

修士(工学)学位論文

**Master's Thesis of Engineering**

並列分散フレームワーク Christie の設計

2018 年 3 月

March 2018

照屋 のぞみ

**NOZOMI TERUYA**



琉球大学

大学院理工学研究科

情報工学専攻

**Information Engineering Course**

**Graduate School of Engineering and Science**

**University of the Ryukyus**

指導教員：教授 和田 知久

**Supervisor: Prof. Tomohisa WADA**

本論文は、修士(工学)の学位論文として適切であると認める。

論 文 審 査 会

(主 査) 和田 知久 印

(副 査) 岡崎 威生 印

(副 査) 名嘉村 盛和 印

(副 査) 河野 真治 印

# 目次

第1章	分散プログラミングの信頼性向上	1
第2章	分散フレームワーク <b>Alice</b> の概要	2
2.1	CodeSegment と DataSegment	2
2.2	DataSegmentManager	2
2.3	Data Segment API	2
2.4	CodeSegment の記述方法	2
2.5	TopologyManager	2
第3章	<b>Alice</b> の問題点	3
3.1	直感的でない API	3
3.2	型がわからない	3
3.3	DataSegmentManager を複数持てない	3
3.3.1	TopologyManager の NAT 超え機能	3
第4章	分散フレームワーク <b>Christie</b> の設計	4
4.1	API の改善	4
第5章	<b>Christie</b> の評価	5
5.1	Akka	5
5.2	Corba	5
5.3	Erlang	5
5.4	Spark	5
5.5	Hazelcast	5
第6章	まとめ	6
第7章	今後の課題	7
第8章	謝辞	8

参考文献	9
発表履歴	11
付録	11

# 目 次

# 表 目 次

# リスト目次

# 第1章 分散プログラミングの信頼性向上

スマートフォンやタブレット端末の普及率が増加している。それに伴いインターネット利用者数も増加しており、ネットワーク上のサービスには、信頼性とスケーラビリティが要求される。ここでいう信頼性とは、定められた環境下で安定して仕様に従った動作を行うことを指す。またスケーラビリティとは、スケーラビリティとは、分散ソフトウェアに対して単純にノードを追加するだけで性能を線形的に上昇させることができる性質である。しかし、これらをもつ分散プログラムをユーザーが一から記述することは容易ではない。

これらの問題の解決のために、当研究室ではデータを Data Segment、タスクを Code Segment という単位で記述するプログラミング手法を導入した分散フレームワーク Alice の開発を実現した。Data Segment は整数や文字列や構造体などの基本的なデータの集まりである。Code Segment は入力となる Data Segment が全て揃ったら処理を開始し計算結果の Data Segment を出力するタスクである。

Alice が実用的な分散アプリケーションを記述でき、仕様の変更を抑えた信頼性の高い拡張を可能にするということは、水族館の例題や TreeVNC の例題から確認された。しかし、Alice では API 設計が直感的でなく、型の整合性がとれない問題があった。また、Alice に NAT 越えの機能を実装しようとした際、Data Segment Manager が 1 つしか持てないたために拡張が困難であることが分かった。

本研究では、Alice から得られた知見をもとに、分散フレームワーク Christie の設計を行う。



## 第2章 分散フレームワーク Alice の概要

2.1 CodeSegment と DataSegment

2.2 DataSegmentManager

2.3 Data Segment API

2.4 CodeSegment の記述方法

2.5 TopologyManager

## 第3章 Aliceの問題点

3.1 直感的でない API

3.2 型がわからない

3.3 DataSegmentManager を複数持てない

3.3.1 TopologyManager の NAT 超え機能

# 第4章 分散フレームワーク Christie の設計

## 4.1 API の改善

## 第5章 Christieの評価

5.1 Akka

5.2 Corba

5.3 Erlang

5.4 Spark

5.5 Hazelcast

## 第6章 まとめ

## 第7章 今後の課題

## 第8章 謝辞

本研究の遂行、また本論文の作成にあたり、ご多忙にも関わらず終始懇切なる御指導と御教授を賜りました河野真治准教授に深く感謝いたします。

そして、数々の貴重な御助言と技術的指導を戴いた伊波立樹さん、並びに並列信頼研究室の皆様、本研究を遂行するにあたり参考にさせていただいた先行研究の Alice, Federated Linda, Jungle, TreeVNC の設計・実装に関わった全ての方々に感謝いたします。

また、本フレームワークの名前の由来となったクリスティー式戦車の生みの親、ジョン・W・クリスティーに敬意を評します。

最後に、日々の研究生生活を支えてくださった米須智子さん、情報工学科の方々、そして家族に心より感謝いたします。

## 参考文献

- [1] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and Types*. Cambridge University Press, New York, NY, USA, 1989.
- [2] Joachim (mathématicien) Lambek and P. J. Scott. *Introduction to higher order categorical logic*. Cambridge studies in advanced mathematics. Cambridge University Press, Cambridge, New York (N. Y.), Melbourne, 1986.
- [3] Michael Barr and Charles Wells. *Category Theory for Computing Science*. International Series in Computer Science. Prentice-Hall, 1990. Second edition, 1995.
- [4] Eugenio Moggi. Notions of computation and monads. *Inf. Comput.*, Vol. 93, No. 1, pp. 55–92, July 1991.
- [5] M. P. Jones and L. Duponcheel. Composing monads. Research Report YALEU/DCS/RR-1004, Yale University, December 1993.
- [6] Tokumori Kaito and Kono Shinji. The implementation of continuation based c compiler on llvm/clang 3.5. *IPSJ SIG Notes*, Vol. 2014, No. 10, pp. 1–11, may 2014.
- [7] 信康大城, 真治河野. Continuation based c の gcc4.6 上の実装について. 第 53 回プログラミング・シンポジウム予稿集, 第 2012 巻, pp. 69–78, jan 2012.
- [8] The agda wiki. <http://wiki.portal.chalmers.se/agda/pmwiki.php>. Accessed: 2016/01/20(Fri).
- [9] Welcome to agda' s documentation! — agda 2.6.0 documentation. <http://agda.readthedocs.io/en/latest/index.html>. Accessed: 2016/01/31(Tue).
- [10] Welcome! — the coq proof assistant. <https://coq.inria.fr/>. Accessed: 2016/01/20(Fri).
- [11] Ats-pl-sys. <http://www.ats-lang.org/>. Accessed: 2016/01/20(Fri).



- [12] Spin - formal verification. <http://spinroot.com/spin/whatispin.html>. Accessed: 2016/01/20(Fri).
- [13] Nusmv home page. <http://nusmv.fbk.eu/>. Accessed: 2016/01/20(Fri).
- [14] The cbmc homepage. <http://www.cprover.org/cbmc/>. Accessed: 2016/01/20(Fri).
- [15] Opencl — nvidia developer. <https://developer.nvidia.com/opencl>. Accessed: 2016/02/06(Mon).
- [16] Cuda zone — nvidia developer. <https://developer.nvidia.com/cuda-zone>. Accessed: 2016/02/06(Mon).
- [17] 翔平小久保, 立樹伊波, 真治河野. Monadに基づくメタ計算を基本とする gears os の設計. Technical Report 16, 琉球大学大学院理工学研究科情報工学専攻, 琉球大学工学部情報工学科, 琉球大学工学部情報工学科, may 2015.
- [18] 徳森海斗. Llvm clang 上の continuation based c コンパイラ の改良. Master's thesis, 琉球大学 大学院理工学研究科 情報工学専攻, 2016.
- [19] 小久保翔平. Code segment と data segment を持つ gears os の 設計. Master's thesis, 琉球大学 大学院理工学研究科 情報工学専攻, 2016.
- [20] 河野 真治比嘉 健太. Continuation based c を用いたプログラムの検証手法.
- [21] Benjamin C. Pierce. *Types and Programming Languages*. The MIT Press, 1st edition, 2002.
- [22] B.C. Pierce. 型システム入門プログラミング言語と型の理論:. オーム社, 2013.
- [23] Ulf Norell. Dependently typed programming in agda. In *Proceedings of the 4th International Workshop on Types in Language Design and Implementation, TLDI '09*, pp. 1–2, New York, NY, USA, 2009. ACM.
- [24] John Backus. The history of fortran i, ii, and iii. *SIGPLAN Not.*, Vol. 13, No. 8, pp. 165–180, August 1978.
- [25] Peter J. Landin. The mechanical evaluation of expressions. *Computer Journal*, Vol. 6, No. 4, pp. 308–320, January 1964.
- [26] Alonzo Church. *The Calculi of Lambda-Conversion*. Princeton University Press, Princeton, New York, 1941.

- [27] P. Hudak, S. Peyton Jones, and P. Wadler (editors). Report on the Programming Language Haskell, A Non-strict Purely Functional Language (Version 1.2). *ACM SIGPLAN Notices*, Vol. 27, No. 5, May 1992.
- [28] N.G de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the church-rosser theorem. *Indagationes Mathematicae (Proceedings)*, Vol. 75, No. 5, pp. 381 – 392, 1972.