White Paper

Data Center
Cloud Computing

Live Migration Downtime and Cost Prediction in Cloud Computing Environments

Authors

Carlos Llano

School of Computation,
Information and Technology
Technical University of Munich
Munich, Germany
carlos.llano@tum.de

Annatam Dey

Data Center Group, Intel,
Bangalore, India
annatam.dey@intel.com

Mohamed Elsaid

Data Center Group, Intel
Munich, Germany
mohamed.elsaid@intel.com

Abstract

Live migration allows the migration of virtual machines from one physical machine to another to perform critical operations such as load balancing, power saving, fault tolerance, and maintenance. Regardless of the method used to perform live migration (i.e., pre-copy, post-copy, hybrid), this process incurs some costs that need to be considered to evaluate its performance. These costs include downtime, migration time, network overhead, CPU utilization overhead, and power consumption overhead.

This paper studies the costs of live migration and their relationships by doing live migrations of virtual machines in a virtual data center created with VMware vSphere. Finally, given the importance of minimal downtime to guarantee service level agreements (SLAs), several regression models are proposed to predict the live migration downtime. The results showed that an ElasticNet regression model can satisfactorily predict the downtime with an accuracy of 92.4 percent.

Index Terms – data center, virtualization, hypervisor, virtual machine, live migration, downtime, VMware, vMotion

Introduction

Live migration refers to the process of moving a virtual machine (VM) from one host (source host) to another (target host) in such a way that the running applications have almost no impact on their availability, thus minimizing downtime and interruption.[1]

There are multiple reasons to perform this procedure, all of which have special relevancy in cloud environments: 1) load balancing, to guarantee a balanced usage of servers and avoid bottlenecks, 2) power saving, to minimize the number of active servers, 3) fault tolerance, such as a copy of a VM can be used immediately in another host in case of failure, and 4) maintenance, to perform necessary hardware or software upgrades while ensuring uninterrupted service availability.

Live Migration relies heavily on the concept of virtualization, which is a technique that aims to allow resource sharing (i.e., CPU, network, storage) between different processes while guaranteeing isolation, in other words, the shared resources are considered as own by the underlying process.

A Virtual Machine (VM) or virtual computer is the result of applying virtualization to the host machine's compute resources, such as the CPU, cache, memory and all other compute hardware can be shared across several VMs withoutinterfering with one another.[2] As a result, multiple and different operating systems (Guest OS) can access the same hardware and run their own applications.

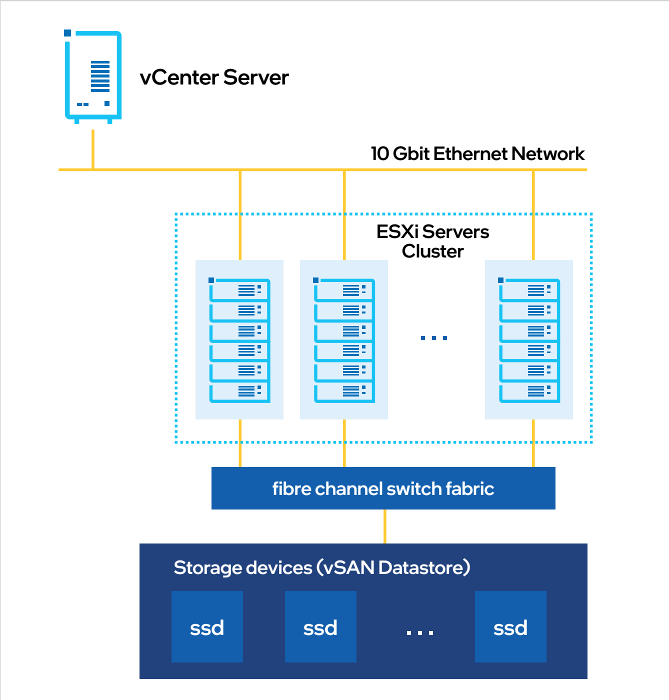

Figure 1. VMware vSphere Components.

The sharing of the host hardware resources between multiple compute environments (i.e., VMs) is managed by a software component called hypervisor. The hypervisor is responsible for creating VMs and it manages their interaction with the host machine to access hardware resources.

In the context of live migration, the hypervisor from the source host is the one that coordinates the entire process. Currently, the most used hypervisors that include live migration as a feature are VMware ESXi, Microsoft Hyper-V, Xen, and KVM.[3]

Live Migration Process

The live migration process is coordinated by the source host hypervisor and it starts with an initialization phase and a reservation phase. In initialization, the VM needed to be migrated is selected as well as the target machine. In reservation, the source machine sends a request to the target machine for the required resources to run the VM. The target machine acknowledges the request and it reserves the required resources. After reservation, there are three main methods to continue the process: pre-copy, post-copy, and hybrid-copy.[4]

In pre-copy live migration, a complete transfer of the RAM memory pages is done at the beginning without shutting down the VM. As the VM continues running in the source host, the modified memory pages (also known as dirty pages) are migrated in an iterative copy phase until a stopping condition is met. The Iterative copy is implemented through several rounds where dirty pages generated in the previous rounds are sent to the target host to ensure memory consistency. This iterative process continues until a stopping condition is met, which can be: 1) if the memory dirtying rate exceeds the memory transmission rate; 2) if the remaining dirty memory becomes smaller than a predefined threshold value, or 3) if the number of iterations exceeds a given value.[5] Once the condition is met, the VM is stopped at the source host, then the last (dirty) memory copy is performed, and the CPU state is transferred to the target host (stop-and-copy phase).

After stopping the source VM and doing a final copy, the pre-copy live migration continues with a validation in the target host, known as Commitment, to make sure that the target host has received a complete copy of the VM, then the target host sends a message to the source host indicating that the migration has been successful. Finally, an Activation phase is performed, where the VM in the target host is activated and the VM in the source host is discarded.

Post-copy live migration follows a different approach when migrating memory. Instead of migrating the entire memory before activating the VM in the target host (as in pre-copy live migration), post-copy live migration transfers the memory after activation. The process starts by stopping the VM in the source host and transferring only the required data to boot the VM in the target host. After the target host activates the VM, the source host sends the memory data in one iteration. This type of live migration reduces the downtime and makes the total time spent in live migration deterministic as it does not deal with continuously modified memory pages (dirty pages). However, taking over the VM without guaranteeing a complete memory transfer can incur failures and data loss in case of network disruption between the source host and the target host.

Hybrid-copy live migration combines the previous approaches by doing a pre-copy phase and post-copy phase. As with pre-copy live migration, hybrid-copy performs iterative copies of the memory data until a fixed number of runs is reached. Then, the VM in the source host is stopped, the CPU state is transferred, and as opposed to pre-copy live migration, the remaining memory dirty pages are migrated after the VM in the target host is activated, following a post-copy live migration approach.

Live Migration Costs

Regardless of the method implemented to perform live migration, there are associated costs that need to be considered such as downtime, migration time, network overhead, CPU utilization overhead, and power consumption overhead.[3]

Downtime is the most critical cost due to its direct impact on service availability. In the context of business applications where SLAs are of much importance, downtime must be negligible (milliseconds) so that no interruption is noted. In live migration, downtime is the time between the stopping of the VM in the source host and its activation in the target host.

Migration time is the time spent from the live migration initialization until the final activation of the VM in the target host. Generally, migration time can take over a couple of seconds to minutes depending on the VM memory size, the memory dirty pages rate (i.e., the speed at which memory pages are being overwritten), and the network throughput.

Network overhead refers to the consumed bandwidth during the migration process when transferring memory copies and CPU state from the source host to the target host.

CPU utilization overhead is due to the increase in operations that need to be done by the hypervisor to perform iterative copies and transfer required data to the target host.

Power consumption overhead is associated with the increase of resource utilization both in the source host and target host during the migration process, specifically in terms of CPU utilization and network bandwidth.[6]

Experimentation Setup

VMware vSphere Data Center Lab

To study live migration in a cloud environment setting, a virtual data center was used to migrate VMs. The data center was created using VMware vSphere, a VMware product that allows the creation and management of virtualized infrastructure, using a bare-metal hypervisor (ESXi), a cluster manager (vCenter), a shared storage network (vSAN - Virtual Storage Area Network), and dedicated software for live migration (vMotion). The data center is comprised of 3 hosts with the following technical specifications: 80 CPU cores (2.3 GHz) and 1 TB of RAM. The storage is provided by a vSAN datastore with around 6 TB of capacity. The network bandwidth in the data center has a maximum of 10 Gbps. The hypervisor installed in each of the hosts is VMware ESXi 7.0.3 version.

vMotion Live Migration

As part of the broad suite of products offered by VMware vSphere, vMotion vMotion is the product tailored to VM live migration. It follows a pre-copy live migration strategy where CPU state and memory pages are transferred from the source host to the target host over a high-speed network. The entire process is started by the vCenter Server with a compatibility check to determine whether or not the VM can be migrated based on hardware requirements and ESXi versions from the source host and the target host. After that, a migration specification is created containing details about the VM to be migrated, the ESXi versions involved, and the vMotion network configuration. The vCenter Server communicates the migration specification to the source and target host ESXi processes, they block any configuration change to the VM to be migrated and start a socket connection between the hosts for transferring data (i.e., CPU state and memory pages). Finally, VMware vMotion guarantees a migration time that can vary from seconds to minutes depending on the VM workload, and a downtime in the order of milliseconds.

Memory Stress

Live migration overall performance is highly influenced by the workload of the VM to be migrated. The workload refers to the amount of usage of CPU, RAM, and network. To simulate live migration in different workloads, a stress tool was used to stress out the VM resources.

Considering that most of the data transfer in live migration relates to RAM memory, a decision was made to consider only memory stress load to evaluate live migration. In that regard, the stress tool selected was stress-ng, which is a Linux tool that allows stress testing through the usage of different types of stressors. An important feature of stress-ng is that it enables generating stress load at a certain percentage (e.g., 25%, 50%, 90%), which reduces the risk of hardware damage and it facilitates the study of migration performance under multiple scenarios.[7] In particular, for this study only 90 percent of the available RAM is stressed.

For reference, other stress tools that have been used in different live migration studies are: kernbench,[8] burn,[9] linpack,[10] and iperf.[11]

Downtime Measurement

Downtime is the time during which the VM is not responsive during live migration, and thus it does not serve any requests for end users or external applications. For its measurement, high-speed pings were used, where continuous pings are sent to the VM during migration from another VM at a millisecond-level interval. The downtime is calculated by multiplying the amount of lost pings times the ping interval.

Virtual Machine Configurations

Multiple VM configurations were used with a set of common specifications. The differentiating factor was the configuration in terms of RAM memory (i.e., 4 GB, 8 GB, 16 GB, 24 GB, 32 GB), to simulate live migration for common RAM settings in cloud computing environments. Two VMs were created, where one of them was migrated in several iterations with different configurations and the other one was used as a helper instance to calculate the live migration downtime through the use of high-speed ping requests to the VM that was migrated. A total of 20 live migrations were performed per configuration.

Exploratory Data Analysis

For each one of the migrations performed in the testing environment, the following live migration costs were collected: migration time, downtime, active memory size, network transmit rate, and peak power change.

Migration time defines the overall time to transfer the VM from the source host to the target host. It is measured in seconds.

Active memory size indicates the memory used by the VM at the moment when it was migrated. It is measured in Kilobytes.

Downtime defines the time between the stopping of the VM to perform the last data transfer (i.e., CPU state and last dirty pages) and its activation in the target host. It is measured in milliseconds (ms).

Network transmit rate refers to the speed at which the source host transmits data to the target host. It is measured in kilobits per second (kbps).

Peak power change refers to the peak power consumption of the source host during the live migration process. It is the result of the increase in resource usage (CPU operations and data transfer) and it is measured in Watts.

To determine how to model the downtime based on these live migration costs, multiple live migrations were performed using VMWare vMotion, and the results were summarized using a correlation matrix. Correlation is a statistical measure that indicates how strong the linear relationship between two variables is, along with the direction of this relationship (i.e., positive, neutral or negative).[12] Correlation between two variables can adopt values between -1 and 1, the nearer the value is to the thresholds the stronger the relationship.

The correlation matrix of the live migration cost parameters evidences a linear relationship with each one of the cost parameters but more strongly with active memory size, migration time and network transmit rate.

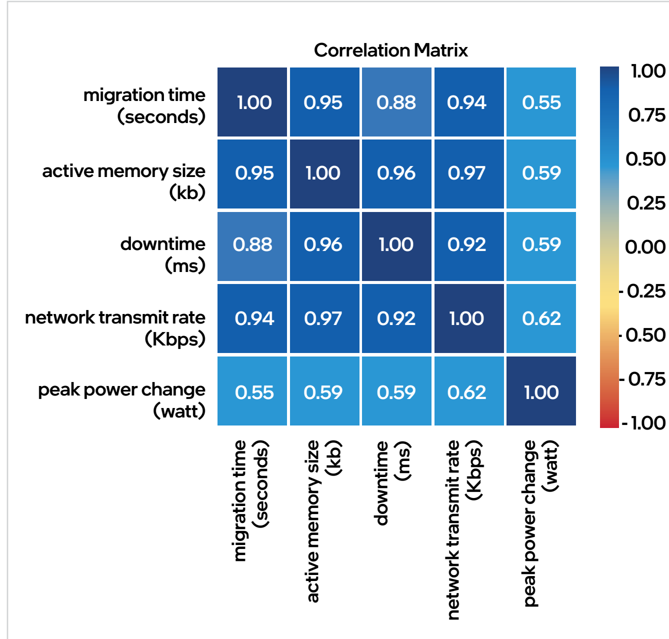

Figure 2. Live Migration Costs Correlation Matrix.

As it was mentioned, during live migration an iterative process is performed to copy memory from the source host to the target host until a stopping condition is met, which can be that the remaining dirty memory becomes smaller than a predefined threshold or the number of iterations exceeds a given value.[5] The increase in active memory size implies that more memory needs to be transferred during live migration, either because it is closer to the threshold or the limit of pre-copy iterations has been reached, which ultimately can make the last copy of dirty pages more time-consuming, thus, increasing the live migration downtime.

The linear relationship between downtime and both migration time and network transmit rate can be seen as a consequence of the underlying increase in active memory size. As the active memory increases, the iterative memory copies require more iterations, which impacts the migration time. The same can be said about network transmit rate, the increase in active memory size provokes that the source host migration module leverages the available bandwidth to transfer as much data as possible until a bandwidth limit is reached or the network is congested.

The linear relationship between downtime and power consumption is not as clear as with the other cost parameters, mainly because the source host peak power consumption in live migration is produced during the iterative memory copies and before the stop-and-copy phase where downtime happens.

Modeling

Using the findings during data analysis of the live migration costs, a modeling strategy and a set of models were studied to obtain a prediction of the downtime of a VM during live migration.

Modeling Strategy

During the exploration of the relationships between the live migration cost parameters, it was found that the behavior of most of them depends highly on the current RAM being used, namely the active memory size of the VM to be migrated. This parameter has a high correlation (i.e., 0.95) with the migration time indicating that it can be used as a good predictor to determine the migration time. On the other hand, the network transmit rate, which refers to the speed at which the data is transferred from the source host, has a high correlation with both active memory size (i.e., 0.97) and migration time (i.e., 0.94), thus making them good candidates to determine its value.

Combining a measurable value like the active memory size, and two predicted values such as the migration time and the network transmit rate, a model can be established to predict the expected downtime of the VM live migration.

The usage of these three predictors for the downtime is justified by their high correlation (around 0.9), as opposed to the peak power consumption, whose relatively low correlation denotes a small influence on the downtime and the rest of the live migration costs parameters.

In conclusion, the proposed modeling strategy makes use of three distinct models to predict the live migration downtime, in such a way that it can be calculated before its initialization, while at the same time using live migration cost parameters that can be easily monitored and measured.

Regression Models

The study of the relationships between downtime and live migration cost parameters showed the presence of a linear relationship, which can be modeled through regression. A regression model establishes a linear equation to model the relationship between the variable to predict (i.e., dependent variable, *y*) and one or more predictors (i.e., independent variables, *xj*).[12]

$$y = \alpha + \sum_{j=1}^{p} x_j \beta_j$$

Depending on the method used to find out the coefficients *βj* of the linear equation, different types of regression models can be distinguished.

Ordinary Least Squares (OLS) regression is a technique that seeks to find the coefficients of a regression equation by minimizing a loss function equal to the sum of the squares of the residuals (i.e., the difference between the actual value and the predicted value).

$$\sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2$$

There are some cases where the linear relationship is much more clear with some of the explanatory variables, that OLS regression makes the other variables have coefficients near zero, thus, they effectively become insignificant for the rediction. For instance, when using the OLS regression to model the downtime based on the active memory size, the migration time and the network transmit rate, only the active memory size has a coefficient different from zero (i.e., 11.645). The large number of insignificant variables present in the regression is an indication that the model is overfitting to particular variables (i.e., active memory size) that have a very strong linear relationship with the variable to predict (i.e., downtime). To prevent this effect, regularization techniques must be used.

Lasso regression utilizes L1 regularization, which adds the absolute magnitude of the regression model coefficients (*βj*) to the ordinary least squares (OLS) loss function, multiplied by a scaling term *λ*:

$$\sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

In other words, extending OLS linear regression to use L1 regularization requires adding an extra term $\lambda \sum_{j=1}^{p} |\beta_j|$ to prevent overfitting or penalize increased model complexity. The value of *λ* sets the strength of the regularization, larger values of *λ* will increasingly affect the *βj* coefficients.

L1 regularization often suffers with predictions when the data exhibits collinearity, which means that some predictors can be expressed to a good extent as linear combinations of other predictors. This is the case with network transmit rate, which has been shown to have linear relationships with migration time and active memory size. To address this issue, L2 regularization can be used. A linear regression model that utilizes L2 regularization is called Ridge regression. L2 regularization adds the squared magnitude of the regression model coefficients (*βj*) to the OLS loss function:

$$\sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

ElasticNet regression blends both L1 and L2 regularization to include both the absolute value and squared magnitude of the coefficients in the loss function:

$$\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \alpha\rho \sum_{j=1}^{p} |\beta_j| + \alpha(1-\rho) \sum_{j=1}^{p} \beta_j^2$$

In the loss function, *α* is the regularization rate, and *ρ* controls the balance between L1 and L2 regularization in the loss function. Namely, *ρ* = 1 results in a pure L1 regularization, *ρ* = 0 results in a pure L2 regularization, while 0 < *ρ* < 1 results in a blended elastic net regularization.

Model	R-squared	MAE	RMSE	MAPE
OLS	0.917	3.015	3.554	0.078
Lasso	0.857	3.548	4.671	0.096
Ridge	0.900	3.281	3.901	0.084
ElasticNet	0.925	2.918	3.384	0.076

Table 1. Regression Models Performance.

Results

A set of regression metrics was used to summarize the performance of all the regression models when predicting the downtime in the live migration of VMs following the modeling strategy.

It can be seen that the standard regression or OLS regression achieves good performance, but as it was mentioned, it does it at the expense of overfitting to active memory size, making all the other live migration cost parameters (i.e., migration time and network transmit rate) insignificant for the prediction.

Lasso regression uses L1 regularization to prevent over-fitting to active memory size and it obtains an acceptable performance as demonstrated by the R-squared metric. It is lower than OLS regression but it makes use of all the live migration cost parameters to achieve a prediction of the downtime. However, some of the predictor variables are highly correlated, which affects the performance of Lasso regres-sion. Ridge regression addresses this issue by performing L2 regularization and it obtains a very good performance with a R-squared of 0.9. To make use of all the benefits of Lasso and Ridge regression, ElasticNet regression was proposed and it represents the best model with an R-squared of 0.925. The magnitudes of the MAE (2.918) and the RMSE (3.384) indicate small prediction errors. In that regard, with the interpretation of the MAPE error (0.076), a notion of accuracy can be given to the overall prediction performance. Thus, the ElasticNet regression model predictions are on average off by 7.6 percent, in other words, it is on average 92.4 percent accurate.

Conclusion

Live migration of virtual machines is an essential feature in the operations of data centers and cloud computing environments. However, as with any IT operation, its applicability has resource costs related to migration time, downtime, active memory size, network transmit rate, and power consumption.

This study explored the relationships between these costs and tried to find a method that can predict not only the downtime but also other live migration costs using as a basis the available virtual machine state (i.e., active memory size), which can be extremely useful for IT admins to decide whether or not to perform the live migration of a certain virtual machine, and if so, have a clear overview of its impact over the IT infrastructure (i.e., migration time, network transmit rate, downtime). With this goal in mind, a testing environment was implemented in a virtual data center created with VMware vSphere and multiple live migration experiments were performed using vSphere vMotion. The study of the live migration costs evidenced high correlation and linear relationships between all costs, except for power consumption. Therefore, a set of regression models was proposed to use migration time, active memory size and network transmit rate to predict downtime. These models included: OLS regression, Lasso regression, Ridge regression, and ElasticNet regression.

The evaluation of the regression models showed that ElasticNet regression is the best model to predict the downtime of a virtual machine during live migration with an average accuracy of 92.4 percent.

Future Work

The prediction power of the model can be further improved including other variables that influence the live migration process such as the memory dirty page rate or the number of iterative copies iterations. Another extension to this study could compare the prediction when using different hypervisors such as Microsoft Hyper-V, Citrix Xen, or KVM. In terms of VM configurations, different settings can be explored in terms of RAM or different types of workloads such as network-intensive workloads. Finally, it is worth mentioning that live migration is highly affected by the data center configuration and, in this regard, it would be valuable to study live migration with different network bandwidths or to compare results when performing live migration between hosts in the same network and hosts in different data centers.

Acknowledgment

We would like to thank Intel Germany for giving us access to the computational resources needed to complete this study.

[1] M. A. Altahat, A. Agarwal, N. Goel, and J. Kozlowski, "Dynamic hybrid-copy live virtual machine migration: Analysis and comparison," Procedia Computer Science, vol. 171, pp. 1459–1468, 2020, third International Conference on Computing and Network Communications (CoCoNet'19). https://www.sciencedirect.com/science/article/pii/S1877050920311352.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," SIGOPS Oper. Syst. Rev., vol. 37, no. 5, p. 164–177, oct 2003. https://doi.org/10.1145/1165389.945462.

[3] M. Elsaid, H. Abbas, and C. Meinel, "Virtual machines pre-copy live migration cost modeling and prediction: a survey," Distributed and Parallel Databases, vol. 40, pp. 1–34, 12 2021.

[4] A. Shribman and B. Hudzia, "Pre-copy and post-copy vm live migration for memory intensive applications," 08 2012, pp. 539–547.

[5] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in Proceedings of the 20th International Symposium on High Performance Distributed Computing, ser. HPDC '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 171–182. https://doi.org/10.1145/1996130.1996154.

[6] Q. Huang, F. Gao, R. Wang, and Z. Qi, "Power consumption of virtual machine live migration in clouds," in 2011 Third International Conference on Communications and Mobile Computing, 2011, pp. 122–125.

[7] P. S. Indukuri, "Performance comparison of linux containers (lxc) and openvz during live migration," Ph.D. dissertation, 10 2016.

[8] D. Fernando, P. Yang, and H. Lu, "Sdn-based order-aware live migration of virtual machines," in IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, 2020, pp. 1818–1827.

[9] F. Salfner, P. Tr¨oger, and A. Polze, "Downtime analysis of virtual machine live migration," 01 2011, pp. 100–105.

[10] M. Elsaid, H. Abbas, and C. Meinel, "Machine learning approach for live migration cost prediction in vmware environments," 06 2019.

[11] R. Torre, R.-S. Schmoll, F. Kemser, H. Salah, I. Tsokalo, and F. H. Fitzek, "Benchmarking live migration performance under stressed con-ditions," in 2021 IEEE 18th Annual Consumer Communications Net-working Conference (CCNC), 2021, pp. 1–2.

[12] V. Bewick, L. Cheek, and J. Ball, "Statistics review 7: Correlation and regression," Critical care (London, England), vol. 7, pp. 451–9, 01 2004.