Comparison Between Experimental, Analytical and Simulation Model of Distributed Computation on ARM processors in High-Performance Computing

P. AUGUSTYNOWICZ, A. BURACZYŃSKA

pawel.augustynowicz@wat.edu.pl[1], a.buraczynska@mini.pw.edu.pl[2]

[1] Military University of Technology, Faculty of Cybernetics
Kaliskiego Str. 2, 00-908 Warsaw, Poland

[2] Warsaw University of Technology, Faculty of Mathematics and Information Science,
Koszykowa Str. 75, 00-662 Warsaw, Poland

The paper presents a comparison between experimental, analytical and simulation model of distributed cryptographic computation regarding password recovery with SHA 1 password hashing. The aim of this paper is compare popular mobile ARM processors with their Intel Atom analogue and determine their usefulness in cryptographic computations from High Performance Computing (HPC) point of view. During the construction process of HPC cluster, three different versions of Raspberry Pi computers were used. Then the constructed model was applied to develop an analytical and simulation models that allow calculating most influential characteristics from HPC clusters administrator's point of view. Reference model was constructed on Intel Atom processors.

Keywords: ARM processors, High-Performance Computing, Analytical Model.

1. Introduction

The paper presents three different approaches to analyse one model: experimental, analytical and simulation one. Three different approaches are needed to determinate the most substantial indicates from system's administrator point of view. Additionally, experimental and analytic models cannot prove that routeing strategy in popular open-source task-management is not suitable for heterogeneous environments. To indicate the best routeing strategy we conduct a simulation with Java Modelling Tool (JMT) software which reveals the usefulness of strategy based on assumption that nodes with the highest compute capability should be burdened the most.

2. Brief history of HPC

High-Performance Computing is a practice of combining computers into a single supercomputer managed from one host station. It is used to conduct advanced scientific and engineering computations and analysis. The areas that need the most time-consuming calculations are cryptography and code breaking.

At the beginning of 2015, Intel Xeon Phi processors were the most popular type of computational units in HPC clusters. They were valued for their high performance, a high number of cores in one unit and bit cache memory. Contrarily their power consumption was very high – on average 270 W for one processor. Intel Xeon Phi processors monopolise the HPC clusters trade and no one could thread Intel position.

In 2016 China due to political reasons was forbidden to use Intel Xeon technologies in their HPC clusters [5]. Therefore they decided to try to create a whole new architecture of HPC based on reduced instruction set computing (RISC) architecture designed by National Research Centre of Parallel Computer Engineering & Technology. Their solution was characterised by high-efficiency, low energy consumption and very high numbers of cores. It indicates that RISC architectures can be successfully applied to HPC clusters.

3. Experimental model

An experimental model of RISC architecture HPC cluster based on three different versions of Raspberry Pi computers [6]. They are valued for their efficiency and low energy consumption.

Raspberry Pi 1 was a single core computer with 700 Mhz ARM 1176 processor. The dynamic power of that solution was evaluated at about 0,2 mV / Mhz, which is a result comparable with the achieves of Intel Core i5. The rapid growth of Raspberry Pi popularity and the customer's requirement for highly efficient solutions lead to further development of Raspberry Pi Foundation product. The second version of Raspberry Pi was equipped with 900 Mhz quad-core ARM Cortex-A7 CPU. It is a 32-bit microprocessor with ARM7 instruction set, 8–64 KB of configurable L1 cache and 128 KB–1 MB of optional configurable L2 cache. In February 2016 Raspberry Pi 3 was announced. It was equipped with 1,2 GHz 64-bit quad-core ARMv8 CPU, which was the next step in the develop of series credit card sized computers. Its performance enables running sophisticated computations and applications.

Examined experimental model was presented in Figure 1. It consists of:

- one Raspberry Pi 1,
- two Raspberry Pi 2,
- two Raspberry Pi 3.

All of the abovementioned computers were connected through Ethernet into one HPC cluster managed from outside terminal station via open-lava software. Distributed computing are provided by MPICH implementation of Message Passing Interface standard [8]. The same standard is implemented in the most of the modern supercomputers. The architecture of Raspberry Pi HPC cluster is depicted in Figure 1.

The Raspberry Pi HPC cluster is applied to conduct a brute-force attack on one-way hash function SHA-1. The aim of the attack is to recover an input data from the hash function based on output data and possible passwords dictionary. From the cryptographic point of view, it is critical to use as many cores as is possible and to keep them working all the time. As a result, the most interesting characteristics of the cluster are:

- average number of arrivals in the system – it should be close to the number of cores,
- the average delay of a computational job running on the core.

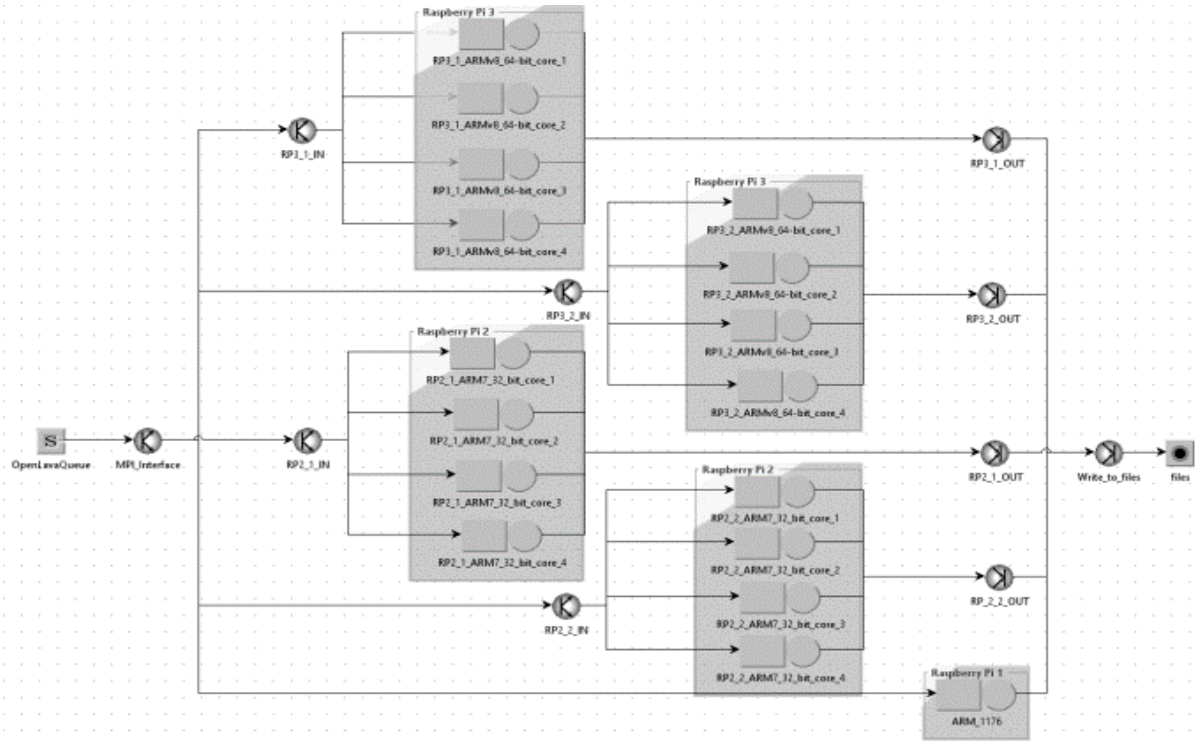

Fig. 1. Architecture of Raspberry Pi HPC cluster

Every node was given a dictionary of 100 possible passwords. Then, a value of SHA-1 function was computed 50,000 times for every possible password. The value 50,000 is connected with the way, that popular applications store user credentials, for example Microsoft Word [7]. The computations were held for ARM 1176, ARM Cortex-A7 CPU and ARMv8 from the HPC cluster and independently for Intel Atom 500 MHz from Intel Edison mobile computer to compare the results.

Tab. 1. The average time of computation for one arrival on different types of nodes

Type:	average time [s]:
ARM 1176	19,2
ARM Cortex A7	17,2
ARMv8	12,6
Intel Atom 500 MHz	16,3

Table 1 depicts the average time of computation conducted on one node on different processors type for one arrival. As it could be seen, the oldest architectures of ARM processors are slower than their Intel Atom analogue. It is connected with the architecture of processors (64 bit ARMv8 CPU is notably more efficient than 32-bit ARM Cortex A7 or 1176), the set of instruction available for different types of processors, cache access speed and processor frequency.

The results confirm that the frequency is no longer the most important and reliable indicator of the quality of processors. Effective processing large amount of cryptographic data requires very high cache access speed and support for popular cryptographic intrinsic instructions like counting Hamming weight of vector (number of one's into it) or carryless multiplication extension.

In conducted experiment after debugging and code review, it was observed that Intel assembler code generated by gcc compiler was shorter and make use of Intel intrinsic instructions which are executed in SIMD (Single Instruction Multiple Data) architectures. As a result, use of one intrisic speeds up the computations at least 2 times. That observation should explain the abyss of performance between Intel and ARM. On the other point of view, the cost of ARMv8 processors is comparable with Intel's solution.

4. Analytical model

Jackson's network is an appropriate analytical model for the above-mentioned model of HPC computation. Nevertheless, it does not consider transmission delays and influence of supervisoring system. Because of that simplification we can treat the analytical model as ideal reference point that is not achievable.

Jackson's network [1] is a sort of queueing network having nodes satisfying special conditions. Firstly, nodes must be FIFO queues having an unlimited number of waiting places. Moreover, service time in the queue obeys the exponential distribution and arrivals comes to the network as a Poisson stream with constant intensity. Precisely, Jackson's network can be formally described as follows:

$$S = < W_0, Q_0, M(\lambda_0), (M_i(\mu_i), n_i, N_i, f_i)_{i \in W} >$$

where:

- W_0 is a set of network nodes, $W_0 = W \cup \{0\}$, where W contains interior nodes denoting single queues and "0" is a node, which indicates surrounding (source and outflow of arrivals);
- Q_0 is a transition matrix representing movements in network; this matrix has the following form

$$Q_0 = \begin{matrix} \\ 0 \\ W \end{matrix} \begin{matrix} \begin{matrix} 0 & W \end{matrix} \\ \begin{bmatrix} 0 & \alpha^T \\ \beta & Q \end{bmatrix} \end{matrix},$$

  where vectors:
  $\alpha^T = (\alpha_1, \ldots, \alpha_W) = (q_{0i})_{i=1,..,W}^T$
  $\beta^T = (\beta_1, \ldots, \beta_W) = (q_{i0})_{i=1,..,W}^T$
  represent communication with outside of network and matrix $Q = [q_{ij}]$ describes probability of transitions from one node to another (precisely q_{ij} for arbitrary i, j denotes the probability transition from node i to node j);
- $M(\lambda_0)$ signifies the probability distribution of input stream, it is a Poisson distribution with intensity λ_0;
- $(M_i(\mu_i), n_i, N_i, f_i)_{i \in W}$ is a description of every single node i from the set W.

In the description of the network, one node (queueing system) is denoted by vector of four elements. These parameters denote:

- $M_i(\mu_i)$ – the probability distribution of the service time, for Jackson's network it is exponential stream with mean $1/\mu_i$;
- n_i – the number of servers in node i;
- N_i – the number of awaiting places in node i;
- f_i – the way the queue is organised (rules of inserting and removing arrivals from the queue).

It is vital to notice that for Jackson's network: $N_i = \infty$ and $f_i = FIFO$ (first input first output) for every $i \in W$. It is also assumed, that in each queue, the service time of the arrivals is drawn independent of the service times in other queues. Upon departure from queue i, the arrivals go to the next queue j randomly with the probability q_{ij}.

To find the probability distribution of the number of arrivals in nodes, we consider the following stochastic process $\xi(t) = = (\xi_i(t))_{i\in W}$, where $\xi_i(t)$ denotes the number of arrivals in node i at time t. It is possible to calculate the stationary distribution of this process using Jackson's theorem (1971). Indeed, theorem states that provided the arrival rate to each queue is such that equilibrium exists, the stationary distribution of $\xi(t)$ exists and will be given by the product-form following expressions [1], [3]:

$$\pi_k = \prod_{i=1}^{W} \pi_i(k_i), \tag{1}$$

where:

$$\pi_i(k_i) = \frac{1}{k_i!}(\rho_i)^{k_i}\pi_i(0) \tag{2}$$

for $0 \le k \le n$,

$$\pi_i(k_i) = \frac{1}{n_i! n_i^{k-n_i}}(\rho_i)^{k_i}\pi_i(0) \tag{3}$$

for $k > n$,

$$\pi_i(0) = \left(\sum_{k=0}^{n_i-1} \frac{\rho_i^k}{k!} + \frac{n_i \rho_i^{n_i}}{n_i!(n_i-\rho_i)}\right)^{-1} \tag{4}$$

and

$$\rho_i = \frac{\lambda_i}{\mu_i}. \tag{5}$$

The distribution $\pi_i(k_i)$ corresponds the stationary distribution for processes $\nu_i(t)$, which signify the number of arrivals at time t in a single queueing system $(M(\lambda_i), M(\mu_i), n_i, \infty)$. For Jackson's network, the above theorem is satisfied if matrix Q_0 is irreducible and $\rho_i = \frac{\lambda_i}{\mu_i} < n_i$ for $i \in W$, where λ_i, $i = 1, \dots, W$ are solutions of the following set of equations:

$$\lambda_j = \lambda_0 \alpha_j + \sum_{j=1}^{W} \lambda_i q_{ij}, j \in W \tag{6}$$

$$\sum_{j=1}^{W} \lambda_i q_{i0} = \lambda_0. \tag{7}$$

Other interesting values describing the network are an average number of arrivals in queue EL and the average time spent on the network by arrival EV. The Little's theorem [2], which is satisfied for Jackson's Network, define relation between this values:

$$EL = \lambda EV \tag{8}$$

where λ is an intensity of Poisson process (input stream). This equality is a base to apply the mean value analysis to the network. Indeed, using the stationary distribution of the single queueing system, the average number of arrivals in queue EL_i can be found by the definition of the expected value for random variable:

$$EL_i = \sum_{k\ge 0} k\pi_k. \tag{9}$$

This follows by Little's theorem that average time spent in the node i by arrival is equal:

$$EV_i = \frac{EL_i}{\lambda_i}. \tag{10}$$

Therefore, calculation of the expected number of arrivals in the network is a sum of EL_i:

$$EL = \sum_{i\in W} EL_i. \tag{11}$$

Likewise, the average time spent on the network by arrival can be expressed by the formula:

$$EV = \frac{EL}{\lambda_0}. \tag{12}$$

In this paragraph, parameters for Jackson's network, which correspond our experiment model will be defined. Every Raspberry Pi computer can be assumed to be single node. Therefore, there are 5 nodes representing Raspberry Pi HPC cluster. Transition matrix Q_0 is defined by zero matrix Q and vectors:
$\alpha^T = (4/17,4/17,4/17,4/17,1/17)$,
$\beta^T = (1,1,1,1,1)$.

Nodes $(M_i(\mu_i), n_i, N_i, f_i)_{i\in W}$ represent Raspberry Pi 3 for $i = 1,2$, Raspberry Pi 2 for $i = 3,4$ and Raspberry Pi 1 for $i = 5$. Construction of Raspberry Pi cluster implies that $n_i = 4$ for $i = 1,2,3,4$ and $n_i = 1$ for $i = 5$. Moreover, we assume that $N_i = \infty$ and $f_i = FIFO$.

Based on experimental results, it is claimed that $\lambda = 12{,}6/17$ and we can estimate μ_i for $i = 1,2,3,4,5$. Using values presented in Table 1 and properties of exponential distribution (the expected value of such distribution is equal $1/\mu_i$) it can be calculated that $\mu_1 = \mu_2 = 1/12{,}6$, $\mu_3 = \mu_4 = 1/17{,}2$ and $\mu_5 = 1/19{,}2$.

In this case, due to $q_{ij} = 0$ for $i,j \neq 0$ the solutions of following equations:

$$\lambda_j = \lambda_0 \alpha_j + \sum_{j=1}^{W} \lambda_i q_{ij}, j \in W \tag{13}$$

$$\sum_{j=1}^{W} \lambda_i q_{i0} = \lambda_0 \tag{14}$$

are $\lambda_j = \lambda_0 \alpha_j$ for $j = 1,2,3,4,5$.

Now, Jackson's theorem will be applied to considered network. Firstly the assumptions of the above-mentioned theorem are fulfilled. In fact, matrix Q is irreducible and $\rho_1 = = \frac{\alpha_1 \lambda_0}{\mu_1} = \frac{4}{17} \cdot \frac{12.6}{17} \cdot 12.6 < 4 = n_1$. Similarly, $\rho_2 = \rho_1 < 4 = n_2$, $\rho_3 = \frac{\alpha_3 \lambda_0}{\mu_3} < 4 = n_3$, $\rho_4 = \rho_3 < 4 = n_4$ and $\rho_5 = \frac{\alpha_5 \lambda_0}{\mu_5} < 1 = n_5$. Then using the formulas from Jackson's theorem the stationary distributions for every node and for the whole network can be found. Having these distributions is sufficient to apply the mean value analysis to our network. Now it is crucial to estimate the average number of arrivals in queue EL_i for every node. For this purpose, the following formulas [3] can be applied:

$$EL_i = \sum_{k \geq 0} k \pi_k \quad (15)$$

$$EL_i = \rho_i + \frac{\rho_i^{n_i+1}}{(n_i - \rho_i)^2 (n_i - 1)!} \pi_i(0), \quad (16)$$

$$\pi_i(0) = \left(\sum_{k=0}^{n_i - 1} \frac{\rho_i^k}{k!} + \frac{n_i \rho_i^{n_i}}{n_i! (n_i - \rho_i)} \right)^{-1}, \quad (17)$$

which are true in case π_k are stationary distributions of systems $(M(\lambda_i), M_i(\mu_i), n_i, \infty)$ with $\rho_i = \lambda_i / \mu_i$. For $i = 1, 2$, it can be estimated that $\pi_i(0) = 0{,}105$ and $EL_i = 2{,}47$, for $i = 3, 4$ that $\pi_i(0) = 0{,}038$, $EL_i = 4{,}53$ and for $i = 5$ that $\pi_i(0) = 0{,}163$, $EL_i = 5{,}13$.

Now, to find the expected time spent on the node i by arrival, it suffices to use the equality $EV_i = \frac{EL_i}{\lambda_i} = \frac{EL_i}{\alpha \lambda_0}$ for $i = 1, 2, 3, 4, 5$.

As a result of calculations: $EV_1 = EV_2 = 14{,}16$, $EV_3 = EV_4 = 25{,}98$ and $EV_5 = 117{,}66$. Finally, the value of the expected total number of arrivals in network is equal:

$$EL = \sum_{i=1}^{5} EL_i = 19{,}13 \quad (18)$$

and the average time spent on the network by arrival:

$$EV = \frac{EL}{\lambda_0} = 25{,}81. \quad (19)$$

5. Simulation model

To get the whole picture of situation the simulation for the analytical model was conducted. During the simulation, the $M|M|n$ model was considered [4]. The simulation model is simplified and the applied distributions are not the best approximation for conducted calculations. The simulation results for $M|M|n$ model vary from the experimental results because of simplifications connected with distributions.

Simulation model is based on following assumptions:

- there is one source of computational task,
- the tasks are distributed to the first free node that is available,
- every node has its own FIFO queue,
- if there is no free node, the task are assigned to the least occupied queue.

During the simulation, system response time was estimated at about 21 seconds. It is consistent with conducted experiment and points that even simplified simulations can be accurate.

Tab. 2. The average queue time in $M|M|n$ model

Type:	average queue time [s]:
Raspberry Pi 3	0,48
Raspberry Pi 2	0,47
Raspberry Pi 1	0,24

Table 2 illustrates the average queue time for different types of units. The applied fork strategy was to fork about 0,24% of arrivals to 4-cores nodes and about 0,04% to 1-core Raspberry Pi node. As a result, the slowest processing unit was laden the least. In cryptographic computations, this is desirable division and should be used.

Very interesting observation is that even for $M|G|n$ [4] simulation model results can vary from the experimental model. It is connected with the advanced queuing model of MPICH software which cannot be simulated in JMT. System response time was estimated at about 17,8 seconds which is the accurate result received for the experimental model. Simulation queuing times are also adequate for Raspberry Pi 3 and Raspberry Pi 2 units with the experiments results. For Raspberry Pi 1 unit the simulation result is smaller than the experimental one. It is the result of above-mentioned different forking strategies. The average queue time in $M|G|n$ model was depicted in Table 3.

Tab. 3. The average queue time in $M|G|n$ model

Type	average queue time [s]
Raspberry Pi 3	0,29
Raspberry Pi 2	0,34
Raspberry Pi 1	0,21

6. Conclusions

In the article, three different models of the same HPC computations were considered. It was revealed that even big simplifications necessary to satisfy Jackson network requirements would not garble the obtained results. Performing three different types of analysis is justified from

the HPC administrator's point of view. For example simulation model would show that different type of arrival forking can improve the efficiency of the whole system.

On the other hand, the ideal reference situation resulting from analytical model is not achievable, but it can indicate the direction of future changes in HPC computations managing systems. As a direct conclusion of held computations and comparisons it was revealed that MPICH software is burdened with high cost of processing computations and management.

Finally, the experiment indicate that ARM processors can be characterized by low energy consumption and decent efficiency on the background of Intel analogue solutions.

7. Bibliography

[1] Baum D., Breuer L., *An Introduction to Queueing Theory: and Matrix-Analytic Methods*, Springer, 2005.

[2] Bertsekas D., Gallager R., *Data Networks*, Prentice-Hall, Upper Saddle River, NJ, 1992.

[3] Kwiecień J., „Systemy kolejkowe markowskie", http://home.agh.edu.pl/~kwiecien/kolejki.pdf.

[4] Sztrik J., *Basic Queueing Theory*, University of Debrecen, Faculty of Informatics, 2012.

[5] „US blocks Intel from selling Xeon chips to Chinese supercomputer projects", 14 March 2017, [Online] http://www.pcworld.com/article/2908692/us-blocks-intel-from--selling-xeon-chips-to-chinese--supercomputer-projects.html

[6] "Raspberry Pi Documentation", 14 March 2017, [online] https://www.raspberrypi.org/documentation

[7] "Microsoft Office password protection", 14 March 2017, [Online] https://msdn.microsoft.com/en-us/library/cc313105.aspx

[8] "MPICH", 15 March 2017, [Online] https://www.mpich.org/

Porównanie empirycznego, analitycznego i symulacyjnego modelu obliczeń High-Performance Computing na procesorach ARM

P. AUGUSTYNOWICZ, A. BURACZYŃSKA

W artykule przedstawiono porównanie empirycznego, analitycznego i symulacyjnego modelu rozproszonych obliczeń kryptograficznych wyznaczania przeciwobrazu funkcji skrótu SHA-1. Celem niniejszej publikacji jest porównanie popularnych zastosowań mobilnych procesorów ARM z ich odpowiednikami firmy Intel oraz ocena ich użyteczności w obliczeniach kryptograficznych prowadzonych przy wykorzystaniu klastrów HPC. Do zbudowania klastra HPC użyte zostały trzy różne wersje minikomputerów Raspberry Pi wyposażone w procesory ARM o różnej mocy i przeznaczeniu. Następnie poza empirycznym klastrem skonstruowano modele analityczny i symulacyjny wyżej wymienionych obliczeń, tak aby uzyskać pełen obraz możliwości architektury procesorów ARM.

Słowa kluczowe: procesory ARM, obliczenia wysokowydajne, model analityczny.