Performance of a computing pipeline with data hazards and different stage time delays

A A Khusainov¹

¹ Komsomolsk-on-Amur State University, 27, Lenina prosp., Komsomolsk-on-Amur, 681013, Russia

E-mail: husainov51@yandex.ru

Abstract. Computing pipelines are used to develop various types of processors, as well as to balance the load of software systems. This work is devoted to methods for calculating the performance of pipelined processors, the stage delays of which may differ. For such pipelines, there is a problem of finding the processing time for a given number of elements, for given hazard probabilities and given stage delays. In this work, this problem is solved for the case when the input data elements have random hazards at distances of 1 and 2, and each input data element is continuously processed by the stages of the pipeline. To check the obtained analytical models, a computer model has been developed that simulates a random process of pipeline data processing. The method of the table of increments has been developed, with the help of which the inhomogeneous linear recurrent equation of the second order is constructed to obtain formulas for calculating the performance. These formulas can be used to optimize the electronic circuits of pipelined processors, as well as to develop multi-threaded pipelines that handle large volumes of requests and data.

1. Introduction
The computational pipeline consists of functional units (i.e. stages) operating in parallel, connected in series by data transmission channels. The input data elements for the pipeline can be instructions, numbers, queries, jobs. For the design of pipelines, analytical models are needed to calculate performance. By the performance of a pipeline, we mean its throughput. This is due to the widespread use of computing pipelines: for the design of signal processors [1, 2], co-processors [3, 4, 5], applications for processing big data [6]. They are used for image processing [7], for load balancing [8, 9], for processing big amounts of data and for supporting multithreaded cloud applications. Among these pipelines there are heterogeneous pipelines, which means that it becomes necessary to study formulas for calculating the performance of pipelines, the stages of which have different delays.

This work focuses on a way to design optimal pipelines, which consists of splitting computations into small steps that do not have to be equal, and then choosing to combine these steps into pipeline stages. In order to achieve maximum performance, it is necessary to have objective functions described by formulas for calculating the processing time of input data elements, which should be minimized. To find the objective functions, we use the methods from [10], which led to a formula for calculating the performance of bounded pipelines. The paper is devoted to the problem of finding the processing time for a given number of elements, for given probabilities of hazards and given stage delays. Processing time as well as stage delays are measured in milliseconds as floating-point
numbers. In Section 2, we build a computer model for a pipeline with different stage delays and arbitrary data hazards. In section 3, using the increment table method, we construct a recurrence relation for the processing time by a pipeline with given hazard probabilities at distances 1 and 2. Then we compare this time with the time obtained by a computer model. Section 4 is devoted to formulas for the performance of a pipeline with different stage delays and two hazards.

2. A method for simulating a pipeline with hazards

The linear computational pipeline (Figure 1) consists of a finite sequence of stages \( f_1, f_2, \ldots, f_m \) with delays \( \tau_1, \tau_2, \ldots, \tau_m \).

![Figure 1. Petri net of the computational pipeline, \( t_i \) - stages, \( n \) - data volume, \( m \) – depth.](image)

For each \( i \in \{1, \ldots, m\} \) time \( t_i \) contains the stage response time and the time of transmission to the channel. The number of stages \( m \) is called the pipeline depth. Each element of the input data array \( x_1, x_2, \ldots, x_n \), in turn, enters the \( f_1 \) stage, which processes it and transfers it to the output channel, for the \( f_2 \) stage, which processes it and transfers it to the output channel, for the next stage, etc. The \( f_m \) stage processes the element and transfers it to the output channel as a result of the pipeline processing the element. The number \( n \) is called the volume of the input data. The pipeline is called continuous if there are no gaps between the stages, and therefore the processing time for each individual element is equal to the sum of the delays of the stages. An element \( x_j \) is said to have a hazard with an element \( x_i \) for some \( i < j \) if it expects (and possibly uses) the result of processing the element \( x_i \). Throughout the article, we will consider linear computational continuous pipelines with data hazards, the stage delays of which may differ.

Let the probabilities \( p_{ij} \) of the hazard of \( x_j \) with \( x_i \) be given, and \( p_{ij} = 0 \) if \( i \geq j \). The processing time for data of volume \( n \) is a random variable, defined on a probability space, in which the elementary events are matrices \( (e_{ij}) \) consisting of zeros and ones, whose coefficients take values 1 with probability \( p_{ij} \). Following [11], we assume that there exist real numbers \( b_d \) such that \( p_{ij} = b_{j-i} \). The number \( b_d \) is called the probability of a hazard at a distance of \( d \). We denote \( \sigma = \sum_{i=1}^{m} \tau_i, \mu = \max \tau_i \). A random value of the processing time of \( n \) elements is a function that associates each elementary event with the processing time \( T_n \) of \( n \) elements. It is calculated by the formula \( T_k = \max (T_{k-1} + \mu, T_{k-r} + \sigma), k \leq n \), where \( r \) is the smallest number for which \( e_{k-r,k} = 1 \). If there are no such \( r \), then \( T_k = T_{k-1} + \mu \). Figure 2 shows a graph of processing time for 500 elements depending on the pipeline depth \( m = 1, \ldots, 10 \), and hazard probabilities \( b[i] \) at distances \( i \). Stages have the same delays \( \mu = t_o + t/f_m \). Sum of delays equals \( \sigma = m\mu \), where \( t_o = 2 \) equals time transmission of the result of the stage operation into the channel, \( t = 20 \) is the sum of delays without data recording time.

3. Method for constructing recurrence relations

Let us consider methods for constructing recurrence relations for calculating the processing time of pipelines with hazards. To construct these relations, it will be convenient for us to denote the maximum of the numbers \( \max(a, b) \) by \( a \lor b \). The set \( \mathbb{R} \cup \{\infty\} \), where \( \mathbb{R} \) is the set of real numbers, can be regarded as a tropical semiring in which \( \lor \) is an additive operation, and \( + \) is a multiplicative operation. Zero of the semiring is \( -\infty \). We will use this. We will denote the processing time of \( n \) elements by \( T_n \). We will write recurrent equations for \( T_n \), with indices \( n \geq 1 \). Consider pipeline with given hazard probabilities \( b_1, b_2 \). We set \( b_0 = 1 - b_1 - b_2 \). Let \( T_n \) be the mathematical expectation of a random variable equal to the processing time of \( n \geq 1 \) elements. Let \( \sigma = \sum_{i=1}^{m} \tau_i, \mu = \max_{1 \leq i \leq m} \tau_i \).
Figure 2. Graph of processing times versus depth with probabilities of hazards b[i].

**Theorem 1.** If $\sigma \geq 2\mu$, then the numbers $T_n$, $n \geq 1$, satisfy the recurrence relation

$$T_{n+2} = (\beta_0 + \beta_1)T_{n+1} + \beta_2 T_n + (1 - \beta_0)\sigma + \beta_0 \mu,$$

with initial conditions $T_1 = \sigma$, $T_2 = (1 + b_1)\sigma + (1 - b_1)\mu$. Here $\beta_1 = b_1$, $\beta_2 = (1 - b_1)b_2$, $\beta_0 = 1 - \beta_1 - \beta_2 = (1 - b_1)(1 - b_2)$. And for $\sigma \leq 2\mu$, they can be calculated by the formula

$$T_n = \sigma + (n-1)(b_1\sigma + (1 - b_1)\mu).$$

We use increment tables for the proof. Let $x_1, \ldots, x_n$ be an array of processed elements. For each $j > 2$ we denote by $\rho(j)$ the smallest distance $r \geq 1$, such that $x_j$ has a hazard with $x_{j-r}$. If $x_j$ has no hazard with previous elements, then we set $\rho(j) = 0$. In the table below, the distance $\rho(n) = 0$ means that the nth element does not hazard with previous ones. A hazard at a distance of $\rho(n) = 1$ denotes a restart. Table 1 contains an analysis of the processing time increments for the nth element.

**Table 1.** Processing time increments.

| $\rho(n-1)$ | $\rho(n)$ | $T_n$, $\sigma \geq 2\mu$ | $T_n$, $\sigma \leq 2\mu$ |
|-------------|-----------|--------------------------|--------------------------|
| 0           | 0         | $T_{n-1} + \mu$          | $T_{n-1} + \mu$          |
| 0           | 1         | $T_{n-1} + \sigma$       | $T_{n-1} + \sigma$       |
| 0           | 2         | $T_{n-2} + \sigma$       | $T_{n-1} + \mu$          |
| 1           | 2         | $T_{n-1} + \mu$          | $T_{n-1} + \mu$          |
| 2           | 2         | $T_{n-2} + \sigma$       | $T_{n-1} + \mu$          |

It gives $T_n$ values depending on previous hazards. The first two lines are obvious. Let $\sigma \geq 2\mu$. In the third line, $T_n = T_{n-1} + \mu \vee T_{n-2} + \sigma = T_{n-2} + 2\mu \vee T_{n-2} + \sigma = T_{n-2} + \sigma$. In the fourth, $T_n = T_{n-1} + \mu \vee T_{n-2} + \sigma = T_{n-2} + \sigma + \mu \vee T_{n-2} + \sigma = T_{n-1} + \mu$. In the fifth, $T_n = T_{n-1} + \mu \vee T_{n-2} + \sigma = T_{n-2} + 2\mu \vee T_{n-3} + \sigma = T_{n-2} + \sigma$. Since $T_{n-3} + \mu \leq T_{n-2}$ and $\sigma \geq 2\mu$, we get $T_n = T_{n-2} + \sigma$. The five analyzed cases are mutually exclusive, whence

$$T_n = b_0(T_{n-1} + \mu) + b_1(T_{n-1} + \sigma) + b_0b_2(T_{n-2} + \sigma) + b_1b_2(T_{n-1} + \mu) + b_2b_2(T_{n-2} + \sigma),$$
for \( n \geq 3 \). Substituting the number \( n + 2 \) instead of \( n \geq 3 \), we prove the required relation. The initial conditions are obtained from the definition of the numbers \( T_1 \) and \( T_2 \).

For \( \sigma \leq 2\mu \), the last column is constructed in the same way, and the formula for \( T_n \) is proved similarly.

Figure 3 shows the results of the application for the experimental verification of Theorem 1.

![Figure 3](image)

**Figure 3.** Change in processing time with increasing number of stages.

To calculate the processing time values, the recurrence relations and the initial values presented in Theorem 1 are used. The dashed lines show the results of the computer model, and the solid lines obtained using Theorem 1. The first graph shows the function of processing time for 1000 elements, depending on the number of stages. By adding a stage, the processing time of each element increases, and an increasing function is obtained. The probabilities of hazards are \( b_1 = 0.1, b_2 = 0.2 \). The stages have delays 1, 2, 2, 9, 4, 2, 7, 20, 1, 1. Figure 4 shows the results of checking recurrence relationships using a computer model. The graph shows processing time as a function of the number of input elements. The probability of hazards is \( b_1 = 0.3, b_2 = 0.2 \). The stage delays are equal to 1, 3, 7, 3, 3, 7, 3, 3, 3, 1. It can be seen that the second graph is close to a straight line.

![Figure 4](image)

**Figure 4.** Checking the recurrent equation using a computer model.
4. Results

Consider the recurrence relation obtained in Theorem 1. Let us find the approximate values of $T_n$ that satisfy this relation. For this purpose, we denote $S_n = T_{n+1}$. We have a linear inhomogeneous recurrent equation $S_{n+2} = (\beta_0 + \beta_1)S_{n+1} + \beta_2 S_n + (1 - \beta_0)\sigma + \beta_0\mu$ with the initial conditions $S_0 = \sigma, S_1 = (1 + b_1)\sigma + (1 - b_1)\mu$, whose coefficients $\beta_i$ are nonnegative for $i = 0, 1, 2$, and $\beta_0 + \beta_1 + \beta_2 = 1$, and the inequalities $\sigma \geq \mu \geq 0$. This inhomogeneous equation has a particular solution $S_n^{part} = \frac{(1 - \beta_0)\sigma + \beta_0\mu}{(1 + \beta_2)} \cdot n$. Let us estimate the difference between this particular solution and the exact solution of the inhomogeneous recurrent equation. Let us introduce the notation $U_n = S_n - S_n^{part}$, $n \geq 0$. The sequence of numbers $U_n$ is a solution to the corresponding homogeneous equation with the initial conditions $U_0 = S_0 - S_0^{part}$, $U_1 = S_1 - S_1^{part}$, whence for $n \geq 2$ the equality $U_n = (1 - \beta_2)U_{n-1} + \beta_2 U_{n-2}$. This equality implies that all $U_n$ belong to the segment with endpoints $U_0 = \sigma$ and $U_1 = (1 + b_1)\sigma + (1 - b_1)\mu - \frac{(1 - \beta_0)\sigma + \beta_0\mu}{1 + \beta_2}$. Since $\mu \leq U_1 \leq 2\sigma$, we get

**Corollary 1.** For a pipeline with different stage delays, the processing time of $n \geq 2$ data elements with an accuracy of $2\sigma$ is approximated by the number $\frac{(1 - \beta_0)\sigma + \beta_0\mu}{1 + \beta_2} (n - 1)$, where $\beta_0 = (1 - b_1)(1 - b_2)$ and $\beta_2 = (1 - b_1)b_2$.

The pipeline throughput is equal to the limit $P = \lim_{n \to \infty} \frac{n}{U_n}$ (the number of items processed per millisecond). Transmission delays to channels may vary. Let us look at examples where they are the same.

**Example 1.** Let the smallest partition of the calculation of the values of some arithmetic expression consist of $m$ steps with logical delays $t_1^0, t_2^0, \ldots, t_m^0$. The transmission time $t_0$ into the channel is set for the exchange between the steps. Suppose that the probabilities of hazards at distances 1 and 2 for the data stream are known. It is required to find the objective function that needs to be minimized to build a pipeline for calculating the values of this expression that has the highest performance. For a pipeline of $k$ steps, $\sigma = kt_0 + \sum_{i=1}^{k} t_i$ will be true, and therefore the problem is reduced to minimization $(1 - \beta_0)(1 - \beta_0)t_0 + \beta_0\mu = (1 - \beta_0)(\sum_{i=1}^{k} t_i + kt_0) + \beta_0(\sum_{i=1}^{k} t_i + t_0)$. Since $\sum_{i=1}^{k} t_i$ is equal to the sequential evaluation time of the expression, it is a constant. We get the objective function $(1 - \beta_0)kt_0 + \beta_0 \sum_{i=1}^{k} t_i$ to minimize processing time for a pipeline with different stage delays.

**Example 2.** Let us consider the problem of finding the optimal number of stages $m$ that make up a uniform pipeline with a total logical delay $t$, which has the highest performance for the probabilities of hazards according to data $b_1$ and $b_2$. For such pipeline, we have $t_i = \frac{t}{m}$ and $\sum_{i=1}^{m} t_i = \frac{t}{m}$. Find the minimum for $(1 - \beta_0)m t_0 + \frac{\beta_0 \sigma}{m}$, where $\beta_0 = (1 - b_1)(1 - b_2)$. The derivative $f'(m)$ equals $(1 - \beta_0)t_0 - \frac{\beta_0 \sigma}{m^2}$. Therefore, the highest performance is achieved when $m^2 = \frac{\beta_0 t_0}{(1 - \beta_0)t_0}$.

Using the fact that the solution of the inhomogeneous recurrent equation is equal to the sum of the particular solution and the general solution of the corresponding homogeneous one, we find the solution of our equation, whence we obtain:

**Corollary 2.** Under the conditions of Theorem 1, the exact value of the mathematical expectation of the processing time of $n$ elements is

$$T_n = \sigma + \frac{(b_1\sigma + (1 - b_1)\mu)(1 + \beta_2) - (1 - \beta_0)\sigma + \beta_0\mu}{(1 + \beta_2)^2} (1 - (\beta_2)^n) + \frac{(1 - \beta_0)\sigma + \beta_0\mu}{1 + \beta_2} (n - 1).$$

This formula in the special case when the stages of the pipeline have the same delays was obtained in [12].
5. Conclusion
We have built a computer model that allows us to analyze graphs showing the dependence of processing time using a pipeline on the amount of data and on stage delays. We have developed an increment table method for constructing inhomogeneous linear recurrence equations whose solutions allow us to calculate the average processing time using heterogeneous pipelines. These recurrence equations and their solutions are verified using a computer model. The results obtained can be applied both for designing a pipeline and for balancing its load. In the future, there is a generalization of the obtained formulas to pipelines with data hazards at a distance of more than 2. The resulting formulas can be applied to optimize pipelines with different stage delays.

Acknowledgments
This work was performed as a part of the Strategic Development Program at the National Educational Institutions of the Higher Education, No. 2011-PR-054.

References
[1] Chowdhury T A, Sehgal A, Kehtarnavaz N 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (Honolulu: IEEE) 2837-2840
[2] Singh D, Singh M, Hakimjon Z 2019 Signal Processing Applications Using Multidimensional Polynomial Splines (Singapore, India: Springer)
[3] Merchant F, Chattopadhyay A, Raha S, Nandy S K, Narayan R 2017 Accelerating BLAS and LAPACK via Efficient Floating Point Architecture Design Parallel Process. Lett. 27 03n04
[4] Woehrle H, Kirchner F 2018 CAEMO-A Flexible and scalable high-performance matrix algebra coprocessor for embedded reconfigurable computing systems Microprocessors and Microsystems 56 47-63
[5] Batra S 2018 Coprocessor Design for High Speed Multiplication Master Thesis (Kurukshetra, India: National Institute of Technology)
[6] Costan A 2019 From Big Data to Fast Data: Efficient Stream Data Management (France, Rennes: Éditions universitaires européennes)
[7] Gu P, Xie X, Ding Y, Chen G, Zhang W, Niu D, Xie Y 2020 iPIM: Programmable in-memory image processing accelerator using near-bank architecture. 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA) 804-817
[8] Moreno A, César E, Guevara A, Sorribes J, Margalef T 2012 Load balancing in homogeneous pipeline based applications Parallel Comput 38 125-139
[9] Moreno A, Sikora A, César E, Sorribes J, Margalef T 2017 HeDPM: load balancing of linepipeline applications on heterogeneous systems The Journal of Supercomputing 73 3738-3760
[10] Khusainov A A 2020 Performance of the bounded pipeline Inform Primen 14 87-93
[11] Emma P G, Davidson E S 1987 Characterization of Branch and Data Dependencies in Programs for Evaluating Pipeline Performance IEEE Transactions on Computers 7 859-875
[12] Khusainov A A 2021 Models for Calculating Pipeline Performance with Data Hazards Current Problems and Ways of Industry Development: Equipment and Technologies Lecture Notes in Networks and Systems 200 (Cham, Switzerland: Springer) 632-642