Received August 18, 2021, accepted September 14, 2021, date of publication October 11, 2021, date of current version October 26, 2021.
*Digital Object Identifier 10.1109/ACCESS.2021.3119397*

# Quantitative Server Sizing Model for Performance Satisfaction in Secure U2L Migration

**HUN JOE[1], SUNGJIN KIM[2], AND BRENT BYUNGHOON KANG[1], (Member, IEEE)**
[1]Graduate School of Information Security, Korea Advanced Institute of Science and Technology (KAIST), Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
[2]Department of Intelligent System Engineering, Cheju Halla University, Halladaehak-ro, Jeju-si 63092, Republic of Korea
Corresponding author: Brent Byunghoon Kang (brentkang@kaist.ac.kr)

This research was supported in part by the MSIP (Ministry of Science, ICT, and Future Planning), Korea, under the Human Resource Development Project for Brain Scouting Program (IITP-2017-0-01889) and Research on Security of 5G-data-intensive Computing Platforms Based on Disaggregated Architecture (IITP-2020-0-00666).

**ABSTRACT** There are many challenges in measuring capacity using metrics such as transactions per minute (TPM) and operation per second (OPS) for all server hardware, which are becoming increasingly obsolete due to the shortening of the lifecycle of hardware and the advent of microprocessors. Instead, the results of accredited performance measurements are used to measure these standards, which are further used as references in the capacity measuring methods employed by industries. Generally, in industries, the capacity of a web application server is defined by OPS, for which no clear transition criterion exists for calculating tpmC using an empirical verification method. Considering secure Unix to Linux (U2L) x86-based server migration, there are no methods to compare and verify the max-jOPS value of Standard Performance Evaluation Corp., which is an industry-recognized performance measurement standard, to the Unix-based benchmark tpmC. In this study, a scenario-based U2L migration was empirically verified by analyzing and comparing pre-to-post with the interpretation of a census statistical system log data, which was conducted on approximately 1.7M households over 21 days with 25,288 maximum concurrent users. We present the correlation through pre-to-post comparison and analysis of U2L for each census statistical system log data by measuring the maximum CPU utilization as U2L migration between heterogeneous CPUs. The correlation is applied to OPS using the tpmC value of TPC-C proportional equation and quantified as a derived conversion ratio of OPS to max-jOPS. Consequently, we formulated and normalized an arithmetic expression, resulting in a CPU core conversion ratio of 0.165 as facile from a Unix-based legacy platform to an x86-based server. Therefore, we propose a new server sizing model for secure U2L migration between heterogeneous CPU architectures, which results in an average of 14.3% improvement in data processing time.

**INDEX TERMS** Server sizing, capacity planning, Unix to Linux (U2L) migration.

## I. INTRODUCTION

In recent years, the computing transition paradigm has been widely adapted for heterogeneous platforms, such as artificial intelligence, cloud computing, big data analysis, and the Internet of Things, in Unix to Linux migration (U2L), which has contributed to an explosive increase in data. Since the early 2000s, cloud services have expanded to all industries, exemplified by the emergence of mega-cloud providers such as Amazon AWS [1], MS Azure [2], and Google CGP [3]. The interest in secure U2L migration as the next-generation system is considerably increasing, transformed by market requests without vendor dependency and by the technological expansion of open-source software. The move to the Linux system architecture has been driven by the rapid development of x86-based CPU performance and the growing demand for major virtualization solutions, such as the maturity of virtualization technology to accommodate enterprise services and the need minimize dependency on Unix platforms owing to its costly architecture. Therefore, scale-out-based infrastructure and open-source software are being rapidly adopted in the core business areas of Internet companies. This x86 proliferation was motivated by the introduction of x86 servers in the information system, which handle large transactions in large corporations, financial institutions, and public institutions. Linux's simple thread structure has excellent performance in the Java environment, and Linux kernel independence, unlike Unix, supports various CPUs and does not depend on

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Zakarya.

fixed or specific hardware. In addition, it has the advantage of utilizing the hardware speed through various platform settings, which is the basis for scenario-based U2L migration. However, there is a lack of reliable cases of U2L migration results regarding the optimal method to access the information system operating in the existing legacy platform.

For stable and successful U2L migration, detailed work of the target system (i.e., performance and server sizing design), such as CPU/memory/disk/network throughput, are required. Therefore, the Transaction Process Performance Council (TPC) [4], which is the most representative internationally certified organization and credibly responsible for the evaluation of transaction processing and databases, provides performance measurement standards such as TPC-C (tpmC) and TPC-H (QphH@Size). In the case of WEB and WAS servers that run on Java applications, the SPECjbb2015 [5] metric max-jOPS benchmark of the Standard Performance Evaluation Corp (SPEC) [6] was adopted. These committees provide international benchmarks and tools to objectively measure the performance of the hardware included in information systems. However, there are limitations to the various performance measurement standards that are utilized as benchmarks in the industry.

First, tpmC and operation per second (OPS), which are commonly used on the legacy platforms of public institutions, cannot easily be converted to max-jOPS because an appropriate calculation method for this does not exist at present. Thus, for U2L migration, server sizing should be examined after additional performance testing of the target system. However, it is practically impossible to measure the performance of a system that operates in real time, such as a 24 × 365 non-disruptive information system that processes large transactions.

Second, the performance measurement standard of SPECjbb family is most commonly used to measure the performance of a single x86-based server; however, it has not been verified in large-scale operating environments such as cloud-based computing featuring a multi-cluster and virtualization environment. Thus, a comparative verification is essential. Moreover, it is difficult to verify the performance measurement data in accurate scale calculations for secure U2L migration.

In this study, we present an approach that provides justification for solving the existing problems.

- First, we introduce a scenario-based U2L migration methodology and research model.
- Second, we aim to establish an equation for a novel server-sizing method, based on specialized variables, using the interpretation of census statistical system log files for secure U2L migration. This is performed by comparing and verifying the pre-to-post CPU utilization and the conversion ratio from tpmC to max-jOPS.
- Finally, we formulate and normalize an arithmetic expression based on the performance criteria of other hardware vendors. This shows the confidence index of attributes and variables for secure U2L migration.

The remainder of this paper is organized as follows. Section II provides the related work. Section III presents an overview of the proposed approach and outlines the details of the model. Section IV discusses the evaluations conducted and the results obtained. Section V discusses the limitations of the proposed model. Finally, Section VI concludes the paper.

## II. RELATED WORK

Most research models for server sizing focus on performance evaluation and workload estimation, which is performed temporarily for a particular system or server and is called server scale performance estimation owing to its predictability. The estimation accuracy is significantly influenced by the estimation methods used. To derive an accurate server sizing, it is necessary to apply an appropriate formula. The various options can be classified as follows: calculation, reference, and simulation. Primarily, a reference method is used to estimate the server sizing of a new system based on previously collected actual log data of a similar system. However, it is difficult to achieve precise results when referencing the performance of a similar system compared with estimating the performance using actual or arithmetic methods. Therefore, it is essential to apply a simulation method to estimate server sizing based on the interpretation of system log files.

### A. APPROACH AND LIMITATION OF THE LEGACY

Industries consider historical usage precedents for secure U2L migrations [7]. However, current methods of server sizing do not establish the accuracy of these dimensioning requirements. For this reason, Pereira [8] sought to prove the correlation between historical usage precedents and the level of confidence on the capacity plan for a cloud computing workload, which considers an architectural meta-model, a method, a sizing equation, and a confidence index. This represents an early stage of research, with the following proposed timelines: assessment of hypotheses and data collection by 2021; proposal elaboration and field application and validation of the proposed model by 2022; thesis compilation, writing, and review and presentation by 2023.

Bo *et al.* [9], using the reference method, emphasized the performance evaluation of the load and transaction processing capabilities between x86 servers and minicomputers. No hardware manufacturers, including IBM, HPE, and DELL, provide tpmC performance measurements [10]. In some cases, these measurements were calculated using the NUMA architecture and standard benchmark-provided TPC-C. Their method analyzes the CPU utilization and transactions per second (TPS) for up to 700 concurrent users on two and three nodes, compares the corresponding minicomputers and x86 servers against standard benchmark tpmC, and proposes a matching server with similar performance. This approach is based on the calculation of similar models using a simple comparative analysis. For example, the CPU utilization of the minicomputer is 20% (two-node) and 30% (three-node), which is higher than that of the x86 server in terms of measured transactions. Because their method does not use actual data from a large operating environment, the lack of sufficient experimental data such as application complexity, network throughput, and n nodes make it difficult to apply this procedure to a large-scale environment.

Somasekaram [11] and Müller *et al.* [12] estimated the SAP® ERP system for non-disruptive enterprise core tasks only. The performance measurement for x86-server-based U2L has not been studied. Park *et al.* [13] calculated the size under the assumption of server introduction and measured the size of an x86 server using the SPECjbb2015 international certified tool. The error, that is, the difference between the scale calculation value and the value measured by the mathematical calculation method, was found to be 7.6%. This is a significant result, confirming that the formula and revision value of the server sizing are properly calculated.

In the enterprise architecture, an appropriate configuration for proof-of-concept testing exists through tools such as vmstat, iostat, netstat, mpstat, and sar performance monitoring in the legacy platform mentioned in the white paper [14] "Migration from UNIX/RISC and Mainframe to Intel-based Solutions." The authors identified peak requirements with benchmark data (from TPC-C or SPECjbb) and used them as sizing tools. However, it is both time-consuming and expensive to obtain performance estimations of the exact server sizing.

In the public institutions of Korea, the standard benchmark for server sizing proposes and applies the arithmetic numerical calculation method proposed in TTA [15] "A Guideline for Hardware Sizing of Information System." However, as it is difficult to provide accurate evidence for revision values, companies providing cloud services generally apply reference methods.

After considering the characteristics of the work of public institutions, the Korean government's National Information Resource Services (NIRS) [16] suggests nine types of servers according to the types of work, as summarized in Table 1, which reference the basic specifications and measure the server sizing that matches the business characteristics.

This presents a uniform guideline wherein a simple homepage is categorized as small, and an internal business system is categorized as medium. Limited references and resources have been used for server sizing in secure U2L migration. In addition, most research papers and case studies [7], [9]–[15] have presented server sizing with hypothetical data regardless of the CPU conversion ratio from Unix to x86-based servers. There are many studies related to server sizing with a CPU conversion ratio that lacks normalized formulas. Furthermore, the papers [22]–[28], [32]–[36] presented a CPU performance estimation that does not take into account U2L migration. Therefore, even if it is time-consuming and cost-intensive to obtain accurate calculation results, it is necessary to consider the secure U2L migration after the calculation by first applying the simulation method based on the measurement data. In this study, we present optimal methods for the conversion of tpmC and max-jOPS. Additionally, we formulate and normalize an arithmetic expression based on the CPU conversion ratio as facile from Unix to x86-based servers.

**TABLE 1. Server type of business services from NIRS.**

| Type | | CPU | MEM | Description |
|---|---|---|---|---|
| Entry | #1 | 2 | 4GB | Small business WEB/WAS Server (for simple homepage) |
| | #2 | 2 | 6GB | Small business WEB/WAS server requiring high SPEC. memory |
| | #3 | 2 | 8GB | Small/Medium business sized WEB/WAS Server, requiring high SPEC. memory |
| Mid. | #4 | 4 | 8GB | General business WEB/WAS/DB server (internal business system) |
| | #5 | 4 | 12GB | Small/Medium business sized WEB/WAS/DB server, requiring high SPEC. memory |
| | #6 | 4 | 16GB | DB server, requiring large CPU |
| Ent. | #7 | 8 | 16GB | DB server, requiring large CPU |
| | #8 | 8 | 24GB | Medium DB server, requiring heavy CPU and high SPEC. memory |
| | #9 | 8 | 32GB | Large DB server, requiring large CPU and high SPEC. memory |

## *B. INDUSTRY STANDARD*

Internationally, each piece of hardware (servers, storage, etc.) derives its performance figures from standards provided by recognized commissions such as TPC®, SPEC®, and Storage Performance Council (SPC®). Typical TPC subcommittees include TPC-C, which specializes in the performance measurement of online query transaction processing (OLTP) of business servers (the measure tpmC defines transaction processing and database benchmarks) [17], and TPC-H [18], which focuses on performance measurement in a complex business analysis application environment. The measured value is QphH@Size, which is a TPC-H composite query-per-hour performance metric. This reflects a variety of measures (performance) for processing queries in specific systems, including providing the database size for processing queries, processing power for a single stream of queries, and indicating the throughput for queries requested by multiple concurrent users.

SPECjbb is a software benchmarking tool developed by SPEC to measure the performance of systems running Java applications. Currently, the SPECjbb2015 [19] has been used. It consists of three components: backends (BE), which contain business logic and data, transaction injector (TxI), which issues transaction requests, and a controller (Ctr). In these components, *Composite* indicates that all components run on a Java virtual machine (JVM), running on one host. *MultiJVM* means that all components are on one host, but each runs a separate JVM. *Distributed* means components are on different hosts but networked, each running a JVM. The SPEC recommends the SPECjbb2015 composite metric max-jOPS to ensure the stability and availability of business services, where the application runs in the operating environment. Therefore, the experimental environment for comparative analysis was applied to a census statistical system with the same architecture.

## III. RESEARCH MODEL AND VARIABLES

The values of operation per second (OPS) for the industry standard are not provided by all hardware vendors. However, web application servers (AP servers) are sized considering transactions per second (TPS) for the expected maximum number of concurrent users, CPU performance increase rate, and maximum utilization rate. In this study, we present a new equation for secure U2L migration, which is then verified by analyzing and comparing the performance ratio of UNIX/RISC CPUs of IBM® PowerPC® and x86-based CPUs of Intel® Xeon® using the real log data of the census statistical system, where the U2L transition ratio was correlated with OPS through the TPC-C proportional equation and max-jOPS based on U2L migration.

### A. RESEARCH MODEL

This study developed a research model that adapted a simulation of the three-way methods for server sizing based on CPU cores, which is presented as a simple server sizing method from the legacy formulas of applying complex equations. Thus, as an integer of the final required CPU cores is $C_{req}$, which is arithmetically calculated by multiplying all the values from $C_{i+0}$ to $C_{i+5}$, as summarized in Table 14.

Formally, the value of the peak CPU utilization is denoted as $M$ for a maximum of k peak value runs as transaction per minute within 21 d. When measurements are represented as a set of runs $M_{max} = \{M_1, M_2, \ldots, M_k\}$, continuous $f$ in the closed section $[a, b]$ must have the maximum value$f(x)$, and each run is represented as a set of maximum values of all measurements computed as follows:

(1). We set $C_{i+0} \subset M = \overset{max}{x \in |a, b|} f(x) \times C_{oc} \times C_{wa}$ where $f$ is continuous in $x_0$, the random number $x \in \{x_0 - \delta, x_0 + \delta\} \bigcap |a, b|$ and $|f(x) - f(x_0)| < 1$ is satisfied for $\delta > 0$. Set $A$ is a random number, and $f$ for each $x \in A$ is $A \bigcap I_x$ bounded, and the set of the open interval $I_x$ is $\{I_x \,|x \in A\}$. The set as an open, converging, and compact set satisfies $A \subset U_{k=1}^{n} I_{xk}$ within limited open intervals $I_{x1}, I_{x2}, \cdots, I_{xn}$. Thus, $f$ for each $k = 1, 2, \cdots, n$ is $A \bigcap I_{xk}$ bounded $|f(x)| < M_k, \forall x \in A \bigcap I_{xk}$ and satisfied for $M_k > 0$. Therefore, the maximum $M = \overset{max}{1 < k < n} M_k$ is $|f(x)| < M, \forall x \in A$, where the multiple $C_{oc}$ is the operating cores (from the legacy platform), and the multiple $C_{wa}$ is the CPU utilization allowance for standard spare.

(2). We multiply (1) by $C_{i+1} \subset C_{tc}$, where $V(t_0)$ is the initial value, $V(t_n)$ is the final value, $t_n - t_0$ is the number of years, and $C_{tc}$ is the compounded growth rate of tpmC. The calculation is as follows:

$$C_{tc}(t_0, t_n) = \left(\frac{V(t_n)}{V(t_0)}\right)^{\frac{1}{t_n - t_0}} - 1 \quad (1)$$

(3). We multiply (2) by $C_{i+2} \subset C_{hr}$, which is the performance difference of tpmC per core for each machine, and $C_{hr}$ is the heterogeneous platform transition rate tpmC in Table 15. The calculation is as follows:

$$C_{hr} = \left(\frac{tpmC_{unix}}{tpmC_{\times 86}}\right) \quad (2)$$

(4). We multiply (3) by $C_{i+3} \subset C_{pm}$, which is the ratio of probable concurrent users to the maximum concurrent users of the legacy servers. The calculation is as follows:

$$C_{pm} = \left(\frac{users_{prob}}{users_{max}}\right) \quad (3)$$

(5). We multiply (4) by $C_{i+4} \subset C_{mc}$, which is the multi-node revision, where $C_{i+5} \subset C_{vo}$ is the virtualization overload. Therefore, the general expression for U2L migration, which is server sizing from Unix to x86-based servers, is given as:

$$f(x) = \prod_{i=0}^{n} C_i \approx \lceil x' \rceil \quad (4)$$

(6). Finally, to obtain the integer core value, the ceiling function $\lceil - \rceil \; \mathbb{R} \to \mathbb{N}$ is applied as follows:

$$\lceil x' \rceil = min\left\{n \in \mathbb{Z} : n \geq x'\right\} \quad (5)$$

Thus, the ceiling function of $x'$ is an integer core equal to or greater than $x'$; ideally, $x' \approx C_{req}$.

### B. CENSUS DATA SELECTION AND MIGRATION CASE

To achieve empirical results to accelerate secure U2L migration, it is necessary to compare and verify the collected system log data for server sizing, which is chosen as the data from the Population and Housing Census of the Korea National Statistical Office [20], a statistical survey with a high participation rate (48.54% in the case of the x86-based census statistical system). The census is a representative online survey system in which 1,750,427 out of 3,605,973 people responded, 20% of the total population of 51,073,542, and the maximum number of concurrent users was 25,288. This system is designed to run the key core survey computing system of the census to identify all population, households, and housing across the country and provide comprehensive data for the scale, structure, distribution, economic, and social characteristics of the subjects. The population census has been conducted every five years since 1925 and housing surveys have been carried out since 1960.

The basic U2L transition facts are as follows:

- Unix-based census statistical system: 50 questions (19 for the short form and 31 for the long form)
- x86-based census statistical system: 52 questions (20 for the short form and 32 for the long form)

The data are summarized in Tables 2 and 3. The report is called Population Projections for Province (2017–2047) [21], and it was distributed in June 2019. It uses the census statistical survey system and has been published on a regular basis.

Thus, there are 50 and 52 census survey items for the Unix and x86 versions, respectively. They have similar stability and

**TABLE 2.** Unix vs. x86-based census survey comparison.

| | Unix-based census | x86-based census |
|---|---|---|
| Sampling | Sample size (in Korea) : 10% of all population | Sample size (in Korea) : 20% of all households |
| Population | 47,932,951 | 51,073,542 |
| Survey | Sample survey: 50 items | Sample survey: 52 items |
| Online Participation | (47.9%), 2,295,988 out of 4,793,295 | (48.54%), 1,750,427 out of 3,605,973 |
| Collected Data Period | 21 days | 21 days |

performance under comparable workloads due to the application function and structure based on similar implementations (Java Heap Size and JDK Version). The application is a reuse-oriented program for a pre-to-post census statistical system. Therefore, the U2L migration configuration was implemented because of the similarity in collecting system log data to extract general empirical results from the analysis. In addition, the online participation of the census survey was 1,750,427 out of 3,605,973 at 20% of all 18,029,865 Korean households for 21 d, where 25,288 maximum concurrent users were measured within a millisecond of time, firing a request to the pre-to-post census servers. However, the considered sample size was applied to the Unix-based census and the x86-based census, being 10% of the national population and 20% of all households, respectively. In this study, the quantitative approach is analyzed in terms of relative performance, which is estimated using the collected system log data for secure U2L migration.

The purpose of this study is to develop an approach that suggests a valid basis for secure U2L migration by comparing and analyzing the performance of the pre-to-post census statistical systems rather than focusing on the statistical survey itself.

### C. U2L MIGRATION VARIABLES

The essence of U2L migration is to achieve mission-critical custom applications operating in the existing Unix-based legacy platform transitioning to x86-based servers running Linux, which provides better performance and reliability.

#### 1) SCENARIO-BASED U2L MIGRATION METHODS

The approach is mostly at Level #1, which changes only the simplest hardware and OS domain, which can be easily accessed, and Level #2, which involves analyzing data and conversions for the WEB/WAS domain and resource effectiveness. A Level #3 step changes the savings and efficient infrastructure architecture of the DBMS domain. The census statistical system in this study was applied to Level #2 to minimize the risk of change, and we are in the process of developing Level #3.

#### 2) U2L MIGRATION IN THE INDUSTRY

The modeling of server sizing refers to a similar server performance estimation for U2L migration, thus resulting in the Unix-based server cores to x86-based server cores = 1 to 1, using experienced traditional server sizing approaches. This is not in compliance with the optimal correlation for conversion between tpmC and max-jOPS owing to the lack of elaboration. The historical usage precedents for U2L migration do not establish the accuracy of these dimensioning server sizing approaches [7], [8].

The major enterprises in Korea have the same issues in U2L migration regarding similar systems and the use of estimation calculation methods. In the public sector, TTA [15] suggested a proper server sizing method for U2L migration but did not verify whether it is accurate to apply an arithmetic numerical-based method rather than one based on collected system log data. Thus, benchmark studies are usually applied to secure U2L migration.

In many previous studies [9]–[15], [22]–[28], [32]–[36], the results were only measured on a single server and applied differently, that is, these were not designed to run core business but as an objective method rather than for large-scale selected data in a three-tier configuration infrastructure.

## IV. IMPLEMENTATION AND EVALUATION

In this section, we present and analyze the results of the implementation and evaluation. First, the AP server selection and migration case of the experimental environment are described. Second, we present the U2L migration transition and empirically verify the results. Finally, we present a new performance measurement standard for "TPCC" (TPC-C values of other CPUs are estimated in accordance with the same principle of rPerf [29]) benchmark migrating from Unix to x86-based servers, where the U2L transition ratio is based on the scenario-based U2L migration methodology between OPS and max-jOPS.

### A. EXPERIMENTAL ENVIRONMENT

The initial server sizing of online Unix-based census applied a calculation method to estimate the preliminary server sizing because of the difficulties in validation; in comparison, the x86-based census applied a reference method. The first online survey system that applied the census was for an internet survey, a mobile survey, and a computer-aided personal interview survey. The corresponding hardware configurations are listed in Table 4.

### B. EVALUATION RESULTS

For our analysis, we present an empirical verification of the collected system log data for the pre-to-post comparison of the two census statistical systems. The server sizing, which is applied to the calculation or reference method, shows the inaccuracy in terms of errors of 39.1% and 2.63% for Unix-based and x86-based census by CPU utilization, respectively. The approaches that led to the imprecise and inaccurate server sizing of the Unix-based census are as follows.

#### 1) UNIX-BASED CENSUS PERFORMANCE ANALYSIS AND RESULTS

A calculation method was applied to an online Unix-based census, as summarized in Table 5, due to the lack of validation

**TABLE 3. Unix vs. x86-based census survey items comparison.**

| Classification | | Full Survey Items (12) | Sample Survey Items (52) | |
|---|---|---|---|---|
| UN Recommend (38) | Population (24) | 1. Name<br>2. Gender<br>3. Age<br>4. Relationship<br>5. Nationality<br>6. Date of entry into Korea | 1. Name<br>2. Gender<br>3. Age<br>4. Relationship<br>5. Nationality<br>6. Date of entry into Korea<br>7. Religion[1]<br>8. Education level<br>9. Field of study[1]<br>10. Place of birth<br>11. Residence a year ago<br>12. Residence five years ago | 13. Limited Activities<br>14. Commute<br>15. Place of commute<br>16. Economic activities<br>17. Employee level<br>18. Industry<br>19. Occupation<br>20. Work location<br>21. Marital status<br>22. Marriage year<br>23. Number of biological children<br>24. Dates children were born[1] |
| | Household (8) | 1. House division | 1. House division<br>2. Number of rooms<br>3. Household type<br>4. Heating | 5. Only residential or business<br>6. Occupation<br>7. Rent<br>8. Owner of house |
| | Housing (6) | 1. Type of living<br>2. Residential gross<br>3. Year of construction 4. Land | 1. Type of housing facility<br>2. Number of rooms<br>3. Number of facilities | 4. Residential gross[2]<br>5. Year of construction[2]<br>6. Land[2] |
| Unique Items (14) | Population (10) | 1. Family origin[3] | 1. Care of Children<br>2. Transportation<br>3. Limited activities[1]<br>4. Means of transportation<br>5. Year(s) in current job | 6. Plan for additional children<br>7. Work before marriage[1]<br>8. Career disruption[1]<br>9. Social activities<br>10. Source of elderly expenses |
| | Housing (4) | | 1. Period of living<br>2. Floor information | 3. Parking<br>4. Other houses owned |

**TABLE 4. Corresponding hardware configuration.**

| #Server | Unix-based census | | | x86-based census | | |
|---|---|---|---|---|---|---|
| | WEB | WAS | DBMS | WEB | WAS | DB |
| CPU Model | IBM Power 6® | IBM Power 6® | IBM Power 6® | Intel Xeon® E5-2660v2 | Intel Xeon® E5-2660v2 | IBM Power 7® |
| Clock (GHz) | 5.0 | 5.0 | 5.0 | 2.2 | 2.2 | 3.72 |
| Total Core(s) | 32 | 32 | 64 | 108 | 204 | 60 |
| Node (s) | 4 | 2 | 1 | 27 | 34 | 2 |
| Per Core(s) | 8 | 16 | 64 | 4 | 6 | 30 |
| Mem (GB) | 64 | 128 | 1,024 | 16 | 32 | 604 |
| H/W Model | IBM p595 | IBM p595 | IBM p595 | eSlim SU7-2254R4 | eSlim SU7-2254R4 | IBM p780 |
| OS | AIX 5.3 | | | RHEL 6.5 | | |
| JAVA | JDK 1.6 | | | JDK 1.6 | | |
| Survey | Internet/Mobile/Interview | | | | | |

and verification of the selected system log data from the first online census survey. Therefore, 3,150,394 tpmC $\approx$ 32 cores were presented and compared to the IBM® Power6 server (IBM® p595, tpmC $=$ 3,379,462).

The collected system log data of the Unix-based census for our analysis, which required the relevancy of the performance measurement, were obtained from the server sizing during the period of the census survey by applying 18.82 cores and 1,987,799 tpmC of maximum CPU utilization (58.82%) compared to 32 cores, as summarized in Table 6.

As a result, the designed architecture of the AP servers had 32 cores and 3,379,462 tpmC, which shows that the reference error range per core is as high as 39.1% when using the recommended configuration. In addition, AP server

**TABLE 5. Legacy server sizing for unix-based census AP servers.**

| | Value | Result | Reference |
|---|---|---|---|
| Concurrent Users | - | 1,118 | Maximum concurrent users per a minute |
| Page view/min | 10.0% | 11,180 | Network session count for 3 times request of users per a minute |
| Web Programming Revision | 2.4% | 26,832 | Static Page : Dynamic Page = 3 : 7, (Load Factor : HTML = 1, Java 3), ((3/10)*1) + ((7 /10) * 3)) = 2.4 |
| Peak Day Workload Revision | 1.4% | 37,565 | Revision factor for peak day |
| Peak Time Revision | 1.4% | 52,591 | Revision factor for peak time |
| Response Time Revision | 2.0% | 105,181 | Revision factor for response time (MS) |
| AP complexity Revision | 3.0% | 315,544 | Revision factor for application complexity |
| Network traffic Revision | 3.0% | 946,633 | Revision factor for network traffic |
| System Spare | 1.6% | 1,514,623 | Revision Factor for the unexpected |
| Objective System Utilization Revision | 1.6% | 2,423,380 | Revision factor for objective system Utilization |
| SPECjbb | - | 2,423,380 | (OPS Revision) |
| Required total tpmC | - | 3,150,394 | OPS (SPECjbb) vs. tpmC conversion rate (tpmC = OPS ×1.3) |

[1]New survey items for x86-based census.
[2]Survey items of administrative data substitution.
[3]Items only in the full survey.

sizing, based on OPS (=1,907,933), which is a representative industry standard, and tpmC (=3,379,462) of the IBM® p595 (Power6 5.0GHz, 32 cores) server, as summarized in Table 7, is a randomly applied calculation method with a transition rate of 1.771 from OPS (IBM® p595, 32 cores). Therefore, the result (1.3) obtained from a Unix-based census server sizing with OPS revision factor is inaccurate and incomplete, as summarized in Table 5.

**TABLE 6. Unix-based CPU utilization analysis and results.**

| CPU (p595@5.0GHz) | | | Core(s) | tpmC |
|---|---|---|---|---|
| IBM® Power6 | tpmC per core(s) | | 1 | 80,779 |
| | | | 32 | 3,379,462 |
| | CPU Utilization(%) | Max. | 58.82 | |
| | | Min. | 1.00 | |
| | | Avg. | 2.24 | |
| Maximum CPU Utilization (Standard: 80%) | | | 18.82 | 1,987,799 |
| Optimization (Spare: 20%) | | | 22.58 | 2,385,359 |
| Recommend Configuration (Final Value) | | | 23 | - |

Moreover, it justifies that the AP servers, to be implemented in false performance estimation, are randomized after the AP server is sized through the revision factor coefficient in the mathematical calculation method. Therefore, the industry standard, which is based on the calculation method, has unaffordable errors and thus wastes the total cost of ownership (TCO) and returns on investment. The proposed legacy calculation theorem in TTA [15] is not as robust as desired because the error range is excessively high at 39.1%, as summarized in Tables 5 and 6. For accurate server sizing based on scale estimation, scenario-based U2L migration needs to apply the reference or simulation method by using empirical analysis and verification. Finally, we present a new U2L migration server-sizing model in the following sections.

### 2) x86-BASED CENSUS PERFORMANCE ANALYSIS AND RESULTS

The x86-based census AP servers for U2L migration were applied as a reference method for server sizing. According to IBM® p595 (Power6 5.0GHz, 64 cores), the performance data of tpmC (6,085,166) and OPS (107,359) are summarized in Table 7. IBM® server products have their own performance comparison, namely relative performance (rPerf), which is equivalent to TPC-C according to the "IBM Power Systems Performance Report [30]." It can be inferred that tpmC has a calculation method that applies a proportional expression to the rPerf values of IBM's official benchmark.

We designed and developed an architecture for the x86-based census with preliminary analyzed Unix-based (IBM® p595) log data. Moreover, the heterogeneous platform between the Unix-based and x86-based census applied the TPC-C, rPerf, and OPS performance benchmarks. In addition, we demonstrate that the maximum growth rate of concurrent users (25,000 expected, 4.1 times growth), multi-node environment revision, and system resource spare are appropriate for our server sizing approach. The corresponding value of tpmC (18,283,081) was estimated by multiplying the baseline variable with the tpmC (2,640,164) of the two-node IBM® x3650 M4 server (16 cores, 1,320,082) compared with the results for the preliminary Unix-based servers (IBM® p595, 32 cores, 3,379,462). Therefore, the AP server selection was determined to be an appropriate configuration for the infrastructure budget, performance, and U2L migration proper cases, as shown in Table 8.

As a result, based on this benchmark, the collected system log data of the x86-based census for our analysis during the census survey period, where the relevancy of the performance measurement required for the AP server sizing, was applied to 165.65 cores. In addition, 15,075,956 tpmC of maximum CPU utilization (58.82%) were compared to 204 cores and the results are summarized in Table 9. This can be estimated as 198.78 cores and 18,091,147 tpmC, which are applied to the maximum CPU utilization (%) by the standard server resource spare (20%) for a multi-node operating environment.

The error range of the server sizing, compared to the initial value, is confirmed to be below 2.63% (recommended 199 cores vs. estimated 204 cores), indicating that there is not much difference in the pre-to-post results. This indicates that the server sizing and revision factor, based on the comparison of load and transaction processing, are calculated to a sufficient accuracy for U2L migration. However, the preliminary initial server sizing is not applied to the CPU utilization of servers operating 32 cores, but only to the number of cores, which is calculated based on the revision data such as the expected total number of concurrent users, number of nodes, and system resource spare. Therefore, it can also be applied as a revision of the arbitrarily calculated minimum revision coefficient. In conclusion, server sizing, after analyzing the collected system log data, is required and applied as a simulation for U2L migration.

### 3) ANALYSIS OF THE MEASURED DATA PERFORMANCE

The census statistical systems are currently configured with IBM®-Power6®-processor-based Unix servers (IBM® p595, 32 cores) and Intel® Xeon® CPU architecture (eSlim SU72254R4, 204 cores, assembled in Korea), respectively. As shown below, we presented and analyzed the comparison of the collected system log data for 21 d, including total concurrent users, CPU utilization, response time, memory usage rate, and TPS.

Fig. 1 presents the maximum and average CPU utilization of pre-to-post census statistical system within 21 d (X axis shows maximum CPU utilization and Y axis shows the number of days), where the Unix and x86-based census AP servers were measured at 58.8% and 81.2%, respectively. The value of increase rate is 38.1%, which is caused by the high utilization at the maximum number of concurrent users (25,288).

As a result, the maximum number of concurrent users of the Unix-based census was 6,065, with a maximum TPS of 866.23 and an average of 31.85. In the case of the

**TABLE 7. TPC-C, rPerf and OPS performance benchmark.**

| Core(s) | IBM® p595 (Power6/5.0GHz) | | | | IBM® x3650 M4 (Intel® Xeon® E5-2690/2.9GHz) | | | |
|---|---|---|---|---|---|---|---|---|
| | tpmC | rPerf | OPS | OPS(JVM) | tpmC | rPerf | OPS | OPS(JVM) |
| 64 | 6,085,166[4] | 553.01[5] | 3,435,485 | 107,359 | - | - | - | - |
| 32 | 3,379,462 | 307.12 | 1,907,933 | 59,623 | - | - | - | - |
| 16 | 1,811,982 | 164.67 | 1,022,986 | 31,968 | 1,320,082[6] | - | 1,578,448 | 197,306 |
| 8 | 958,424 | 87.10 | 541,095 | 16,909 | 1,135,197 | - | 1,357,377 | 169,672 |
| 1 | 80,779 | 7.34 | 45,605 | 1,425 | 95,678 | - | 114,404 | 14,301 |

**TABLE 8. Server sizing for x86-based census application servers.**

| # Server | Server Sizing(tpmC) | Application Servers (AP Server) Selection | | | | | | | Comparison(tpmC) |
|---|---|---|---|---|---|---|---|---|---|
| | | H/W Model | CPU Model | Clock(GHz) | Total Core(s) | Node(s) | Per Core(s) | tpmC | |
| x86-based Census | 18,283,081 | eSlim SU7-2254R4 | Intel Xeon E5-2660v2 | 2.2 | 204 | 34 | 6 | 91,012 (Per core) | 18,566,448 (Final Value) |

**TABLE 9. Server sizing for x86-based census servers.**

| CPU (Intel@2.2GHz) | | | Core(s) | tpmC |
|---|---|---|---|---|
| E5-2660v2 | tpmC per Core(s) | | 1 | 91,012 |
| | | | 204 | 18,566,448 |
| | CPU Utilization(%) | Max. | 81.20 | |
| | | Min. | 1.04 | |
| | | Avg. | 1.93 | |
| Maximum. CPU Utilization (Standard: 80%) | | | 165.65 | 15,075,956 |
| Optimization (Spare: 20%) | | | 198.78 | 18,091,147 |
| Recommend Configuration (Final Value) | | | 199 | - |

**TABLE 10. Unix vs. x86-based census transition workloads.**

| DIV. | Unix-based census IBM p595 Servers | | | x86-based census Intel E5-2660v2 Servers | | |
|---|---|---|---|---|---|---|
| | Max. | Min. | Avg. | Max. | Min. | Avg. |
| Concurrent Users | 6,065 | 11 | 716 | 25,288 | 229 | 1,043 |
| CPU Utilization | 58.82 | 1.00 | 2.24 | 81.20 | 1.04 | 1.93 |
| Memory (heap) | 89.50 | 21.56 | 52.92 | 34.99 | 17.38 | 19.92 |
| TPS | 866.23 | 17.59 | 31.85 | 2677.40 | 0.8 | 2.28 |
| Response Time (MS) | 51.94 | 0.04 | 0.24 | 12.31 | 0.17 | 0.21 |

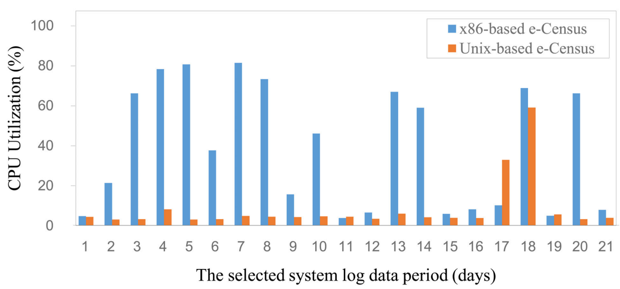



**FIGURE 1. Unix vs. x86-based census with maximum CPU utilization.**

x86-based census, the maximum number of concurrent users was 25,288 with a maximum TPS of 2677.40 and an average of 2.28. However, the response time improved by a maximum of 321.9% and an average of 14.3%, which was confirmed by the workload distribution due to the increase in multiple nodes, as summarized in Table 10.

### *C. U2L MIGRATION TRANSITION VERIFICATION*

The server sizing of the x86-based platform can be compared to max-jOPS as an industry standard of performance while applying OPS to AP servers. Thus, for a more accurate formula, we present a new server-sizing method for U2L migration based on the simulation method. This is applied to an error of 1.41% and an OPS to max-jOPS ratio of 0.07%.

#### 1) SERVER SIZING TRANSITION FROM UNIX TO x86 SERVERS

The corresponding hardware of the x86-based census servers are the eSlim SU7-2254R4 domestic servers (assembled in Korea), based on the Intel® Xeon® E5-2660v2 CPU [31], for which max-jOPS = 22,482 and tpmC = 91,012 per core, as summarized in Table 11.

#### 2) COMPARISON BETWEEN OPS AND MAX-jOPS

The first performance estimation of Intel® Xeon® E5-2690 on IBM® Power6® and the second substitution to E5-2260v2 do not provide the OPS value of E5-2660v2, officially provided by TPC-C. We calculate this by applying OPS proportionality to the E5-2690 CPU of Intel® architecture x86 servers, which provides similar and official performance tpmC. In addition, the max-jOPS of x86-based servers is compared to the server sizing by applying the performance value of OPS that matches rPerf, as summarized in Table 12.

According to the performance estimation of AP servers, as summarized in Table 12, the TPC-C proportional equation is applied to the OPS (14,301) of Intel® E5-2690 CPU to calculate and apply OPS (13,603) of Intel® E5-2660v2 CPU with a similarity comparison of 95.12%. In conclusion, the server sizing of AP servers as compared with the Unix vs. x86-based census is verified and applied to the same

[4]tpmC value of IBM® p595 5.0GHz, 64 cores: 6,085,166.
[5]rPerf value of IBM® p595 5.0GHz, 64 cores: 553.01.
[6]tpmC value of IBM® x3650M4 2.9GHz, 16 cores: 1,320,082.

**TABLE 11.** U2L migration tpmC and max-jOPS.

| CPU | Intel Xeon Gold-6154 | Intel Xeon E5-2680v4 | Intel Xeon E5-2699v4 | Intel Xeon E5-2699v3 | Intel Xeon E5-2660v2 |
|---|---|---|---|---|---|
| Clock(GHz) | @3.00 | @2.40 | @2.20 | @2.30 | @2.20 |
| Core(s) | 18 | 28 | 22 | 18 | 10 |
| Core(s) per Socket | 18 | 14 | 1 | 1 | 5 |
| CPU Socket(s) | 1 | 2 | 1 | 1 | 2 |
| Threads | 36 | 56 | 44 | 36 | 20 |
| max-jOPS | 55,364 | 77,234 | 53,945 | 44,390 | 22,482 |
| Critical-jOPS | 36,012 | 26,235 | 30,533 | 15,654 | 3,148 |
| tpmC(Core) | 149,571 | 109,755 | 94,140 | 94,120 | 91,012 |

arithmetic principle of TPC-C, and OPS to max-jOPS, which is quantified using comparative analysis data, and is computed as follows:

$$tpcOPS2 = \frac{(spTPCC \times tpcOPS1)}{coreTPCC} \quad (6)$$

In the case of IBM, the performance ratio linearly increases with increasing numbers of cores and shows similar performance characteristics[7] regardless of the workload of the WEB/WAS/DB server, as summarized in Table 13.

The maximum CPU utilization of x86-based census AP servers (eSlim SU7-2254R4 2.2GHz, 10 cores) is measured, tpmC (=15,075,956), as summarized in Table 9. OPS (14,301) and tpmC (95,678) of IBM® x3650 M4 (Intel® Xeon® E52690 2.9GHz, 16 cores) per core is summarized in Table 12, where max-jOPS (22,482) is 165.65 cores based on the Intel® E5-2660v2 10 cores. Therefore, the values of OPS (=2,253,405) and max-jOPS (=372,414) can be quantified, and the conversion ratio of OPS to max-jOPS is calculated to be ≈0.165. Thus, we formulated and normalized an arithmetic expression, which was derived based on the CPU core conversion ratio of ≈0.165, from the Unix-based legacy platform to the x86-based server. Therefore, U2L migration is proportional to the performance transition value of 0.165.

#### 3) U2L MIGRATION TRANSITION EMPIRICAL VERIFICATION

The U2L migration of the existing Unix-based legacy platform was designed and implemented for the corresponding hardware configuration based on the actual utilization, work grade, and workloads. In particular, the performance standard benchmarks of OPS, tpmC and max-jOPS are not quantitatively measured by hardware vendors due to the complexity of product line up. Therefore, in this paper, we propose a completely different and new U2L migration transition formula, where the arithmetic calculations are derived from simulation method results by the pre-to-post performance analysis of the census. In addition, we present the empirical verification of the new server-sizing model in Section III A. The research model, from Unix-based to x86-based servers as summarized in Table 14, normalizes easily per the preliminary CPU core unit. However, the number of CPU cores is converted into an integer larger than the calculated value.

In general, TTA [15] applies multi-node revision coefficients from 1.3 to 1.5, depending on the operating environment. Thus, the final value of $C_{req}$, can be simply calculated arithmetically by multiplying all the values. In addition, applying the above arithmetic formula, the required server sizing of the x86-based census is 201.16 cores, and it is converted into 204 cores (6 cores × 34-nodes) as an integer for U2L migration. This shows that the error range of the performance estimation is 1.41%.

In this study, we approached U2L migration, which is approximately in a similar application configuration excluding structure and complexity, where the experimental system log data were measured. However, the equivalent performance transition rate (=1.56) of IBM® Power® vs. Intel® Xeon® CPU was applied to U2L migration, with the representative hardware-to-hardware vendor's performance increase ratio, as summarized in Table 15. The top row contains the CPU names and tpmC per core. The values in the table represent the performance increase ratio.

As a result, the U2L migration (based on WEB and WAS) from Unix to x86-based servers was 36.0% lower than the Unix-based legacy platform, because the high-cost infrastructure to the open architecture and the data processing time was improved by an average of 14.3%, as shown in Table 16.

## V. DISCUSSION

When we started developing a new server-sizing model for U2L migration, some implementation issues were considered. First, the AP server of census, with a three-tier configuration environment, used Oracle WebLogic® based on the Java Development Kit (JDK) 1.6, and the x86-based census used Redhat®, based on Jboss® 6.7, for U2L migration. Therefore, to implement a census statistical system that collects data from the same survey items, the operation environment was implemented using a similar Java heap size and previous application source code recycling. In addition, to confirm the general results of the analysis, the data collection period was set to 21 d.

Second, the most important SPEC metric, the max-jOPS values, were measured according to the specifications of each hardware vendor. This is not expressed in units of cores but in terms of physical CPUs, which is not accurate when estimating the proportions of core units. Therefore, the NUMA architecture is generally applied. In this study, the performance ratio of the existing IBM® p595 increased

[7]The value of tpmC GAGR from IBM® Power10® [30], which was newly announced in October 2020, is presented as excluded because of the unpublished tpmC performance measurements officially.

**TABLE 12.** Comparison of CPU performance benchmark.

| CPU Model | Clock (GHz) | tpmC (core) | rPerf | OPS (JVM) | max-jOPS (10 cores) | Critical-jOPS (10 cores) | Comparison |
|---|---|---|---|---|---|---|---|
| IBM Power6 | 5.00 | 80,779 | 7.34 | 1,425 | - | - | 95.12% |
| Intel Xeon E5-2690 | 2.90 | 95,678 | - | 14,301 | - | - | |
| Intel Xeon E5-2660v2 | 2.20 | 91,012 | - | 13,603 | 22,482 | 3,148 | |

**TABLE 13.** IBM power series tpmC CAGR.

| CPU Model | Power9 | Power8 | Power7+ | Power7 | Power6+ | Power6 | Power5+ | Power5 | Power4+ | Power4 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 233,754 | 194,549 | 133,838 | 115,728 | 81,236 | 80,779 | 52,395 | 49,518 | 24,074 | 11,813 |
| | 2018 | 2014 | 2012 | 2010 | 2008 | 2007 | 2005 | 2004 | 2002 | 2001 |
| Power9 | 1 | | | | | | | | | |
| 2018 | 1 | | | | | | | | | |
| Power8 | 1,202 | 1 | | | | | | | | |
| 2014 | 0.832 | 1 | | | | | | | | |
| Power7+ | 1,747 | 1,454 | 1 | | | | | | | |
| 2012 | 0.573 | 0.688 | 1 | | | | | | | |
| Power7 | 2,020 | 1,681 | 1,156 | 1 | | | | | | |
| 2010 | 0.495 | 0.595 | 0.865 | 1 | | | | | | |
| Power6+ | 2,877 | 2,395 | 1,648 | 1,425 | 1 | | | | | |
| 2008 | 0.348 | 0.418 | 0.607 | 0.702 | 1 | | | | | |
| Power6 | 2,894 | 2,408 | 1,657 | 1,433 | 1,425 | 1 | | | | |
| 2007 | 0.346 | 0.415 | 0.604 | 0.698 | 0.994 | 1 | | | | |
| Power5+ | 4,461 | 3,713 | 2,554 | 2,209 | 1,550 | 1,542 | 1 | | | |
| 2005 | 0.224 | 0.269 | 0.391 | 0.453 | 0.645 | 0.649 | 1 | | | |
| Power5 | 4,721 | 3,929 | 2,703 | 2,337 | 1,641 | 1,631 | 1,058 | 1 | | |
| 2004 | 0.212 | 0.255 | 0.37 | 0.428 | 0.61 | 0.613 | 0.945 | 1 | | |
| Power4+ | 9,710 | 8,081 | 5,559 | 4,807 | 3,374 | 3,355 | 2,176 | 2,057 | 1 | |
| 2002 | 0.103 | 0.124 | 0.18 | 0.208 | 0.296 | 0.298 | 0.459 | 0.486 | 1 | |
| Power4 | 19,788 | 16,469 | 11,330 | 9,797 | 6,877 | 6,838 | 4,435 | 4,192 | 2,038 | 1 |
| 2001 | 0.051 | 0.061 | 0.088 | 0.102 | 0.145 | 0.156 | 0.225 | 0.239 | 0.491 | 1 |

**TABLE 14.** Verification of new server sizing method for U2L migration.

| | Value | Result | Reference |
|---|---|---|---|
| $C_{i+0}$ | 58.82% × 32.0 × 1.2 | 22.58 | Preliminary, maximum CPU utilization × cores × CPU system spare (Value = 20%, Maximum 80% standard) |
| $C_{i+1}$ | 0.815 | 18.41 | tpmC increase rate per core (IBM p595 to x86-based servers) |
| $C_{i+2}$ | 1.56 | 28.71 | Performance change rate of heterogeneous hardware vendor's (Industry-standard : tpmC) |
| $C_{i+3}$ | 4.17 | 119.72 | $users_{prob}$ (Value = 25,288) / $users_{max}$ (Value = 6,065) |
| $C_{i+4}$ | 1.4 | 167.61 | Multi-node revision |
| $C_{i+5}$ | 1.2 | 201.13 | CPU virtualization overload |
| $C_{req}$ | 18,305,244 tpmC | 202 Core(s) | 204 cores (6 cores × 34-Node) ≈ 18,566,448 tpmC |
| Error | - | 1.41% | [(Standard-benchmark) - (Measured value) / (Measured value) × 100] |

**TABLE 15.** Performance standard from hardware vendors.

| CPU Model (tpmC) | Power® 233,754 | SPARE® 202,864 | Itanium® 183,808 | Xeon® 149,571 |
|---|---|---|---|---|
| (IBM) Power® 233,754 | 1.0 | 0.87 | 0.79 | 0.64 |
| (ORACLE) SPARC® 202,864 | 1.15 | 1.0 | 0.91 | 0.74 |
| (HPE) Itanium® 183,808 | 1.27 | 1.10 | 1.0 | 0.81 |
| (INTEL) Xeon® 149,571 | 1.56 | 1.36 | 1.23 | 1.0 |

**TABLE 16.** Hardware cost and response time (ms).

| Classification | Hardware Total Cost[8] | | | Response Time (ms) | |
|---|---|---|---|---|---|
| | WEB | WAS | DBMS | Max. | Avg. |
| Unix-based Census | $ 61.3 | $ 58.4 | $107.3 | 51.94 | 0.24 |
| x86-based Census | $ 41.4 | $ 46.0 | $ 90.0 | 12.31 | 0.21 |

almost linearly, as summarized in Table 13. As it shows similar performance regardless of the workload characteristics of the WEB/WAS/DB server, we compared the IBM® rPerf values with the industry's most trusted performance metrics: tpmC and max-jOPS. When adopting this approach, after calculating the OPS with the OPS conversion ratio of tpmC (=95.12%), the maxjOPS per core was calculated after applying the same maxjOPS (=22,482) based on Intel® Xeon® E5-2660v2 10 cores. However, in this study, we focus on CPU cores due to this being the main factor in determining the x86-based platform for secure U2L migration. We do not consider the difference in application performance because the same JDK version was used, under the premise that the Unix vs. x86-based census survey items are implemented equally. In addition, the OLTP server and storage were excluded from U2L migration, targeting AP servers that perform business logic. Database servers were built on the same Unix-based

[8]Unit: Hundred thousand dollars.

platform (IBM® p780). The reason for this was the step-by-step migration strategy to minimize the risk of change. In the future, we will conduct additional research with Level #3 U2L Migration, including DB servers.

## VI. CONCLUSION

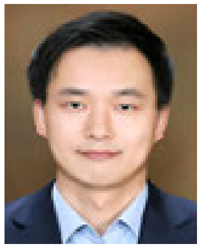
The historical usage precedents for U2L migration do not establish the accuracy of the dimensioning server sizing requirements. This is not in compliance with the optimal correlation for conversion between OPS and max-jOPS owing to the lack of elaboration. In this study, we performed a case study to compare and verify the performance correlation in scenario-based U2L migration by analyzing the collected log data of the census statistical system, using the server sizing simulation method. To this end, tpmC based on actual data per core was calculated and the pre-to-post performance was verified between the existing IBM® Power-based Unix servers before migration and new Intel® CPU-based x86 Linux servers after migration. In addition, the effects of pre-2-node and post-32-node operating environments on the performance of the census statistical system, according to the workload distribution, were verified. After calculating the OPS of x86 servers based on the result of OPS, where tpmC is compared, the conversion ratio of 0.165 was calculated based on the performance standard benchmarks provided by max-jOPS. This demonstrates how U2L migrates AP servers to x86-based max-jOPS, which is relatively simple based on empirical verification; however, this should be validated by the simulation method of server sizing. Finally, based on the analyzed performance measurement data, we proposed a new server sizing model of "Quantitative Server Sizing Model for Performance Satisfaction in Secure U2L Migration" guideline. This study provides a good example of future large-scale U2L migration between heterogeneous CPUs. Moreover, this approach broadens the choice of users who want to migrate to U2L using relatively inexpensive but widely used x86 servers.

## REFERENCES

[1] *Cloud Services—Amazon Web Services (AWS)*. Accessed: Oct. 12, 2021. [Online]. Available: https://aws.amazon.com/
[2] *Cloud Computing Services | Microsoft Azure*. Accessed: Oct. 12, 2021. [Online]. Available: https://azure.microsoft.com/
[3] *Cloud Computing Services | Google Cloud*. Accessed: Oct. 12, 2021. [Online]. Available: https://cloud.google.com/
[4] *TPC-Homepage*. Accessed: Oct. 12, 2021. [Online]. Available: http://www.tpc.org/
[5] *SPECjbb2015 Benchmark User Guide*. Accessed: Oct. 12, 2021. [Online]. Available: https://www.spec.org/jbb2015/docs/userguide.pdf
[6] *SPEC—Standard Performance Evaluation Corporation*. Accessed: Oct. 12, 2021. [Online]. Available: https://www.spec.org/
[7] M. Amiri and L. Mohammad-Khanli, "Survey on prediction models of applications for resources provisioning in cloud," *J. Netw. Comput. Appl.*, vol. 82, pp. 93–113, Mar. 2017.
[8] C. D. C. Pereira, "A functional paradigm for capacity planning of cloud computing workloads," in *Proc. IEEE/ACM 43rd Int. Conf. Softw. Eng., Companion Proc. (ICSE-Companion)*, May 2021, pp. 281–283, doi: 10.1109/ICSE-Companion52605.2021.00128.
[9] Y. Bo, W. Jun, W. Hua, and W. Gang, "Comparison and analysis of X86 server and minicomputer application in power enterprises," *J. Phys., Conf.*, vol. 1187, Apr. 2019, Art. no. 022062.
[10] *TPC-C All (TPC Benchmark Results)*. Accessed: Oct. 12, 2021. [Online]. Available: http://www.tpc.org/tpcc/results/tpcc_results5.asp
[11] P. Somasekaram, "Evaluation and analysis of hardware sizing for a mission critical enterprise application," Linnaeus Univ., Växjö, Sweden, 2013. Accessed: Oct. 12, 2021. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:lnu:diva-300
[12] H. Muller, S. Bosse, and K. Turowski, "On the utility of machine learning for service capacity management of enterprise applications," in *Proc. 15th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Nov. 2019, pp. 274–281, doi: 10.1109/SITIS.2019.00053.
[13] C. L. Park, T. S. Lee, and J. H. Ra, "Comparison between HW sizing calculation and specjbb performance measurement," *J. Inf. Technol. Archit.*, vol. 16, no. 2, pp. 147–154, 2019.
[14] INTEL. (2012). *Migration From Unix/Risc and Main-Frame to Intel-Based Solutions*. [Online]. Available: https://www.intel.com/content/dam/www/public/en/us/documents//white-papers/unix-risc-and-mainframe-migration-paper.pdf/
[15] TTA(Telecommunications Technology Association). (2018). *A Guideline for Hardware Sizing of Information Systems*. [Online]. Available: https://www.tta.or.kr/eng/
[16] *National Information Resources Service*. Accessed: Oct. 12, 2021. [Online]. Available: https://nirs.go.kr/eng
[17] (2010). *TPC Benchmark C Standard Specification Revision 5.11*. [Online]. Available: https://www.tpc.org/TPCDocumentsCurrentVersions/pdf/tpc-cv5.11.0.pdf/
[18] (2018). *TPC Benchmarkh Standard Specification Revision 2.18*. [Online]. Available: https://www.tpc.org/TPCDocumentsCurrentVersions/pdf/tpc-hv2.18.0.pdf
[19] SPECjbb2015. (2021). *Specjbb2015 Results*. [Online]. Available: https://spec.org/cgibin/osgresults?conf=jbb2015;op=dump;format=csvdum/
[20] Statistics Korea. (2021). *Population and Housing E-Census News*. [Online]. Available: https://www.census.go.kr/dat/prs/prsList.d
[21] Statistics KOREA. (2021). *Population and Housing e-Census*. [Online]. Available: https://kostat.go.kr/portal/eng/pressReleases/8/1/index.board/
[22] D. Adamson and L. Lancaster, "Methods and systems for deter-mining hardware sizing for storage array systems," U.S. Patent 10 235 399, Mar. 19, 2019.
[23] T. L. Duc, R. G. Leiva, P. Casari, and P.-O. Östberg, "Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–39, Oct. 2019, doi: 10.1145/3341145.
[24] H. Oi and F. Nakajima, "Performance analysis of large receive offload in a xen virtualized system," in *Proc. Int. Conf. Comput. Eng. Technol.*, vol. 1, Jan. 2009, pp. 475–480.
[25] B. Xia, T. Li, Q. Zhou, Q. Li, and H. Zhang, "An effective classification-based framework for predicting cloud capacity demand in cloud services," *IEEE Trans. Services Comput.*, vol. 14, no. 4, pp. 944–956, Jul. 2021, doi: 10.1109/TSC.2018.2804916.
[26] S. O. Duffuaa and M. M. Al-Shayea, "Systematic and effective maintenance capacity planning," in *Proc. Ind. Syst. Eng. Conf. (ISEC)*, Jul. 2020, pp. 1–4, doi: 10.1109/ISEC49495.2020.9230185.
[27] T. P. Roseline, C. J. M. Tauro, and M. Miranda, "An approach for efficient capacity management in a cloud," in *Proc. IEEE Int. Conf. Current Trends Adv. Comput. (ICCTAC)*, Mar. 2017, pp. 1–6.
[28] K. C. Anupama, R. Nagaraja, and M. Jaiganesh, "A perspective view of resource-based capacity planning in cloud computing," in *Proc. 1st Int. Conf. Adv. Inf. Technol. (ICAIT)*, Jul. 2019, pp. 358–363, doi: 10.1109/ICAIT47043.2019.8987357.
[29] IBM. (2021). *The Relative Performance Metric for Power Systems Servers*. [Online]. Available: https://www-03.ibm.com/systems/Power/hardware/notices/rperf.html/
[30] IBM. (2021). *IBM Power Systems Performance Report*. [Online]. Available: https://www-03.ibm.com/systems/Power/hardware/reports/systemperf.html/
[31] INTEL. (2021). *Software Developer Zone*. [Online]. Available: https://software.intel.com/en-us/articles/intel-xeon-processor-scalable-family-technical-overview/
[32] M. Zakarya, "An extended energy-aware cost recovery approach for virtual machine migration," *IEEE Syst. J.*, vol. 13, no. 2, pp. 1466–1477, Jun. 2019, doi: 10.1109/JSYST.2018.2829890.
[33] S. Izukura, K. Yanoo, H. Sakaki, and M. Kawatsu, "Determining appropriate IT systems design based on system models," in *Proc. IEEE 37th Annu. Comput. Softw. Appl. Conf.*, Jul. 2013, pp. 834–835, doi: 10.1109/COMPSAC.2013.142.

[34] L. Bulej, V. Horký, and P. Tuma, "Do we teach useful statistics for performance evaluation?" in *Proc. 8th ACM/SPEC Int. Conf. Perform. Eng. Companion*, Apr. 2017, pp. 185–189, doi: 10.1145/3053600.3053638.
[35] K. B. Sandeep, M. Raghavendra, and K. J. Shetty, "Performance analyzer: An approach for performance analysis of enterprise servers," in *Proc. IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2016, pp. 480–484, doi: 10.1109/RTEICT.2016.7807867.
[36] D. Gmach, J. Rolia, and L. Cherkasova, "Comparing efficiency and costs of cloud computing models," in *Proc. IEEE Netw. Oper. Manage. Symp.*, Apr. 2012, pp. 647–650.

**HUN JOE** received the B.S. degree in computer engineering from Kensington London College, in 2000, and the M.S. degree in computer science from King's College London, U.K., in 2002. After graduation, he has been working as a Senior Researcher with the Cloud Computing Department, Samsung SDS and Korea Advanced Institute of Science and Technology (KAIST) for Ph.D. His research interests include cloud computing, information security, and software-defined networking from the perspective of performance/optimization. He is also actively involved in the research related to the modeling and analysis of IT architecture at the Korea Statistical Information Institute.



**SUNGJIN KIM** received the B.S. degree in computer science from The Ohio State University, Columbus, USA, the M.S. degree in computer science from Sogang University, Seoul, Republic of Korea, and the Ph.D. degree from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2019. He is currently an Assistant Professor with Cheju Halla University. His current research interests include network security, machine learning, big data analytics, social network analysis, web security, and malware detection and analysis.

**BRENT BYUNGHOON KANG** (Member, IEEE) received the B.S. degree from Seoul National University, the M.S. degree from the University of Maryland at College Park, and the Ph.D. degree in computer science from the University of California at Berkeley. He is currently an Associate Professor with the Graduate School of Information Security, Korea Advanced Institute of Science and Technology (KAIST). Before KAIST, he was with George Mason University as an Associate Professor. He has been working on systems security areas, including botnet defense, OS kernel integrity monitors, trusted execution environment, and hardware-assisted security. He is also a member of USENIX and ACM.