorithm
   - Output Snapshots
   - Graphs
   - Memory Usage
3. Conclusion
   - Conclusion

Graphs

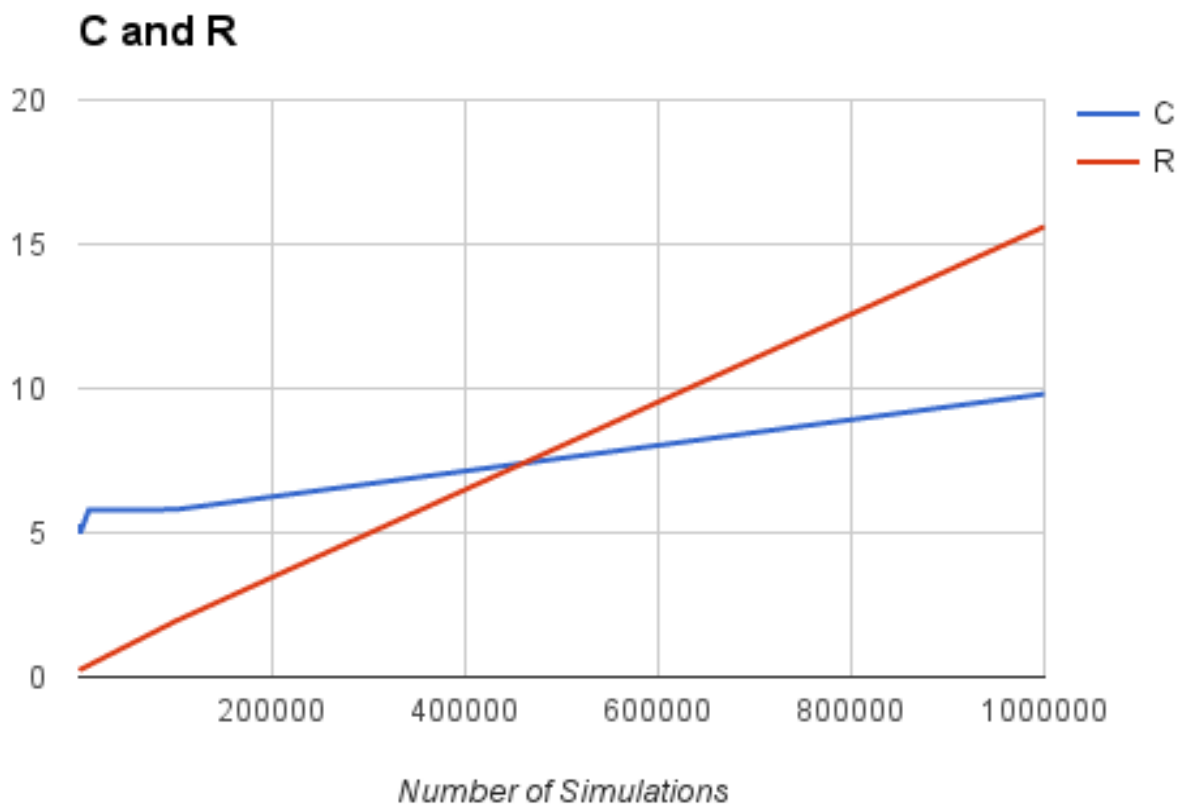
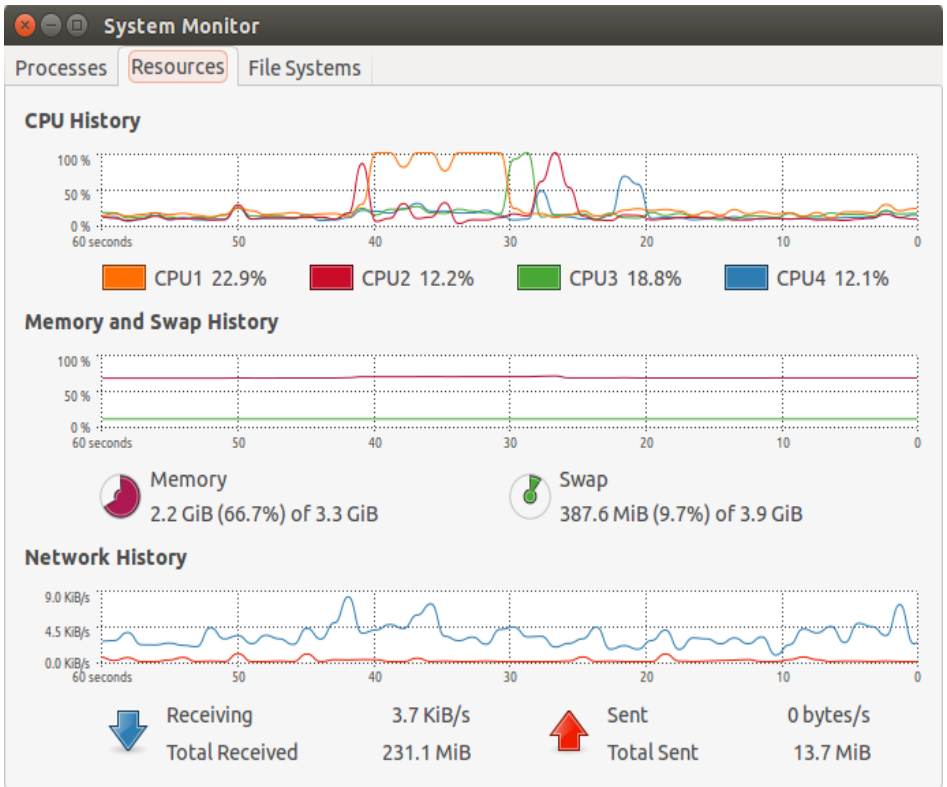


Figure : Graph Between 'C' and 'R' for different number of simulations

Outline

1. Problem
   - Problem Statement
   - Brownian Movement
   - Geometric Brownian Motion
   - Financial Modeling
2. Serial Code
   - Algorithm
   - Output Snapshots
   - Graphs
   - Memory Usage
3. Conclusion
   - Conclusion

Output Snapshots for 'R' and 'C'



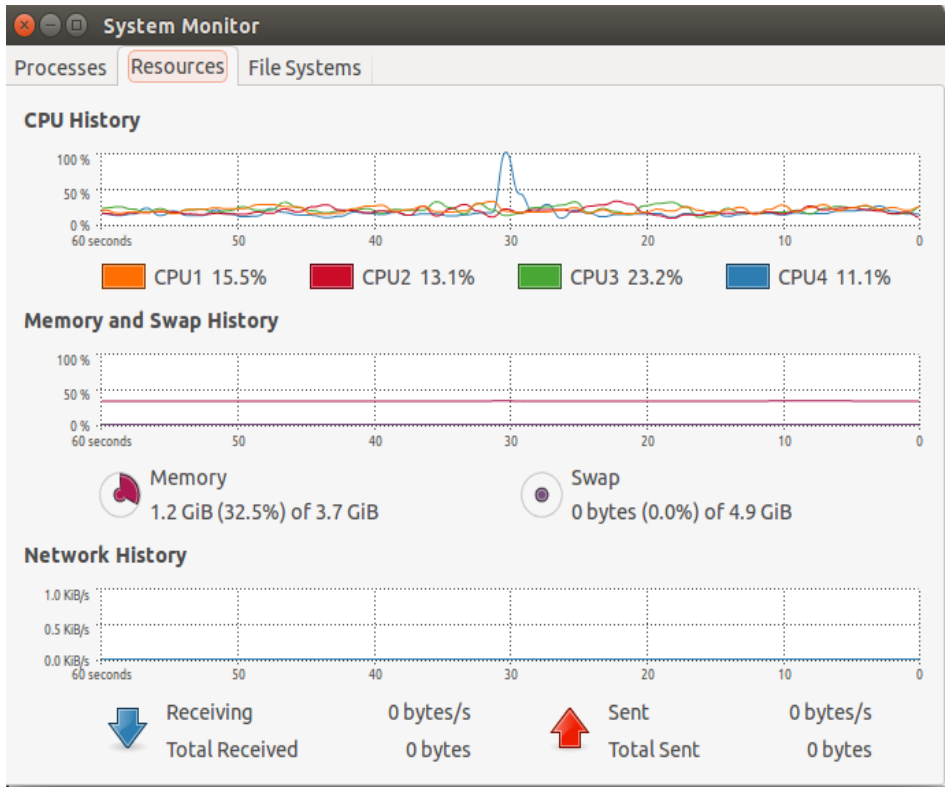


Figure : System Monitor(CPU Usage) for 1000000 simulations

Outline

1. Problem
   - Problem Statement
   - Brownian Movement
   - Geometric Brownian Motion
   - Financial Modeling
2. Serial Code
   - Algorithm
   - Output Snapshots
   - Graphs
   - Memory Usage
3. Conclusion
   - Conclusion

conclusion

- With Increasing Number of Simulations, Execution time Increases
- Time for 1000000 simulations is double in case of 'C' and approx. 14 times in 'R' as compared to 100000 simulations
- CPU usage is not at full efficiency in any of the cases which is not the ideal condition.
- It is most probable that parallelizing the code will increase the performance of the algorithm.

Thank You