

# Cerium Task Manager におけるマルチコア上での並列実行機構の実装

當 眞 大 千<sup>†</sup> 金 城 裕<sup>†</sup> 河 野 眞 治<sup>†</sup>

本研究室で作成した Cerium Task Manager<sup>1)</sup> は, Task 単位で記述するゲームフレームワークである. 今までは, PlayStation 3/Cell<sup>2)</sup> 上でのみ, 並列実行を可能にしていたが, 今回新たに Mac OS X, Linux 上での並列実行に対応した. 本論文では, 既存の Cerium Task Manager の実装と新たに実装した並列実行の機構について説明する.

## Implementation of Cerium Parallel Task Manager on Multi-core

DAICHI TOMA,<sup>†</sup> YUTAKA KINJO<sup>†</sup> and SHINJI KONO<sup>†</sup>

We have developed Cerium Task Manager<sup>1)</sup> that is a Game Framework on the PlayStation 3/Cell<sup>2)</sup>. Cerium Task Manager new supporting parallel execution on Mac OS X and Linux. In this paper, we described implementation of existing Cerium Task Manager and a new parallel execution.

### 1. はじめに

プロセッサメーカーは, 消費電力, 発熱及びクロックの限界という観点から, マルチコア構成の路線を打ち出しており, 今後ますますマルチコアプロセッサが主流になると想像できる.

マルチコアプロセッサ上で, リソースを有効活用するためには, 並列プログラムを行う必要があるが, 効率の良い並列プログラムを書くことは難しい.

そこで, 本研究室で作成した Cerium Task Manager<sup>1)</sup> をマルチコアプロセッサに対応させることで, マルチコアプロセッサ上での Task 単位による並列プログラミングをサポートする.

今まで, Cerium Task Manager は, PlayStation 3/Cell<sup>2)</sup> 上でのみ, 並列実行を可能にしていたが, 今回新たに Mac OS X, Linux 上での並列実行に対応した.

本論文では, まず既存の Cerium Task Manager の実装について説明する. その後, 新たに実装した並列実行の機構について説明する.

### 2. Cerium Task Manager

Cerium Task Manager は, Cell 用に開発されたゲームフレームワークであり, Rendering Engine を

含む.

Cerium Task Manager では, 並列処理を Task 単位で記述する. 関数やサブルーチンを Task として扱い, Task には, input データ, output データ及び依存関係を設定する. Cerium Task Manager によってこれらの Task は管理され, 実行される.

Cerium Task Manager は, PlayStation 3/Cell, Mac OS X 及び Linux 上で利用することができ, それぞれのプラットフォームで同じプログラムを動作させることができる. これにより, アーキテクチャに依存しないプログラムを記述することが可能である.

Cerium Task Manager では, プログラムの様々なレベルでパイプラインが構成されるので, プログラムの性能向上が見込める (図 1).

また, Task 自体は入力データから, 出力データを計算するだけなので非常に単純だが, その入出力データをダブルバッファリングとして切り替えたり, 適切な並列度が得られるように徐々に生成するのは非常に煩雑となる. さらに, これらのデータ管理は, 並列実行を行うアーキテクチャに特化した処理が必要となる<sup>3)</sup>. Cerium Task Manager を利用することで, このような処理を代わりに行ってくれるため, 並列計算の実装に集中することができる.

### 3. マルチコア上での並列実行の機構

PlayStation 3/Cell 上の場合, 各 SPE に Task が割り当てられ, 並列に実行される.

<sup>†</sup> 琉球大学  
University of the Ryukyus

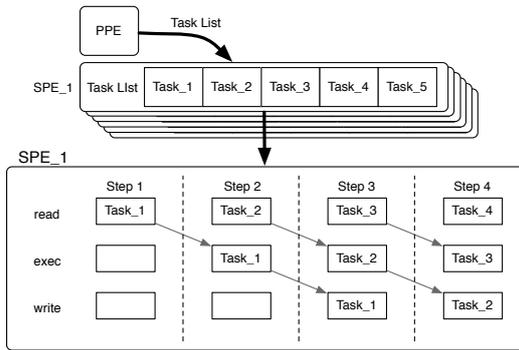


図 1 Scheduler

今回新たに、Mac OS X, Linux 上でも並列に実行させることを可能にした。これは、PlayStation 3/Cell の Mailbox に対応させる形で、Synchronized Queue を用いて Mac OS X, Linux 側の Cerium Task Manager へ移植したものである。操作しているスレッドが常に 1 つになるよう、パイナリセマフォで管理されている。各スレッドは、input 用と output 用として Synchronized Queue を 2 つ持っており、管理スレッドからタスクを受けて並列に実行するようになっている。

また、PlayStation 3/Cell と違い各 CPU で同じメモリ空間が利用できるため、DMA 転送を用いていた箇所をポインタ渡しをするように修正し、速度の向上を図った。

#### 4. ベンチマーク

Word Count, Sort 及び Prime Counter の例題を用いて、計測した。比較対象として、PlayStation 3/Cell においても同様の例題を用いて計測している。表 1 に結果を示す。

##### 実験環境

CentOS/Xeon

- OS : CentOS 6.0
- CPU : Intel®Xeon®X5650 @2.67GHz \* 2
- Memory : 128GB
- Compiler : GCC 4.4.4

PlayStation 3/Cell

- OS : Yellow Dog Linux 6.1
- CPU : Cell Broadband Engine @ 3.2GHz
- Memory : 256MB
- Compiler : GCC 4.1.2

表 1 より、CentOS 上で 6 CPU を利用した場合、1 CPU を利用した場合と比較して Word Count の例題で約 5.1 倍、Sort の例題で約 5.2 倍、Prime Counter の例題で、約 5.3 倍の速度向上が見られる。しかしながら、24 CPU を利用した場合、12 CPU を利用した

表 1 Benchmark

	Word Count	Sort	Prime Counter
1 CPU (Cell)	2381 ms	6244 ms	2081 ms
6 CPU (Cell)	1268 ms	1111 ms	604 ms
1 CPU (Xeon)	354 ms	846 ms	266 ms
6 CPU (Xeon)	70 ms	163 ms	50 ms
12 CPU (Xeon)	48 ms	127 ms	36 ms
24 CPU (Xeon)	40 ms	100 ms	31 ms

場合と比較して速度は上がっているものの速度向上率が落ちている。これは並列化率が低いために性能を活かすことができず、速度向上が頭打ちになっているとアムダールの法則<sup>4)</sup>から考えられる。並列化率の向上は今後の課題である。

#### 5. まとめ

本稿では、既存の Cerium Task Manager の実装と新しい並列実行の機構について説明した。新しく実装した並列実行の機構を用いることによって、Mac OS X, Linux 上でのマルチプロセッサ環境に対応できる。

今後の課題として、並列化率を向上させ、プロセッサ数が増えた時の速度向上率を改善する。また、現在の Cerium Task Manager は Task の種類が増え、Open CL<sup>5)</sup> に比べても記述が煩雑であるなどの欠点がある。これは Task の依存関係を、ユーザ側ではなくシステム側が記述するようにすることで解決できると考える。

#### 参考文献

- 1) 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, April 2008.
- 2) Sony Corporation. Cell BroadbandEngine™アーキテクチャ, 2006.
- 3) 金城裕, 河野真治. Cerium における datasegment api の設計. 日本ソフトウェア科学会第 28 会大会, Sep 2011.
- 4) Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, and Doug Lea. *Java Concurrency in Practice*. Addison-Wesley Professional, 2005.
- 5) Aaftab Munshi, Khronos OpenCL Working Group. *The OpenCL Specification Version 1.0*, 2007.
- 6) International Business Machines Corporation, Sony Computer Entertainment Incorporated, Toshiba Corporation. *Cell Broadband Engine Linux Reference Implementation Application Binary Interface Specification*, 2007.