ＲＡＩＤ構成のソリッド・ステート・ドライブの シミュレーションプラットフォームの研究

Research on Simulation Platforms of Solid-State Drives with RAID Configuration

電気電子情報通信工学専攻 荒川　飛鳥
Asuka ARAKAWA

1　研究の背景・目的

パーソナルコンピューターや携帯電話の普及とインターネットの発達により、インターネットがない生活が考えられない世の中になった。情報機器の普及により流通する情報量は増加し、データを保存するストレージの容量も増加している。現在、記憶装置として主にハード・ディスク・ドライブ（HDD）が用いられており、単体の書き込み・読み出しといった入出力速度の高速化や高信頼化のために、HDD を並列化したディスク・アレイ技術である RAID（Redundant Arrays of Inexpensive Disks）[1] が用いられている。しかし、HDD は機械的に駆動する部品を用いて入出力を行っており、ネットワークや CPU の高速化に対して相対的なアクセス速度の低下が課題となっている。そこで、HDD よりも高速で低消費電力な記憶装置として NAND フラッシュメモリを用いたソリッド・ステート・ドライブ（SSD）が注目されている。

そのような背景の中、ソリッド・ステート・ドライブを HDD で行っていた RAID 構成を用いて高速化と高信頼化を実現する研究が盛んに行われている [2–5]。これらの論文では、ディスクの故障に対して復旧するための RAID の仕組みを利用し SSD の信頼性を高める技術や、その並列性を活かしより高速なストレージシステムを提案している。提案するアルゴリズムをシミュレーターを用いて実装することで、より短期間での評価が可能となり、多くの研究でシミュレーターが用いられている。

本論文では、シミュレーターを用いた RAID 構成の SSD の入出力性能や消費電力の評価を行うシミュレーションプラットフォームを開発した。RAID 構成の SSD シミュレーターは、竹内研究室で開発した単体 SSD のシミュレーションプラットフォームを拡張し、RAID コントローラーや並列動作機能を追加することで実装されている。さらに、動作の検証のために RAID1 のコントローラーを実装した。

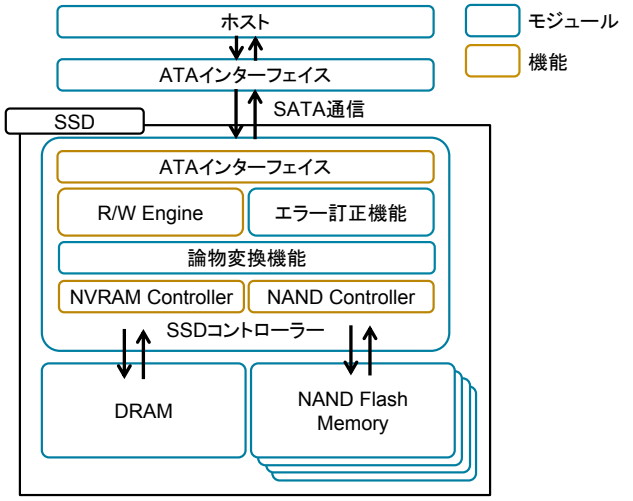

図 1　SSD シミュレーターの概要

2　単体 SSD のシミュレーションプラットフォーム

SSD の各モジュールを C++ でモデル化し、SystemC [6] により各モジュールを並列に独立して動作させることでシミュレーターを実装しているのが、SSD シミュレーターである（図 1）。各機能や構造、その通信などは全て C++ ベースで記述され、ホストや ATA インターフェイス、SSD コントローラーといった各モジュールは実際の製品では独立して動作するため、そのハードウェアモデリングを SystemC を使用しトランザクションベースのモデリング（TLM, Transaction Level Modeling）で実装している。各モジュールや機能は、自由に切り換え、追加が可能であることがこの SSD シミュレーターの特徴であり、実行時に与えるパラメーターによって設定と切り替えが可能な仕組みになっている。また、ホストには様々なアプリケーションが接続できるようにしてありデータベースシステム等のアプリケーションや、事業用サーバーから取得し

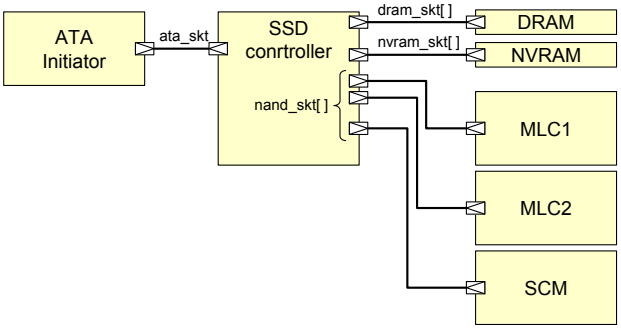

図 2　SSD シミュレーターの各モジュール間の接続図

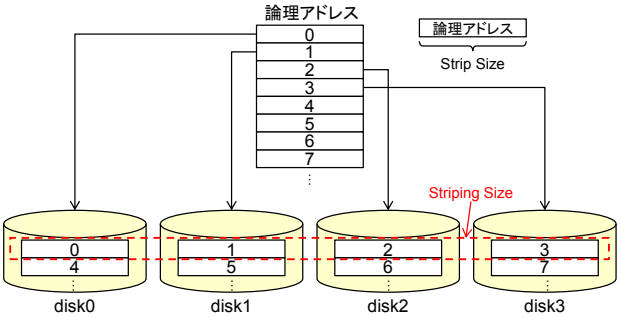

図 3　RAID0

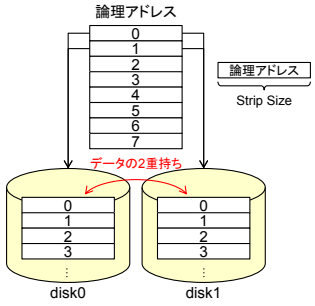

図 4　RAID1

たアクセスパターンのログ（リアルトレース）などが利用できる。

図 2 に、各モジュールの接続図を示す。独立して動作するモジュール間は、TLM レベルのソケットを用いて接続されており、モジュール間のデータの転送やメッセージのやり取りは全てこのソケットを介して行われる。ATA イニシエーターと SSD コントローラー間は 1 対のみだが、SSD コントローラーと MLC,SCM 等のメモリ間は SSD の構成によって動的にそのソケット数が変化する。

3　RAID の概要

RAID(Redudant Array of Independent Disks）[1]とは、安価で低容量のハード・ディスク・ドライブ（HDD）を用いて、大容量で信頼性の高いストレージを構築するための技術である。冗長性の持たせ方により RAID のレベルは複数あり、現在では主に

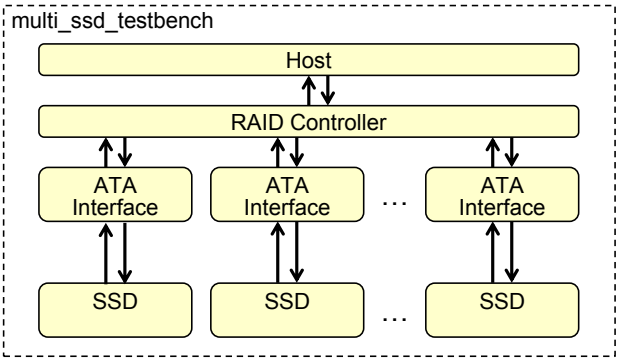

図 5　RAID 構成の SSD シミュレーションプラットフォーム

RAID0、RAID1、RAID5、RAID6 とその組み合わせが用いられている。図 3 は RAID0 の実装の概念図である。RAID0 はストライピングと呼ばれるように、データをスプリットサイズに分け複数のディスクに並列に書きこむことで高速化を図る。この RAID レベルは冗長性はない。図 4 に示す RAID1 はミラーリングと呼ばれるように、全く同じデータを 2 台のディスクに書きこむことでデータの損失を防ぐ仕組みである。この RAID レベルは読み出しのときに 2 台のディスクを並列動作させることで高速化を行っている。

4　RAID 構成 SSD のシミュレーションプラットフォームの開発

竹内研究室で実装されているソリッド・ステート・ドライブ（SSD）のシミュレーターは、単体の SSD のシミュレーションしかできなかった。そこで、シミュレーションプラットフォームを新規で実装し、複数台の SSD を並列化し同時に動作させることができるシミュレーション環境を構築した。図 5 に実装したプラットフォームの概要を示す。各 SSD は独立して動作させる必要があるため、SSD ごとにリクエストを処理するためのスレッドとして ATA Interface の階層がある。また、ホストから来たリクエスト処理及び、各 SSD へのリクエストの生成を担うのが RAID コントローラーの階層である。SSD の台数及び、各 SSD の特性は任意に指定できるようにした。

図 6 には、RAID 構成 SSD におけるソケットの接続を表す。既存の単体 SSD のシミュレーターでは、SSD コントローラーや NAND フラッシュメモ

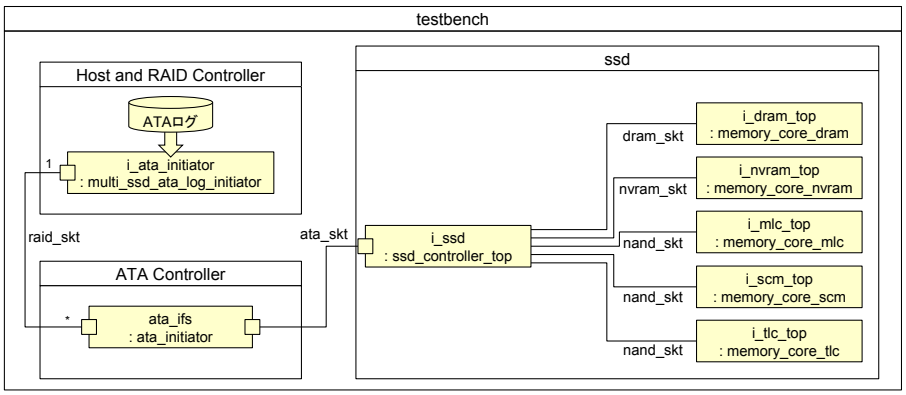

図 6　RAID 構成の SSD シミュレーションプラットフォームのモジュールの配置

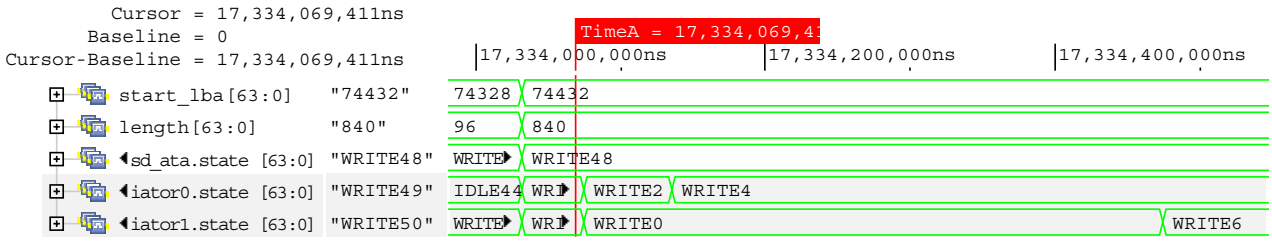

図 7　並列動作の検証結果

リスト 1　SSD2 台を用意する際の実行コマンド

```
1 ./main.x --multi_ssd 2 --use_log=/localdisk/
  ata_log/tpcc-new.log --log_v 0 --identify 2 --
  param "--rw_engine 3 --lp_mapper_type 3 --
  dump_write_step 10 --dump_write_window 10
  --xml=setting.xml @--rw_engine 3 --
  lp_mapper_type 3 --dump_write_step 10 --
  dump_write_window 10 --xml=setting.xml"
```

リといった SSD 内のモジュールとシリアル ATA で SSD と接続された ATA イニシエーターが同じ階層にあり、SSD のみを増やすことができなかった。そこで、SSD のモジュールを” ssd” という単位でひとまとめにし、それとは別に ATA コントローラーと RAID コントローラーを実装し、別々の階層に分けた。また、RAID コントローラーの共通する機能を親クラスで定義し、その派生クラスで RAID レベルに応じて設計しなければならない機能を実装するようにしたことで、容易に新たな RAID アルゴリズムを実装できるようにした。

シミュレーションの実行時には、単体 SSD のシミュレーションを行う際と同じようにコマンドライン引数を用いて SSD の特性等を設定する。RAID 構成 SSD のシミュレーション時には、リスト 1 のように、” @” 区切りで生成する台数分のパラメーターを設定することで、各ディスクごとに別々のチップ構成やアルゴリズムを使用できるようにした。

5　RAID1 構成 SSD の性能評価

最後に、実装した RAID 構成 SSD のシミュレーションプラットフォームが正しく動作しているかを波形表示ソフトを用いて検証した。その波形を図 7 に示す。この波形は、リスト 1 のコマンドを用いて、SSD を 2 台並列に書込み命令を送ったものである。下の 2 つの波形に各 SSD で現在処理している内容が表示されている。説明の都合上、最も下の波形を SSD1、下から 2 番目の波形を SSD0 と呼ぶ。波形のはじめをみると、SSD1 のステータスが WRITE となっているが、SSD0 は IDLE のステータスになっており、SSD1 の動作が終了するのを待っていることがわかる。時間が経過し、SSD1 の処理が終わると SSD0 と SSD1 が同時に WRITE ステータスとなっており、同時に書き込み動作を行っていることがわかる。

表 1　R/W 混在のリアルトレースの特性

トレース名	種別	ユーザー領域 (GB)	総書き込み量 (GB)	総読み出し量 (GB)	平均書き込みリクエスト長 (KB)	平均読み出しリクエスト長 (KB)
financial1	金融サーバー	0.53	14.72	2.65	3.76	2.25
financial2	金融サーバー	0.46	2.13	6.62	3.32	2.28
websearch2	web サーバー	0.43	0.43	5.69	15.86	17.14
hm_0	ハードウェアモニタリング	2.54	21.67	9.96	8.56	7.37
proj_0	事業別共用ディレクトリ	3.31	145.96	8.97	40.19	17.84
proj_3	事業別共用ディレクトリ	5.9	6.7	18.24	18.33	8.99
prxy_0	プロキシサーバー	0.98	54.05	3.05	4.66	8.33
prxy_1	プロキシサーバー	4.52	75.59	129.62	13.53	12.33
web_0	web SQL サーバー	7.32	18.55	17.36	10.38	30.00

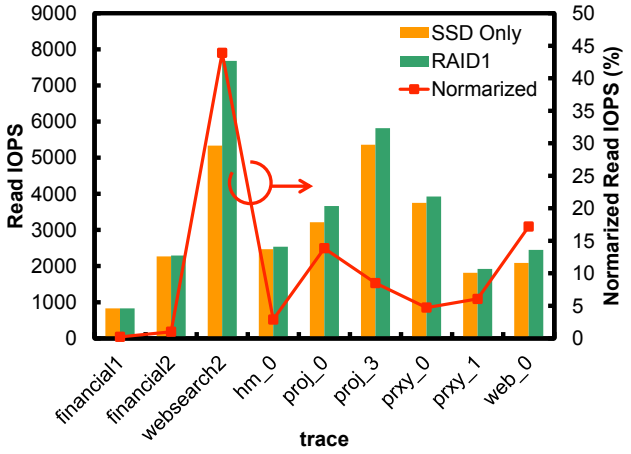

図 8　RAID1 構成の SSD の読み出し IOPS

6　RAID1 構成 SSD の性能評価

開発したシミュレーションプラットフォームを用いて、RAID1 構成の SSD と単体の SSD との読み出し速度を比較した。検証は、表 1 に示す特性の異なる 9 つのリアルトレースを用いて行った。その結果を図 8 に示す。SSD Only は 1chip 構成の SSD1 台を、RAID1 は SSD Only で用いたディスク 2 台で RAID1 構成にしたものである。図中の左縦軸（折れ線グラフ）は、SSD Only を基準として読み出し IOPS を正規化し改善率を表した。シミュレーションの結果、最も性能差があったのは websearch2 で、その差は 43.9% であった。全てのトレースで平均して 10.93% 読み出し IOPS が向上した。RAID1 構成では、各ディスクに対して同時に読み出しリクエストを送っているが、どちらかのディスクが遅い場合や、単一ディスクの場合と処理時間に差がない場合は読み出し IOPS は向上しない。そのため、読み出し IOPS は 2 倍とならなかったと考えられる。

7　結論と今後の展望

本論文では、RAID 構成 SSD のシミュレーションプラットフォームを開発した。各ディスクごとに構成やアルゴリズムを替えられ、RAID レベルの新規実装も容易な点が特徴である。RAID0 で SSD を用いた場合には最大で 43.9%、平均で 10.93% 読み出し IOPS が向上した。今後は、NAND フラッシュメモリの特性を考慮した SSD 向けのアルゴリズムを開発する必要がある。

発表論文

[1] Asuka Arakawa, Chao Sun and Ken Takeuchi, “Database Storage Engine and SSD Controller Co-design for SSD Performance Enhancement and Energy Reduction”, IEEE Non-Volatile Memory Workshop (NVMW), Mar 2015.

[2] 荒川飛鳥, 孫超, 宮地幸祐, 竹内健 “SSD コントローラーとミドルウェアの協調設計” IEEE 電子情報通信学会, 2014 年 4 月.

[3] 荒川飛鳥, 孫超, 竹内健 “ソリッド・ステート・ドライブとリレーショナルデータベースの統合アーキテクチャ”, 電子情報通信学会, 2015 年 5 月.

参考文献

[1] P. David, et al. "A Case for Redundant Arrays of Inexpensive Disks (RAID)." SIGMOD Conference, pp.109-116, 1988.

[2] S. Koo, et al."Dual RAID technique for ensuring high reliability and performance in SSD." ICIS, pp.399-404, 2015.

[3] X. Wu, et al."RAID-Aware SSD: Improving the Write Performance and Lifespan of SSD in SSD-based RAID-5 System." BdCloud, pp.99-103, 2014.

[4] K. Park, et al."Reliability and Performance Enhancement Technique for SSD array storage system using RAID mechanism." pp.140-145, 2009.

[5] J. Wan,et al."S2-RAID: A New RAID Architecture for Fast Data Recovery."MSST, pp1-9, 2010.

[6] SystemC ホームページ http://www.systemc.org/