

# Code Gear、Data Gear に基づく OS のプロトタイプ

仲松 栞<sup>†1</sup> 照屋 のぞみ<sup>†2</sup> 河野 真治<sup>†2</sup>

NAKAMATSU SHIORI,<sup>†1</sup> ERUYA NOZOMI<sup>†2</sup> and SHINJI KONO<sup>†2</sup>

1. 研究目的と背景
2. 非破壊的木構造データベース Jungle
3. Index
4. Index の差分 Update
5. Differential Jungle Tree

CbC(Continuation based C)<sup>1)</sup> 図1は

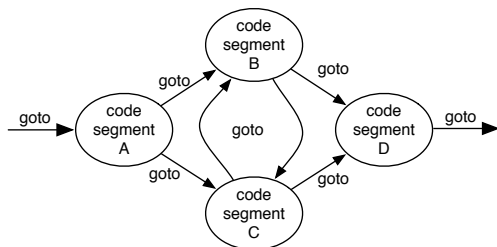


図1 gotoによるCode Segment間の接続

## 6. 非破壊 Red Black Tree の実装

## 7. Gears OS の構成

- Context
- TaskQueue

- TaskManager
- Persistent Data Tree
- Worker

Data Gear の Allocation 用の情報

## 8. TaskQueue

G

## 9. Persistent Data Tree

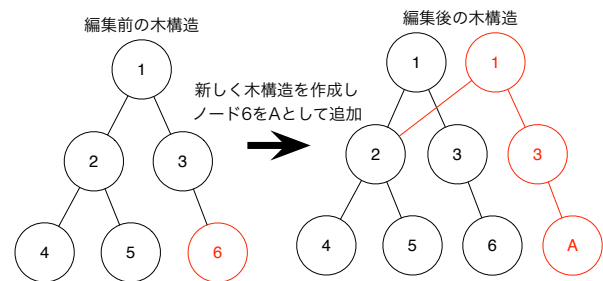


図2 木構造の非破壊的編集

- 各ノードは赤または黒の色を持つ。
  - ルートの色は黒である。
  - 赤ノードは2つの黒ノードを子として持つ(赤ノードが続くことはない)。
  - ルートから最下位ノードへのパスに含まれる黒ノードの数はどの最下位ノードでも一定である。
- これらの条件によってルートから最も遠い最下位ノードへのパスの長さはルートから最も近い最下位ノードへのパスの長さの2倍に収まることが保証される。

<sup>†1</sup> 琉球大学大学院理工学研究科情報工学専攻  
Interdisciplinary Information Engineering, Graduate  
School of Engineering and Science, University of the  
Ryukyus.

<sup>†2</sup> 琉球大学工学部情報工学科  
Information Engineering, University of the Ryukyus.

## 10. TaskManager

### 11. プロトタイプ of 動作

- 配列サイズを元に index, alignment, 配列へのポインタを持つ Data Gear に分割。
  - Data Gear を Persistent Data Tree に挿入。
  - 実行する Code Gear(Twice) と実行に必要な Data Gear への key を持つ Task を生成。
  - 生成した Task を TaskQueue に挿入。
  - Worker の起動。
  - Worker が TaskQueue から Task を取得。
  - 取得した Task を元に必要な Data Gear を Persistent Data Tree から取得。
  - 並列の処理される Code Gear(Twice) を実行。
- 要素数  $2^{17} * 1000$  のデータを 640 個の Task に分割し、コア数を変更して測定を行った結果を表 1、図 3 に示す。

Processor	Time(ms)
1 CPU	1315

表 1 要素数  $2^{17} * 1000$  のデータに対する Twice

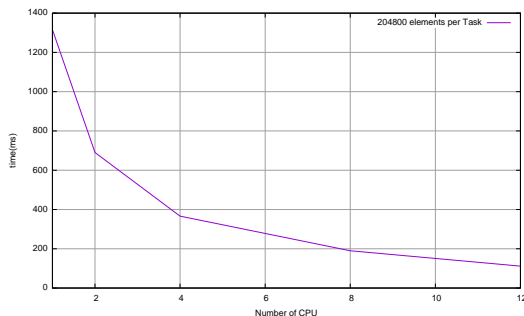


図 3 要素数  $2^{17} * 1000$  のデータに対する Twice

## 12. 分散環境での JungleDB の書き出し実験方法の提案

## 13. ま と め

### 参 考 文 献

- 1) TOKKMORI, K. and KONO, S.: Implementing Continuation based language in LLVM and Clang, *LOLA 2015* (2015).
- 2) 宮國 渡, 河野真治, 神里 晃, 杉山千秋: Cell 用の Fine-grain Task Manager の実装, 情報処理学会システムソフトウェアとオペレーティング・システム研究会 (OS) (2008).

- 3) 照屋のぞみ, 河野真治: 分散フレームワーク Alice の PC 画面配信システムへの応用, 第 57 回プログラミング・シンポジウム (2016).
- 4) 河野真治, 杉本 優: Code Segment と Data Segment によるプログラミング手法, 第 54 回プログラミング・シンポジウム (2013).
- 5) Aaftab Munshi, Khronos OpenCL Working Group: *The OpenCL Specification Version 1.0* (2007).
- 6) : CUDA, <https://developer.nvidia.com/category/zone/cuda-zone/>.
- 7) 小久保翔平, 伊波立樹, 河野真治: Monad に基づくメタ計算を基本とする Gears OS の設計, 情報処理学会システムソフトウェアとオペレーティング・システム研究会 (OS) (2015).