SAINT LOUIS UNIVERSITY

Optimizing System Design with Market Data Simulation

CSCI 3300/CSCI 5300 Spring 2019

Executive Summary

This project aims to deliver software for evaluating the performance of a design for a parallel system architecture. Our software solution will be applicable for systems processing real-time market data. However, the solution is not limited to market data systems and may apply to other data processing systems.

Statement of Need

Performance testing is one of the stages of performance focused systems engineering projects. However, testing for performance after the system has already been developed is costly, as it is too late to change the design if test results are unacceptable. We need a way to estimate system performance as part of the design phase, which will help improve system quality and reduce development cost.

Background

Simulation modeling is used in a variety of industries to improve the performance of existing systems and to predict the performance of system designs. Some examples of applying simulation modeling are: improving performance and throughput of industrial plants, testing of structural designs, and optimization of business processes. Simulation modeling can also be applied to complex distributed computer systems. A Ticker Plant, a computer system normalizing financial market data, fits this category. This proposal discusses a framework for modelling complex distributed computer systems. The model will be applied to simulating the design of a Ticker Plant.

Scope

The goal of the project is to deliver simulation software that system engineers can use to evaluate their designs. Note that here, a *system design* is one of the inputs to our simulation software. The project will **not** address the following processes:

1. Extracting market data rates from recorded data.
2. Creating system architecture designs

Problem Description

To address project scope, two major simulation components are needed:

1. A framework for simulating distributed parallel system component layout. The framework must support easy modifications of the design layout, which will allow to test new design prototypes quickly, and
2. data rates to be used as the input to the simulation. For the purposes of this project, it is sufficient generate data rates based on various statistical distributions.

Simulation Framework

The framework for optimizing system design aims to help developers determine how to utilize available hardware resources to obtain the best system performance. Three possible use cases of the framework are shown in Figures 1, 2, and 3.

Figure 1 shows a sketch of a generic monolithic system, where all processing is done within one system component with the processing rate of C. Here, C can be either a constant or a function of the type of data being processed. The inputs to this component can be buffered. If the inputs arrive at a constant rate R, where R > C, the buffer will eventually run out of space and subsequent inputs will not be processed. We know that the arrival rates of market data fluctuate and can be greater than C at one time and less than C at another time. A monolithic system model must be supported by the simulation framework, to allow evaluation of the minimal buffer size needed to process all market data without loss.

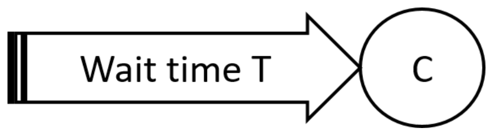

Figure 1: Monolithic System

It is theoretically possible to process all market data without loss using a monolithic system, with a sufficiently large buffer. However, the total wait time of each input will be long and the buffer may need to be unrealistically large. To reduce the wait time, system designers utilize task parallelism, where the monolithic system is broken into N components, with buffering between them. The framework needs to support designs with parallel components, as shown in Figure 2.

Each component shown in Figure 2 does less work than the one component of the monolithic system, and can process the data at a faster rate. Therefore, the wait time in each buffer is less than the wait time of the monolithic system buffer. Using simulation, we can determine the ideal setting for N and buffer sizes, given hardware resources constraints.

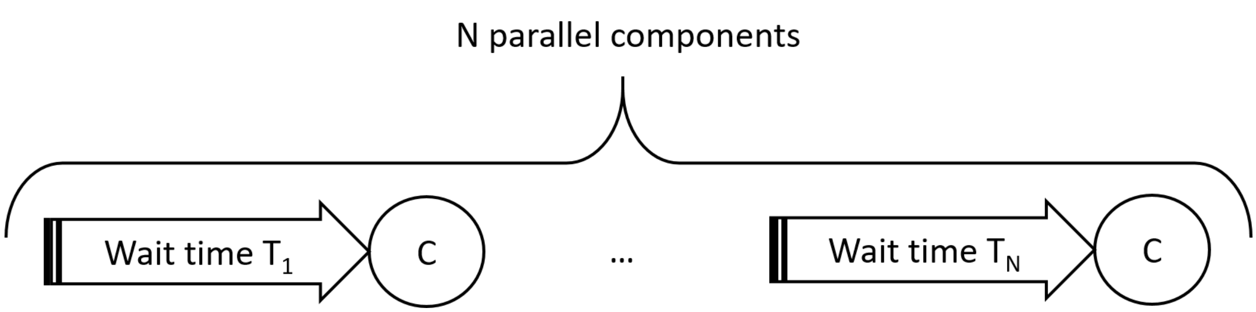

Figure 2: System with N parallel components

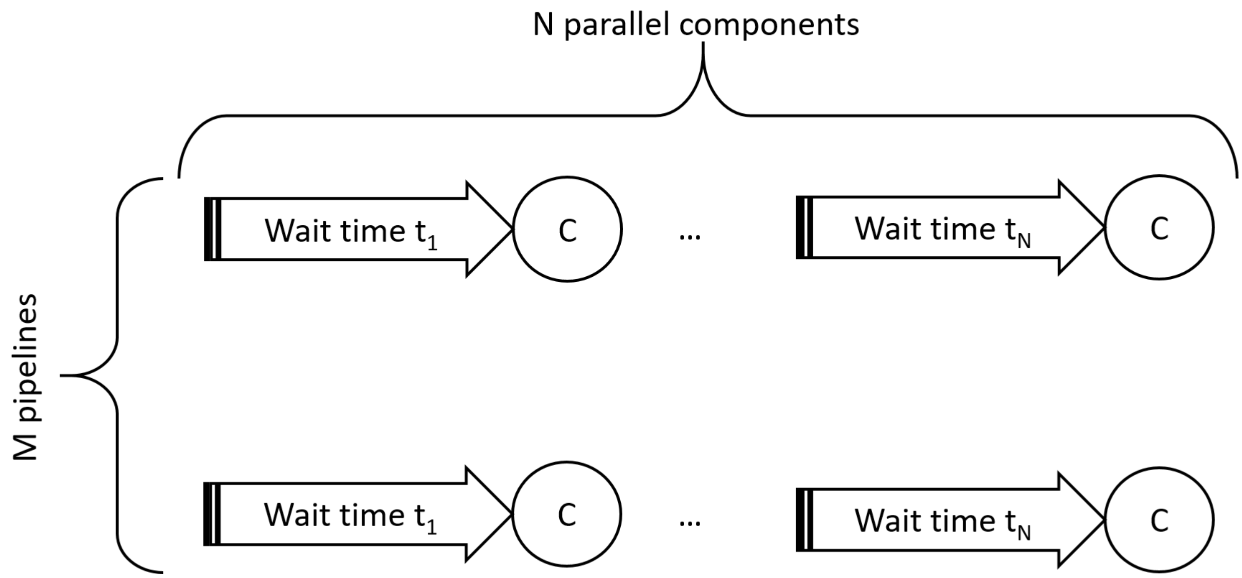

Figure 3: System with M pipelines for N parallel components

Data parallelism, where input data is distributed across multiple computing pipelines, is another way to scale up the performance of a system. Data parallelism allows to reduce the wait time for each input and to increase system throughput. Figure 3 shows a sketch of a system utilizing data parallelism with M pipelines. Simulation allows to determine the tradeoffs between increasing the number of parallel components N, versus increasing the number of pipelines M, given the limited hardware resources.

Processing time, C, of each component can be of the following forms:

- Constant time for each component (different components may have different constant processing time).
- A function of the type of data being processed (different components may have different functions for processing different types of data).

The simulation framework needs to support both modes of operation. In either case, the processing time of each component will be provided as input (or configuration) of the simulation framework.

Deliverables

The following deliverables will be provided at the end of the project:

- Executable software for running simulations
- Documentation on how to use and configure this software
- Source code for the software