PROJECT NO. 50

NKR: ON TOP SCHEDULER FOR APACHE MESOS

MS.PASINEE SANTIVORRANANT

MR.SUPAPAT SRI-ON

MS.PARATTHA WEERAPONG

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)
FACULTY OF ENGINEERING
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI
2020

Project No. 50

NKR: On top scheduler for Apache Mesos

Ms.Pasinee Santivorranant

Mr.Supapat Sri-on

Ms.Parattha Weerapong

A Project Submitted in Partial Fulfillment
of the Requirements for
the Degree of Bachelor of Engineering (Computer Engineering)
Faculty of Engineering
King Mongkut's University of Technology Thonburi
2020

Project Committee

.... (Asst Prof.Rajchawit Sarochawikasit)	Project Advisor
.... (Asst Prof. Dr. Khajonpong Akkarajitsakul)	Committee Member
.... (Asst Prof. Dr. Phond Phunchongharn)	Committee Member
.... (Asst Prof. Sanan Srakaew)	Committee Member

Project Title	Project No. 50
	NKR: On top scheduler for Apache Mesos
Credits	3
Member(s)	Ms.Pasinee Santivorranant
	Mr.Supapat Sri-on
	Ms.Parattha Weerapong
Project Advisor	Asst Prof.Rajchawit Sarochawikasit
Program	Bachelor of Engineering
Field of Study	Computer Engineering
Department	Computer Engineering
Faculty	Engineering
Academic Year	2020

Abstract

In a multihop ad hoc network, the interference among nodes is reduced to maximize the throughput by using a smallest transmission range that still preserve the network connectivity. However, most existing works on transmission range control focus on the connectivity but lack of results on the throughput performance. This paper analyzes the per-node saturated throughput of an IEEE 802.11b multihop ad hoc network with a uniform transmission range. Compared to simulation, our model can accurately predict the per-node throughput. The results show that the maximum achievable per-node throughput can be as low as 11% of the channel capacity in a normal set of α operating parameters independent of node density. However, if the network connectivity is considered, the obtainable throughput will reduce by as many as 43% of the maximum throughput.

Keywords: Multihop ad hoc networks / Topology control / Single-Hop Throughput

หัวข้อปริญญานิพนธ์	หัวข้อปริญญานิพนธ์บรรทัดแรก หัวข้อปริญญานิพนธ์บรรทัดสอง
หน่วยกิต	3
ผู้เขียน	นางสาวภาสินี สันติวรนันท์ นายศุภพัฒน์ ศรีอ่อน นางสาวปรัษฐา วีระพงษ์
อาจารย์ที่ปรึกษา	ผศ.ดร.ราชวิชช์ สโรชวิกสิต
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมคอมพิวเตอร์
คณะ	วิศวกรรมศาสตร์
ปีการศึกษา	2563

บทคัดย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์ เพื่อศึกษาความพึงพอใจในการให้บริการงานทั่วไปของสานักวิชา พื้นฐานและภาษา เพื่อเปรียบเทียบระดับความพึงพอใจต่อการให้บริการงาน ทั่วไปของสานักวิชาพื้นฐานและภาษา ของนักศึกษาที่มาใช้บริการสานักวิชาพื้นฐานและภาษา สถาบัน เทคโนโลยีไทย-ญี่ปุ่น จาแนกตามเพศ คณะ และชั้นปีที่ศึกษา เพื่อศึกษาปัญหาและข้อเสนอแนะของ นักศึกษามาเป็นแนวทางในการพัฒนาและปรับปรุงการให้บริการของสานักวิชาพื้นฐานและภาษา

คำสำคัญ: การชุบเคลือบด้วยไฟฟ้า / การชุบเคลือบผิวเหล็ก / เคลือบผิวรังสี

ACKNOWLEDGMENTS

Acknowledge your advisors and thanks your friends here..

CONTENTS

PAGE

LIST OF TABLES

TABLE **PAGE**

LIST OF FIGURES

LIST OF SYMBOLS

SYMBOL		UNIT
α	Test variable	m^2
λ	Interarival rate	jobs/ second
μ	Service rate	jobs/ second

LIST OF TECHNICAL VOCABULARY AND ABBREVATIONS

ABC	=	Adaptive Bandwidth Control
MANET	=	Mobile Ad Hoc Network

CHAPTER 1 INTRODUCTION

1.1 Problem Statement and Approach

Nowadays, several different types of applications, which are short or long-lived jobs, container orchestration, or MPI jobs, are executed in clouds or large computer clusters. Multiple users can demand difference resources to execute their tasks. Apache Mesos is a Middleware for the data center by introducing an abstraction layer that provides an entire data centers as a single large server. Instead of focusing on one application that running on a specific server. Mesos resource-isolation allows multi-tenant — the ability to run multiple applications on a single machine. Default sharing for multiple resources in this multi-tenant environment is defined by the Dominant Resource Fairness (DRF). Mesos receives the resources based on their current usage, which are responsible for scheduling their tasks within the allocation. In multiple schedulers can cause the fairness-imbalance in a multi-user environment, liked a greedy scheduler. It consumes more than its share of resources. Running multiple small tasks is better than launching large ones in terms of time spent waiting for enough resources.

Therefore, this project aims to improve the fairness of the scheduler by reducing the unfair waiting time due to higher resource demand in a pending task list and use log data to improve the whole cluster.

1.2 Objectives

- To study about job scheduling in Apache Mesos
- To study how to develop an algorithm to improve performance of scheduler in large-scale clustered environments.
- · To evaluate result and compare with Apache Mesos scheduler by using difference job types in the list (short job, long job, MPI)

1.3 Scope

- This project focuses on the reduction of job failed.
- Design and develop an add-on architecture on top of the Apache Mesos scheduler, to track and distribute the incoming tasks.
- What are the limitations of existing approaches?

1.4 Project Schedule

CHAPTER 2 BACKGROUND THEORY AND RELATED WORK

Explain theory, algorithms, protocols, or existing research works and tools related to your work.

2.1 Recommender Systems

Table 2.1 test table method1

Center	Center	left aligned	Right	Right aligned
Center	Center	left aligned	Right	Right aligned
Center	Center	left aligned	Right	Right aligned
Center	Center	left aligned	Right	Right aligned
Center	Center	left aligned	Right	Right aligned

2.2 Text Processing Algorithms

2.2.1 Algorithm I

You can place the figure and refer to it as Figure 2.1. The figure and table numbering will be run and updated automatically when you add/remove tables/figures from the document.

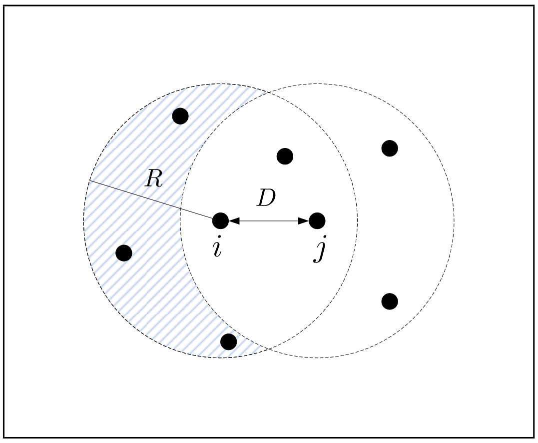

Figure 2.1 The network model

2.2.2 Algorithm II

Add more subsections as you want.

2.3 Development Tools

CHAPTER 3 PROPOSED WORK

Explain the design (how you plan to implement your work) of your project. Adjust the section titles below to suit the types of your work. Detailed physical design like circuits and source codes should be placed in the appendix.

3.1 System Architecture

Table 3.1 test table x1

SYMBOL		UNIT
α	Test variable	m^2
λ	Interarrival rate	jobs/ second
μ	Service rate	jobs/ second

3.2 System Specifications and Requirements

3.3 Hardware Module 1

3.3.1 Component 1

3.3.2 Logical Circuit Diagram

3.4 Hardware Module 2

3.4.1 Component 1

3.4.2 Component 2

3.5 Path Finding Algorithm

3.6 Database Design

3.7 GUI Design

CHAPTER 4 IMPLEMENTATION RESULTS

You can title this chapter as **Preliminary Results** or **Work Progress** for the progress reports. Present implementation or experimental results here and discuss them.

CHAPTER 5 CONCLUSIONS

This chapter is optional for proposal and progress reports but is required for the final report.

5.1 Problems and Solutions

State your problems and how you fixed them.

5.2 Future Works

What could be done in the future to make your projects better.

APPENDIX A

FIRST APPENDIX TITLE

Put appropriate topic here

This is where you put hardware circuit diagrams, detailed experimental data in tables or source codes, etc..

This appendix describes two static allocation methods for fGn (or fBm) traffic. Here, λ and C are respectively the traffic arrival rate and the service rate per dimensionless time step. Their unit are converted to a physical time unit by multiplying the step size Δ. For a fBm self-similar traffic source, Norros [**?**] provides its EB as

$$C = \lambda + (\kappa(H)\sqrt{-2\ln\epsilon})^{1/H} a^{1/(2H)} x^{-(1-H)/H} \lambda^{1/(2H)} \tag{A.1}$$

where $\kappa(H) = H^H(1-H)^{(1-H)}$. Simplicity in the calculation is the attractive feature of (A.1).
The MVA technique developed in [**?**] so far provides the most accurate estimation of the loss probability compared to previous bandwidth allocation techniques according to simulation results. Consider a discrete-time queueing system with constant service rate C and input process λ_n with $\mathbb{E}\{\lambda_n\} = \lambda$ and $\mathrm{Var}\{\lambda_n\} = \sigma^2$. Define $X_n \equiv \sum_{k=1}^{n} \lambda_k - Cn$. The loss probability due to the MVA approach is given by

$$\varepsilon \approx \alpha e^{-m_x/2} \tag{A.2}$$

where

$$m_x = \min_{n\geq 0} \frac{((C-\lambda)n + B)^2}{\mathrm{Var}\{X_n\}} = \frac{((C-\lambda)n^* + B)^2}{\mathrm{Var}\{X_{n^*}\}} \tag{A.3}$$

and

$$\alpha = \frac{1}{\lambda\sqrt{2\pi\sigma^2}} \exp\left(\frac{(C-\lambda)^2}{2\sigma^2}\right) \int_C^\infty (r - C) \exp\left(\frac{(r-\lambda)^2}{2\sigma^2}\right) dr \tag{A.4}$$

For a given ε, we numerically solve for C that satisfies (A.2). Any search algorithm can be used to do the task. Here, the bisection method is used.
Next, we show how $\mathrm{Var}\{X_n\}$ can be determined. Let $C_\lambda(l)$ be the autocovariance function of λ_n. The MVA technique basically approximates the input process λ_n with a Gaussian process, which allows $\mathrm{Var}\{X_n\}$ to be represented by the autocovariance function. In particular, the variance of X_n can be expressed in terms of $C_\lambda(l)$ as

$$\mathrm{Var}\{X_n\} = nC_\lambda(0) + 2\sum_{l=1}^{n-1}(n-l)C_\lambda(l) \tag{A.5}$$

Therefore, $C_\lambda(l)$ must be known in the MVA technique, either by assuming specific traffic models or by off-line analysis in case of traces. In most practical situations, $C_\lambda(l)$ will not be known in advance, and an on-line measurement algorithm developed in [**?**] is required to jointly determine both n^* and m_x. For fGn traffic, $\mathrm{Var}\{X_n\}$ is equal to $\sigma^2 n^{2H}$, where $\sigma^2 = \mathrm{Var}\{\lambda_n\}$, and we can find the n^* that minimizes (A.3) directly. Although λ can be easily measured, it is not the case for σ^2 and H. Consequently, the MVA technique suffers from the need of prior knowledge traffic parameters.

APPENDIX B

SECOND APPENDIX TITLE

Put appropriate topic here

Next, we show how $\mathrm{Var}\{X_n\}$ can be determined. Let $C_\lambda(l)$ be the autocovariance function of λ_n. The MVA technique basically approximates the input process λ_n with a Gaussian process, which allows $\mathrm{Var}\{X_n\}$ to be represented by the autocovariance function. In particular, the variance of X_n can be expressed in terms of $C_\lambda(l)$ as

$$\mathrm{Var}\{X_n\} = nC_\lambda(0) + 2\sum_{l=1}^{n-1}(n-l)C_\lambda(l) \tag{B.1}$$

Add more topic as you need

Therefore, $C_\lambda(l)$ must be known in the MVA technique, either by assuming specific traffic models or by off-line analysis in case of traces. In most practical situations, $C_\lambda(l)$ will not be known in advance, and an on-line measurement algorithm developed in [**?**] is required to jointly determine both n^* and m_x. For fGn traffic, $\mathrm{Var}\{X_n\}$ is equal to $\sigma^2 n^{2H}$, where $\sigma^2 = \mathrm{Var}\{\lambda_n\}$, and we can find the n^* that minimizes (A.3) directly. Although λ can be easily measured, it is not the case for σ^2 and H. Consequently, the MVA technique suffers from the need of prior knowledge traffic parameters.