International Journal of Distributed Systems and Technologies
Volume 9 • Issue 1 • January-March 2018

Analysis of Frequently Failing Tasks and Rescheduling Strategy in the Cloud System

Hongyan Tang, School of Software and Microelectronics, Peking University, Beijing, China
Ying Li, National Engineering Center of Software Engineering, Peking University, Beijing, China
Tong Jia, School of Software and Microelectronics, Peking University, Beijing, China
Xiaoyong Yuan, Department of Computer and Information Science and Engineering, University of Florida, Florida, USA
Zhonghai Wu, National Engineering Center of Software Engineering, Peking University, Beijing, China

ABSTRACT

To better understand task failures in cloud computing systems, the authors analyze failure frequency of tasks based on Google cluster dataset, and find some frequently failing tasks that suffer from long-term failures and repeated rescheduling, which are called killer tasks as they can be a big concern of cloud systems. Hence there is a need to analyze killer tasks thoroughly and recognize them precisely. In this article, the authors first investigate resource usage pattern of killer tasks and analyze rescheduling strategies of killer tasks in Google cluster to find that repeated rescheduling causes large amount of resource wasting. Based on the above observations, they then propose an online killer task recognition service to recognize killer tasks at the very early stage of their occurrence so as to avoid unnecessary resource wasting. The experiment results show that the proposed service performs a 93.6% accuracy in recognizing killer tasks with an 87% timing advance and 86.6% resource saving for the cloud system averagely.

KEYWORDS

Failure Frequency, Killer Tasks, Rescheduling, Resource Usage Pattern, Time Series

INTRODUCTION

With its on-demand accessing and pay-as-you-go manner, cloud computing has been a new paradigm for providing computing resources. Nowadays, more and more organizations and companies in different areas decide to run their applications on cloud system gradually. Higher dependability and availability of cloud platform is essential to ensure quality of services of applications. However, due to its ever-increasing heterogeneous, distribution and large-scale, failures are more norm than expectation in cloud computing system, which leads to a high demand on capability of fault tolerance and fast recovery. Therefore, a deep understanding of the failure pattern can lead not only to effective solutions, but also to a better fulfillment of the objective of high dependability.

Among different types of failures in cloud computing system, task failure is the most basic and common one because task is the minimal scheduling unit running on a single machine. To deal with task failures, one of the most effective method is rescheduling (Soualhia, Khomh & Tahar, 2015).

DOI: 10.4018/IJDST.2018010102

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

However, there are some tasks behaving like a "crash-loops" which deterministically fail shortly after execution while rescheduling repeatedly (Reiss, Tumanov, Ganger, Katz & Kozuch, 2012b). These tasks may do great harm to the cloud computing system, because they continually compete for resources and then waste resources due to failure. Meanwhile, they significantly increase the scheduling workload. As a result, in this paper, we call this type of tasks "killer task."

To better understand killer task, we develop a study on Google cluster workload traces[1], which contain the workload measurements of 25M tasks of 650K jobs on more than 12,000 nodes during a one month period. In Google cluster, each task represents execution and resource allocation unit of job, and is scheduled onto a single machine. We analyze failure frequency of tasks and find that some tasks suffer from continual failures and repeated rescheduling, for example, we find 1,151 tasks experiencing more than 500 times of failures and rescheduling during their lifetime. After analyzing the characteristics and rescheduling strategies of these tasks, we propose an online killer task recognition service to help cloud systems recognize killer tasks automatically and proactively, to save computing resources and promote the stability of the cloud system.

Prior studies on failures of cloud computing systems focus on characterization of job failures (Chen, Lu & Pattabiraman, 2014a), server failures (Garraghan, Townend & Xu, 2014; Birke, Giurgiu, Chen, Wiesmann & Engbersen, 2014) and failure prediction (Chen, Lu & Pattabiraman, 2014b; Rosa, Chen & Binder, 2015a). To the best of our knowledge, we are the first to recognize killer tasks and perform online recognition at their early stage. We make the following contributions in this paper:

- Exploring killer tasks by analyzing failure frequency of tasks in Google cluster and revealing their characteristics;
- Investigating resource usage pattern of killer tasks and non-killer tasks and identifying differences between them;
- Disclosing rescheduling strategies of killer tasks performed in Google cluster and evaluating their positive and negative impacts quantitatively;
- Proposing framework of online killer task recognition service that make use of resource usage time series to recognize killer tasks at the early stage of their occurrence;
- Implementing the prototype of proposed service and verifying it with experiments. The results show that it can precisely recognize killer tasks with 93.6% of accuracy and provide cloud system an average of 633 minutes to take proactive actions, with 86.6% of resource saving.

The rest of this paper is organized as follows. Section 2 presents the related work. In section 3, we analyze failure frequency of tasks to find killer tasks, followed by investigation and comparison of resource usage pattern of killer tasks and non-killer tasks. The time series based online recognition service of killer tasks is proposed in Section 4. Section 5 presents the experimental results and benefits from killer task recognition, followed by the discussion in Section 6. Section 7 concludes our work.

RELATED WORK

Google Trace Analysis

There have been many analysis efforts focusing on task events and resource usage in the Google trace data. Reiss et al. (2012a) reveal the heterogeneity and dynamicity of machines and workloads in Google cluster. They also find that many longer-running jobs have relatively better resource usage predictability. Mishra et al. (2010) characterize resource usage of tasks and propose a *k-means* based method of task classification. They find that most of the CPU and memory are consumed by a minority of longer running tasks. Di et al. (2013) characterize Google applications from the perspective of task events and resource usage. They find that resource utilized by per application follows a typical *Pareto* principle. They then conduct application clustering via an optimized *k-means* algorithm and find

21 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/analysis-of-frequently-failing-tasks-and-rescheduling-strategy-in-the-cloud-system/196265

Related Content

Model-Based Application Deployment on Cloud Computing
Aouat Asmaa, Deba El Abbassia, Benyamina Abou EL Hassanand Benhamamouch Djilali (2019). *International Journal of Distributed Systems and Technologies (pp. 110-127).*
www.irma-international.org/article/model-based-application-deployment-on-cloud-computing/226964

Serendipity Reloaded: Fair Loading in Event-Based Messaging
Daniel Cuttingand Aaron Quigley (2010). *Principles and Applications of Distributed Event-Based Systems (pp. 90-118).*
www.irma-international.org/chapter/serendipity-reloaded-fair-loading-event/44397

A Failure Detection System for Large Scale Distributed Systems
Andrei Lavinia, Ciprian Dobre, Florin Popand Valentin Cristea (2011). *International Journal of Distributed Systems and Technologies (pp. 64-87).*
www.irma-international.org/article/failure-detection-system-large-scale/55422

A Deep Q-Network Eith Experience Optimization (DQN-EO) for Atari's Space Invaders and Its Performance Evaluation
Elis Kulla (2022). *International Journal of Distributed Systems and Technologies (pp. 1-13).*
www.irma-international.org/article/a-deep-q-network-eith-experience-optimization-dqn-eo-for-ataris-space-invaders-and-its-performance-evaluation/296249

Non-Contact Abnormal Physiological Status Detection During Sport and Training
Jinlin Yang (2021). *International Journal of Distributed Systems and Technologies (pp. 16-26).*
www.irma-international.org/article/non-contact-abnormal-physiological-status-detection-during-sport-and-training/287860