

# 패리티 캐시를 이용한 DDR 메모리 저장 장치용 RAID 레벨 5의 성능 분석

## Performance Analysis of Parity Cache enabled RAID Level 5 for DDR Memory Storage Device

구본근\*, 곽윤식\*, 정승국\*\*, 황정연\*\*

Bon-Gen Gu\*, Yun-Sik Kwak\*, Seung-Kook Cheong\*\*, and Jung-Yeon Hwang\*\*

### 요 약

본 논문에서는 DDR 메모리 기반의 저장 장치로 구성되는 패리티 캐시를 이용한 RAID 레벨-5의 성능을 시뮬레이션을 통해 분석하였다. 시뮬레이션을 위해 본 논문에서는 시뮬레이션 모델을 개발하고, 시뮬레이션을 통해 획득하고자 하는 성능 분석 자료를 도출하였다. 또 본 논문에서 개발한 시뮬레이션 모델을 기반으로 시뮬레이터를 구현하였으며, 이를 이용하여 다양한 파라미터를 이용하여 시뮬레이션을 수행하였다. 시뮬레이션 결과에 따라 응용 분야의 스토리지 접근 패턴을 튜닝하면 DDR 메모리 기반의 저장 장치로 구성되며 패리티 캐시를 이용한 RAID 레벨-5 스토리지 시스템이 효과적으로 이용될 수 있을 것으로 기대된다.

### Abstract

In this paper, we analyze the performance of the parity cache enabled RAID level-5 via the simulation. This RAID system consists of the DDR memory-based storage devices. To do this, we develop the simulation model and suggest the basic performance analysis data which we want to get via the simulation. And we implement the simulator based on the simulation model and execute the simulator. From the result of the simulation, we expect that the parity cache enabled RAID level-5 configured by the DDR memory based storage devices has the positive effectiveness to the enhancing of the storage system performance if the storage access patterns of applications are tuned.

Key words : RAID level-5 , DDR memory, storage, DDR based storage

### I. 서 론

고성능, 고신뢰성 컴퓨팅 분야의 연구 중에서 그 리드 컴퓨팅, 클라우드 컴퓨팅 등이 많은 연구, 개발의 대상이 되고 있다. 이러한 시스템은 대용량의 데

이터 대상으로 의미있는 처리를 수행하며, 처리를 위해 준비된 데이터 및 처리가 완료된 데이터들은 중앙의 스토리지 서버에 저장되기 보다는 시스템을 구성하고 있는 각 노드에 분산, 중복 저장되어 있다. 따라서 이러한 응용 분야에서는 데이터가 분산 저장되어

\* 충주대학교(Chungju National University)

\*\* 한국전자통신연구원(ETRI)

· 제1저자 (First Author) : 구본근

· 교신저자 : 곽윤식

· 투고일자 : 2010년 11월 9일

· 심사(수정)일자 : 2010년 11월 9일 (수정일자 : 2010년 12월 17일)

· 게재일자 : 2010년 12월 30일

있는 스토리지 시스템의 성능 및 신뢰성 향상이 주요 연구 주제 중의 하나이다.

스토리지 시스템의 성능과 신뢰성 확보를 위해서는 운영체제, 장치 구동기 등의 시스템 소프트웨어가 적절히 설계되어야 할 뿐만 아니라 데이터를 저장하는 저장 장치(storage device)의 성능과 신뢰성 확보가 중요하다. 특히 기계적인 메커니즘을 이용하여 데이터를 저장, 접근하는 자기 디스크는 컴퓨터 시스템의 성능 향상에 있어 주요 병목 요소 중 하나이다.

스토리지 시스템의 성능 및 신뢰성 향상을 위한 주요 연구 방향으로는 저장 장치의 성능과 신뢰성 확보를 위해 새로운 저장 매체의 개발 및 복수개의 저장 장치로 스토리지 구성하고 운영하기 위한 방안 등이 있다.

최근에 새로운 저장 매체로 소개되고 있는 것은 반도체 메모리 기술을 이용하는 것으로 NAND 플래시 메모리 기술을 이용한 SSD(Solid State Drive)와 컴퓨터의 주기억 장치로 사용되고 있는 DDR 메모리를 이용한 저장 장치[1,2]가 있다. NAND 메모리와 DDR 메모리의 동작 특성이 다르기 때문에 이를 이용한 저장 장치의 동작 특성도 달라 성능과 신뢰성 확보를 위한 방안들도 다르다.

저장 데이터에 대한 접근의 병렬성을 높이고, 저장 장치에서 발생할 수 있는 고장에 대처하기 위해 복수개의 저장 장치를 구성, 운영하는 대표적인 방안으로는 RAID(Redundant Array of Independent Disks)가 있다[3,4,5].

RAID는 저장 장치 접근의 병렬성을 통해 성능 향상을 이룰 수 있으며, 데이터 저장의 중첩성 또는 여분의 정보를 생성, 이용함으로써 저장 장치에 고장이 발생하여도 서비스를 계속할 수 있는 신뢰성을 제공한다. RAID는 병렬성과 여분의 정보 생성 등 구성 방법에 따라 RAID 레벨로 분류된다.

현재 일반적으로 이용되고 있는 RAID 레벨은 0, 1, 5 및 그 변형들이 있으며, 고성능과 고신뢰성을 위한 것으로 RAID 레벨-5가 있다. RAID 레벨-5의 특징으로는 한 저장 장치에 고장이 발생하여도 스토리지 시스템의 서비스는 계속 제공될 수 있도록 각 저장 장치에 저장되는 블록들을 이용하여 패리티 그룹을 형성하고, 이 그룹에 포함되는 데이터 블록들에 대한

패리티 블록을 생성, 저장한다는 것이다. 임의의 저장 장치에서 고장이 발생하여도 그 저장 장치의 데이터 블록이 속한 패리티 그룹의 정상적인 데이터 블록과 패리티 블록을 이용하여 고장이 발생한 저장 장치의 데이터 블록의 내용을 재생성하여 서비스를 할 수 있다.

앞서 기술한 것과 같은 RAID 레벨-5의 동작 특성으로 인해 데이터 블록 읽기 성능은 동일한 수의 저장 장치로 구성된 RAID 레벨-0의 성능과 비슷하다. 그러나 데이터 블록 쓰기 성능은 그 블록이 속한 패리티 그룹의 패리티 블록 수정을 위해 기본적으로 4회의 디스크 접근이 필요하기 때문에 RAID 레벨-0의 쓰기 성능에 비해 상대적으로 낮다.

RAID 레벨-5의 성능을 향상시키기 위한 다양한 기법들이 개발, 연구되고 있으나, 기존의 많은 고속화 기법들은 자기 디스크의 동작 특성을 고려하여 설계되었기 때문에 새로운 저장 매체인 플래시 메모리 SSD 또는 DDR 메모리 기반의 저장 장치로 RAID 시스템을 구성한 경우에는 최상의 성능을 얻지 못하고 있다[6].

[7]에서는 새로운 저장 매체로 연구, 개발되고 있는 DDR(-2 또는 -3) 메모리 기반의 디스크를 이용하여 RAID 레벨-5를 구성할 경우에 RAID 기능을 고속화할 수 있는 패리티 캐시를 제안하였다. [7]에서 제안한 패리티 캐시는 데이터 저장을 할 때 이미 저장되어 있는 내용과 배타적 논리합을 한 결과를 저장하기 때문에 패리티 연산을 위한 별도의 배타적 논리합 연산을 할 필요가 없다는 장점을 가지고 있다. 그러나 실제로 적용되기 위해서는 다양한 형태의 성능 평가가 선행되어야 한다.

본 논문에서는 [7]에서 제안한 패리티 캐시를 이용한 RAID 레벨-5의 동작 특성과 성능을 평가하기 위한 시뮬레이션 모델을 제안한다. 이를 위해 패리티 캐시를 이용한 RAID 레벨-5의 동작 흐름을 기술하고, 본 논문에서 시뮬레이션을 통해 획득하고자 하는 통계량을 정의한다.

제한한 시뮬레이션 모델을 기반으로 하여 구현한 시뮬레이터에 대해 기술하며, 앞서 기술한 통계량을 획득하기 위해 시뮬레이터에서 요구되는 각종 파라미터들을 기술한다. 또한 시뮬레이션을 수행한 결과

를 분석함으로써 패리티 캐시를 이용하는 RAID 레벨-5의 동작 특성 및 성능을 기술한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 패리티 캐시를 구성하는 가장 하위 레벨의 소자인 패리티 셀(parity cell)을 소개하고, 제 3 장에서는 이를 이용한 RAID 레벨-5의 동작 흐름과 분석을 위해 사용될 통계량을 정의하는 시뮬레이션 모델을 기술한다. 제 4 장에서는 시뮬레이션 모델에 따라 구현한 시뮬레이터와 이를 위한 시뮬레이션 파라미터를 제시한다. 제 5 장에서 시뮬레이션을 수행한 결과를 분석하여 패리티 캐시를 이용한 RAID 레벨-5의 동작 특성 및 성능을 기술하며, 제 6 장에서는 본 논문의 내용을 정리하고 추후 연구 과제에 대해 기술한다.

## II. 패리티 캐시

RAID 레벨-5의 데이터 블록 쓰기 성능 저하는 앞서 기술한 것과 같이 패리티 블록 계산을 위한 오버헤드가 그 원인이다. 이 오버헤드의 요소로는 추가적인 저장 장치 접근과 패리티를 계산하기 위한 반복적인 배타적 논리합 연산의 수행이다.

자기 디스크 기술을 이용하는 저장 장치에서 추가적인 데이터 블록 접근은 다른 컴퓨팅 요소의 처리 시간과 비교하면 매우 긴 시간이 필요하다. 따라서 기존의 RAID 레벨-5 제어기들은 고속화를 위해 저장 장치의 접근 횟수를 최소화하거나 미루기 위한 다양한 정책을 사용하고 있다.

이 경우에 저장 장치 접근 시간에 비해 패리티 블록을 계산하기 위한 배타적 논리합 연산 시간은 상대적으로 짧기 때문에 전체 서비스 시간에 미치는 영향은 크지 않다고 할 수 있다.

하지만 본 논문에서 고려하는 저장 장치는 기계적인 메커니즘을 갖는 자기 디스크 또는 반도체 저장 방식의 특성상 쓰기 연산에 긴 시간이 요구되는 플래시 메모리가 아니라 컴퓨터 시스템의 주기억장치로 사용되는 DDR 메모리로 구성되는 저장 장치이다. 따라서 자기 디스크 또는 플래시 메모리를 이용한 저장 장치와 비교할 때 이 저장 장치의 서비스 시간에 저장 장치 접근 시간 또는 접근 횟수가 미치는 영향은

상대적으로 적다고 할 수 있다.

따라서 패리티 블록 계산을 위한 배타적 논리합 연산 시간이 블록 입출력 서비스 시간에 미치는 영향이 상대적으로 크다고 할 수 있다. 따라서 배타적 논리합 연산 시간을 감소시키는 것이 입출력 서비스 시간의 개선에 기여할 수 있다.

이를 위해 [7]에서는 패리티 캐시와 그것을 이용한 RAID 레벨-5 제어기의 기본 구조를 제안하였다. 패리티 캐시는 데이터 블록을 저장하는 데이터 캐시와 달리 RAID 레벨-5의 패리티 블록만을 저장하는 특수 목적용 캐시라고 할 수 있다.

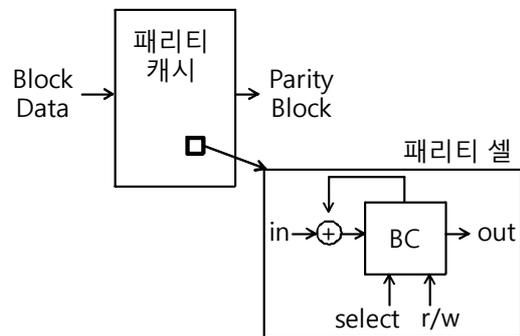


그림 1. 패리티 캐시 블록 다이어그램  
Fig. 1. Block Diagram of Parity Cache

그림 1은 [7]에서 제안한 패리티 캐시의 블록 다이어그램을 나타내고 있다. 패리티 캐시는 1 비트 정보를 저장하는 패리티 셀들로 구성되어 있다. 패리티 셀은 그림 1에 나타낸 것과 같이 외부에서 전송된 1 비트 데이터와 패리티 셀이 이미 저장하고 있던 데이터를 입력으로 하는 배타적 논리합 게이트의 출력을 저장한다. 이러한 동작 방법으로 인해 앞서 기술한 패리티 그룹에 속한 데이터 블록들을 패리티 캐시에 저장을 하면 패리티 캐시에는 패리티 블록이 생성, 저장된다. 따라서 패리티 블록을 생성하기 위해 RAID 레벨-5 제어기에서 별도의 배타적 논리합 연산을 수행할 필요가 없다.

## III. 패리티 캐시 RAID 시뮬레이션 모델

### 3-1. 패리티 캐시 RAID 레벨-5 동작 흐름

시뮬레이션 모델은 대상 시스템의 동작 특성을 잘 반영할 수 있어야 한다. 패리티 캐시를 사용하는 RAID 레벨-5—이하 RAID 레벨-5로 표기함—은 데이터 입출력의 효율을 위해 데이터 캐시도 이용한다. 그림 2는 RAID 레벨-5의 동작 흐름을 나타낸 것이다. 디스크 입출력 요구가 읽기인 경우와 쓰기인 경우에 동작 흐름은 달라진다.

읽기 요구인 경우에 디스크 입출력 요구의 대상이 되는 모든 데이터 블록이 데이터 캐시—이하 D-Cache로 표기함—에 존재하는가를 검사한다. 만약 모든 대상 데이터 블록이 D-Cache에 있다면 D-Cache에 저장되어 있는 데이터 블록을 이용하여 읽기 요구에 대응을 하면 된다.

만약 읽기 요구의 데이터 블록들 중에 일부 데이터 블록이 D-Cache에 적재되어 있지 않다면 저장 장치로부터 대상 데이터 블록을 읽어와 적재하여야 하기 때문에 D-Cache의 캐시 슬롯을 할당한 후 저장 장치 접근이 이루어진다. 이 과정은 읽기 요구의 대상이 되는 데이터 블록들 중에서 D-Cache에 적재되지 않은 블록의 수만큼 반복 수행된다.

쓰기 요구인 경우에는 데이터 블록이 속한 패리티 그룹의 패리티 블록을 수정하여야 하기 때문에 읽기 요구의 경우보다 많은 동작 과정이 필요하다. 쓰기 요구의 대상이 되는 데이터 블록이 D-Cache에 적재되어 있는가를 검사한다. 이것은 이후 패리티 블록을 갱신할 때 D-Cache에 쓰기 대상이 되는 데이터 블록이 D-Cache에 적재되어 있는가에 따라 처리가 달라지기 때문이다. 만약 D-Cache에 대상 데이터 블록이 적재되어 있지 않다면 D-Cache의 캐시 슬롯을 할당한다. 이 경우에 저장 장치에 저장되어 있는 데이터 블록을 읽어 올 필요가 없다.

D-Cache에 대한 검사가 끝난 후에는 패리티 캐시—이하 P-Cache로 표기함—에 대상 블록들이 속한 패리티 그룹의 패리티 블록이 적재되어 있는가를 검사한다. 만약 대상 패리티 블록이 P-Cache에 적재되어 있지 않다면 P-Cache의 캐시 슬롯을 할당한다. 새롭게 할당되었거나 이미 대상 패리티 블록을 적재하고 있는 캐시 슬롯의 패리티 블록을 갱신하기 위해서 데이터 블록을 저장하여야 한다. 앞서 기술한 것과 같이 패리티 캐시에 데이터 블록을 저장하는 것은 이미

저장하고 있는 내용과 배타적 논리합을 수행한 결과를 저장하기 때문에 데이터 블록을 패리티 캐시에 저장하는 것만으로 패리티 블록을 갱신할 수 있다. 이때 앞서 수행한 단계에서 대상 데이터 블록이 D-Cache에 적재되어 있었다면 이 데이터 블록은 갱신되기 전의 블록이며, 패리티 블록을 계산하기 위해 새로운 데이터 블록과 더불어 P-Cache에 적재되어야 한다.

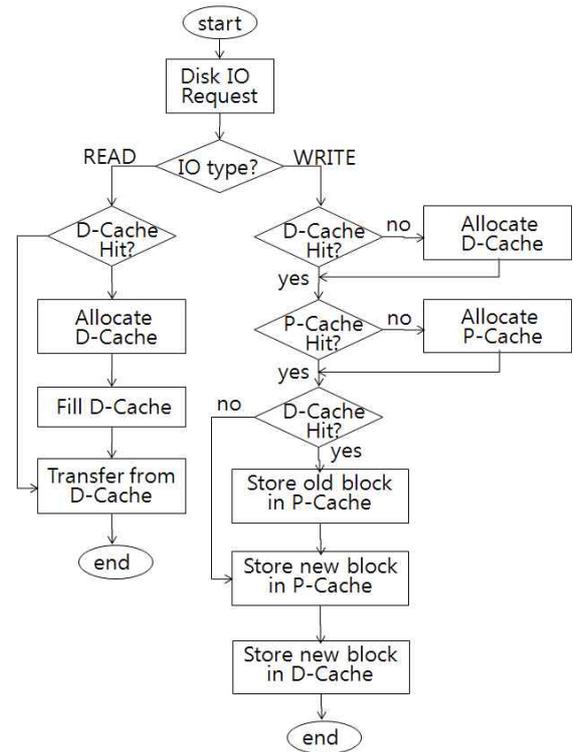


그림 2. 패리티 캐시 RAID 레벨-5 동작 흐름  
Fig. 2. Flowchart of Parity Cache RAID Level-5

앞서 기술한 과정을 통해 패리티 블록에 대한 처리가 완료된 후 최종적으로 데이터 블록을 D-Cache에 적재함으로써 RAID 레벨-5의 쓰기가 완료된다.

### 3-2. 분석을 위한 수집 통계량

시뮬레이션을 통해 RAID 레벨-5의 동작 특성 및 성능을 평가하기 위해 본 논문에서 다섯 종류의 통계량을 수집하고 분석한다.

RAID 레벨-5에서는 데이터 캐시와 패리티 캐시를 사용하고 있는데 시뮬레이션에서는 각종 파라미터에 따라 데이터 캐시와 패리티 캐시의 적중률을 수집한

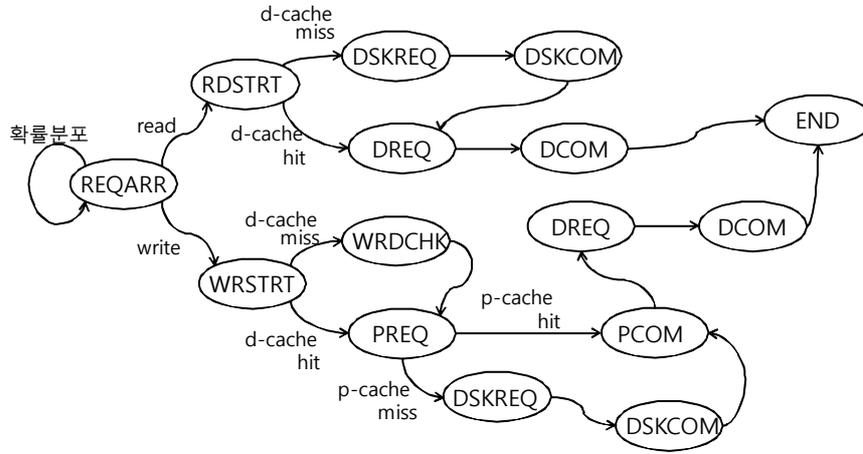


그림 3. 사건 흐름  
Fig. 3. Event Flow

다. 높은 캐시 적중률은 시스템의 성능 향상에 기여할 수 있어 RAID 레벨-5가 최상의 성능을 제공하기 위한 운영 환경 및 조건을 도출할 수 있다.

각 디스크 입출력 요구 당 평균 서비스 시간은 사용자의 관점에서 중요한 성능 평가 요소이다. 평균 서비스 시간은 많은 시스템 요소들에 의해 결정되지만 가장 짧은 평균 서비스 시간을 갖는 운영 조건을 검토함으로써 RAID 레벨-5가 최상의 성능을 나타낼 수 있는 응용 분야를 예측할 수 있다.

저장 장치 접근 당 평균 데이터 블록 수는 디스크 입출력 요구에서의 블록 수와 캐시의 적중률 등에 영향을 받는다. 이 통계량은 주어진 응용 분야에서 디스크 서비스를 제공할 때 각 저장 장치에 실질적으로 접근을 하는 평균 블록 수를 나타낸다. 따라서 RAID 레벨-5가 최상의 성능을 나타낼 수 있는 응용 분야를 예측하거나 최상의 성능을 나타낼 수 있도록 응용 분야에서의 스토리지 접근 패턴을 조정할 수 있다.

RAID 레벨-5의 동작을 위한 주요 자원으로는 데이터 캐시, 패리티 캐시, 저장 장치들이다. 이러한 자원들의 사용율(utilization)은 각 응용 분야에서의 스토리지 접근 패턴에 따라 각 자원의 구성을 변경하도록 하여 RAID 레벨-5가 최상의 성능을 나타낼 수 있도록 할 수 있다. 예를 들어, 특정 자원의 사용률이 일정 수준 이상으로 높다면 그 자원을 확장함으로써 그 자원이 성능 향상의 병목이 되는 것을 방지할 수 있다.

본 논문에서의 RAID 레벨-5는 앞서 기술한 것과

같이 DDR 메모리를 저장매체로 하는 저장 장치를 대상으로 하고 있다. 따라서 RAID 레벨-5의 기능을 수행함에 있어 저장 장치의 접근 횟수는 자기 디스크 또는 플래시 SSD보다는 스토리지 성능에 미치는 영향은 적다. 그것은 DDR 메모리가 매우 짧은 대칭적 접근 시간의 특성을 갖기 때문이다. 비록 접근 횟수가 성능에 큰 영향을 주지 않더라도 저장 장치 접근 횟수의 감소는 성능 향상에 기여할 수 있다. 따라서 시뮬레이션 결과 분석에 따라 응용 분야 또는 스토리지 접근 패턴에서 최소 접근에 이를 수 있도록 조정할 수 있을 것이다.

캐시 메모리의 적중률, 각 입출력 요구당 평균 서비스 시간, 각 저장 장치 접근당 평균 블록 수, 각 자원의 사용률 및 저장 장치 접근 회수는 밀접한 연관성을 가지고 있다. 따라서 시뮬레이션 결과는 각 통계량과의 연관성을 고려하여 분석되어야 한다.

#### IV. 시뮬레이터 구현

##### 4-1. 시뮬레이터

본 절에서는 앞서 기술한 시뮬레이션 모델에 따라 RAID 레벨-5를 시뮬레이션하기 위한 시뮬레이터의 구현에 대해 기술한다.

본 논문에서 시뮬레이터를 구현하기 위해 SMPL을 이용하였다. SMPL은 M.H.MacDougall이 이산 사

건(discrete event)을 기반으로 시뮬레이션 모델을 구현할 수 있도록 그의 저서에서 소개한 C 언어 함수들로 구성된 라이브러리이다[8].

시뮬레이터 구현을 위해 본 논문에서는 10개의 사건을 정의하였고, 각 사건간의 전이는 그림 3에 나타내었다. 그림 3에서 ‘REQARR’로 표시한 사건은 새로운 디스크 서비스 요구의 도착을 위한 것으로 확률 분포에 따라 도착을 한다. M/M/1 큐잉 이론에서는 정해진 시간동안 도착하는 수는 포아송 분포를 따르며, 각 도착들 간의 간격은 지수 분포를 따르는 것으로 알려져 있다. 본 논문에서도 평균 도착 시간을 이용한 지수 분포에 따라 새로운 디스크 서비스 요구를 생성하였다. 디스크 서비스 요구가 생성되면 정해진 비율에 따라 데이터 블록 읽기 또는 쓰기 요구인가에 따라 ‘RDSTRT’ 또는 ‘WRSTRT’ 사건으로 전이한다.

‘RDSTRT’ 사건은 읽기 요구를 처리하기 위한 것으로 우선 읽기 요구의 대상이 되는 모든 데이터 블록이 D-Cache에 적재되어 있는가를 검사한다. 만약 모든 블록이 D-Cache에 적재되어 있다면—d-cache hit으로 표시— D-Cache에 적재된 내용을 이용하여 디스크 서비스를 수행할 수 있다. 이때 D-Cache에 접근하기 위해 ‘DREQ’로 표시한 사건으로 전이한다. ‘DREQ’ 사건은 D-Cache에 접근하는 것을 시뮬레이션하기 때문에 정해진 시간이 지난 후 사용한 D-Cache를 해제하는 ‘DCOM’ 사건으로 전이한다.

만약 읽기 요구의 대상 데이터 블록들 가운데 전부 또는 일부 블록이 D-Cache에 적재되어 있지 않다면—d-cache miss로 표시— 저장 장치로부터 해당 블록들을 D-Cache에 적재한 후 서비스를 수행해야 한다. 이때, ‘DSKREQ’는 저장 장치에 접근하는 것을 시뮬레이션하는 것으로 저장 장치 접근 시간이 지난 후 ‘DSKCOM’ 사건으로 전이한다. ‘DSKCOM’ 사건은 사용한 저장 장치를 해제하는 것으로 저장 장치로부터 읽은 데이터 블록들을 D-Cache에 적재를 하였기 때문에 ‘DREQ’ 사건으로 전이를 하여 디스크 읽기 요청을 처리한다.

‘WRSTRT’ 사건은 쓰기 요구를 처리하기 위한 것으로 쓰기 요구의 대상 블록이 모두 D-Cache에 적재되어 있는가를 검사한다. 만약 대상 블록들 중 일부 블록이 D-Cache에 적재되어 있지 않다면—d-cache

miss로 표시— 새 블록을 적재할 D-Cache의 슬롯을 할당 받아야 한다. 이를 처리하는 사건은 그림 3에 ‘WRDCHK’로 표시되어 있다.

D-Cache에 적재 여부를 검사한 후에는 수정될 데이터 블록이 속한 패리티 그룹의 패리티 블록이 P-Cache에 적재되어 있는가를 검사한다. 만약 해당 패리티 블록이 P-Cache에 적재되어 있다면—p-cache hit으로 표시— 패리티 블록을 갱신하기 위해 P-Cache에 접근하는 사건인 ‘PREQ’와 접근 완료 사건인 ‘PCOM’로 전이한다.

P-Cache에 패리티 블록이 적재되어 있지 않다면—p-cache miss로 표시— 새로운 패리티 블록을 저장하기 위해 P-Cache의 슬롯을 할당하며, 사용 가능한 P-Cache 슬롯이 없다면 이들 중 한 개를 선정, 필요하다면 저장 장치에 저장함으로써 공간을 마련한다. 패리티 블록에 대한 처리가 완료된 후에 쓰기 대상이 되는 데이터 블록들을 D-Cache에 저장을 한다.

#### 4-2. 시뮬레이션 파라미터

시뮬레이터를 통해 앞서 기술할 각종 통계량을 획득하기 위해 본 논문에서는 표 1에 나타난 파라미터를 사용하였다.

표 1. 시뮬레이션 파라미터  
Table 1. Simulation Parameter

종류	파라미터	
단위시간	1	
저장 장치 수	4, 8, 16	
D-Cache 대비 P-Cache 용량	동일, D-Cache용량/저장장치수	
디스크 접근 요청 블록 수	16, 32, 64, 128	
read/write 비	0.5, 0.7, 0.9	
저장장치 접근 시간	10, 25, 50	
입출력 요청 시작 블록 번호 생성 분포	random	
	정규 분포 표준편차 비	20%,40%,60%

본 논문의 시뮬레이터에서 사용하는 단위 시간은 D-cache 또는 P-Cache의 블록에 접근하는 시간으로 설정하였다. 블록의 크기가 512 바이트일 때와 4 KByte일 때 실제 접근 시간을 달라지지만 RAID 레벨 5의 모든 동작이 블록 단위로 이루어지기 때문에 한

블록을 캐시에 저장하거나 읽는 시간을 단위 시간으로 사용하는 것은 문제가 되지 않는다.

RAID 레벨-5는 기본적으로 복수개의 저장 장치로 구성되며, 저장 장치의 수에 따라 그 성능 또는 특성이 달라질 수 있다. 따라서 본 논문에서는 저장 장치의 수를 4개, 8개, 16개로 하여 각각 시뮬레이션을 수행하였다.

P-Cache의 용량은 D-Cache 용량과 같거나 저장 장치의 수를 나눈 용량—저장 블록의 수—을 사용하였다. 이것은 D-Cache가 데이터 블록을 저장하지만 P-Cache는 패리티 블록만을 저장하기 때문에 D-Cache 만큼의 용량이 필요하지 않을 것이다.

디스크 접근 요청 블록 수는 입출력 요구의 대상 블록의 최대 수를 의미한다. 시뮬레이션을 할 때에는 이 파라미터를 최대값으로 하여 임의적으로 대상 블록의 수를 결정한다.

DDR 메모리 기반의 저장 매체는 기본적으로 read/write의 종류에 상관없이 동일한 접근 시간을 갖는다. 하지만 RAID 레벨-5에서는 read 요구와 write 요구를 구분하고, 별도로 처리하기 때문에 read/write 비는 평균 서비스 시간에 영향을 준다.

저장 장치의 접근 시간은 일반적으로 알려져 있는 디스크 접근 시간에 해당하며, 캐시 메모리의 접근 시간과 비교하면 매우 길다. 하지만 본 논문의 RAID 레벨-5는 저장 매체가 DRAM 메모리이기 때문에 상대적으로 짧은 접근 시간을 갖는다. 본 논문에서는 캐시 메모리 접근 시간 대비 10배, 25배, 50배의 접근 시간을 갖는다고 가정을 하였다.

입출력 요청 시작 블록 번호는 입출력 요청을 생성할 때 요청의 대상이 되는 블록의 수뿐만 아니라 시작 블록 번호도 생성한다. 이때 요청의 시작 블록 번호를 임의적으로 생성하거나 특정 분포를 사용하여 시작 번호를 결정할 수 있다. 특정 분포를 이용하는 것은 스토리지 시스템에 저장되어 있는 모든 데이터에 대해 균등하게 접근하기 보다는 특정 데이터에 대해 더 많은 접근이 이루어지는 것을 부분적으로 시뮬레이션할 수 있다. 본 논문에서는 임의의 시작 블록 번호를 결정하는 것과 더불어 정규 분포를 이용하여 시작 블록 번호를 생성하였다. 정규 분포를 이용할 경우에 밀집되는 정보를 표준편차를 이용하여 지정할 수 있다. 표준편차가 작으면 작을 수로 정규분포에 의해 생성되는 값은 평균값에 밀집되며, 표준편차가 크면 클수록 생성되는 값은 평균값에 대한 밀집

도가 떨어진다. 본 논문에서는 평균 디스크 접근 요청 블록 수에 표준 편차의 비를 곱한 값을 표준 편차로 사용을 하였다. 즉, 사용된 표준 편차의 비가 20%인 경우가 60%인 경우보다 요청 블록 시작 번호가 평균값에 밀집된다.

### V. 시뮬레이션 결과 및 분석

시뮬레이션은 앞서 기술한 파라미터 조합의 각각에 대해 5회씩 수행되었으며, 그 결과의 평균을 이용하였다.

#### 5-1. 캐시 적중률

본 논문의 RAID 레벨-5는 패리티 캐시와 데이터 캐시를 이용하고 있다. 캐시 적중률은 입출력의 대상 블록의 분포에 영향을 받는다. 입출력 대상 블록이 스토리지의 전체 논리 블록에 균등하게 분포되어 있다면 캐시 적중률은 낮을 것이다.

그림 4는 입출력 요구 대상 블록의 분포에 따른 캐시 적중률을 나타낸 것이다. 그림 4에서 ‘random’으로 표시한 것은 대상 블록의 시작 위치를 전체 논리 블록에서 임의로 결정하는 경우를 나타낸다. 그러나 많은 응용에서는 특정 블록들에 대한 접근이 빈번하게 발생하기 때문에 본 논문에서는 앞서 기술한 것과 같이 정규분포를 이용하여 블록 접근의 지역성을 시뮬레이션하였다.

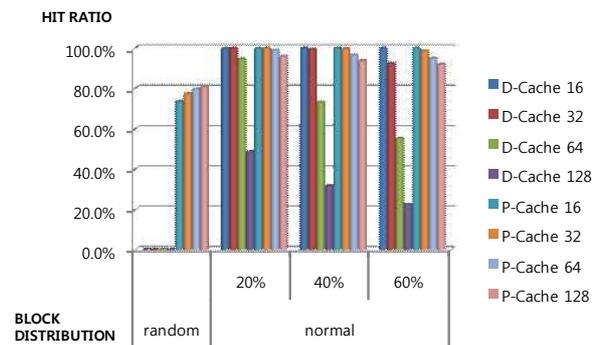


그림 4. 블록 분포에 따른 캐시 적중률  
Fig. 4. Cache Hit Ratio for Block Distribution

그림 4에서 ‘D-Cache x’로 표시한 것은 최대 블록의 크기가 x일 때 D-Cache의 캐시 적중률을 나타낸 것이며, ‘P-Cache x’로 표시한 것은 입출력 요구에 대상이 되는 블록의 시작 위치를 정규분포로 생성할 때 최대 블록 크기 x와 표준편차 비를 곱한 값을 표준편차로 사용했을 때의 캐시 적중률을 나타낸 것이다.

시뮬레이션 결과에 의하면 입출력 요구 대상 블록이 전체 논리 블록들 가운데 임의적으로 위치할 경우에는 읽기/쓰기 요청에 대해 해당 블록을 D-Cache에서 찾지 못해 그림 3에 표시한 ‘d-cache miss’가 발생하였다. 반면 P-Cache는 읽기/쓰기 요청의 비율에 따라 달라지지만 70%~80% 사이의 적중률을 나타내고 있다.

생성된 입출력 요구의 블록들이 지역성을 갖는 경우에는 높은 D-Cache와 P-Cache의 적중률을 나타내고 있다. 이 경우에 최대 블록의 크기가 큰 경우에는 작은 경우보다 상대적으로 낮은 적중률을 보이고 있는데, 최대 블록의 크기가 큰 경우에 해당 블록이 모두 D-Cache에 있을 확률이 상대적으로 낮기 때문이다. 반면 P-Cache의 적중률은 변화가 많지 않는데 그것은 데이터 블록과는 달리 패리티 블록은 패리티 그룹 당 하나씩 있으며, 이 패리티 블록들이 P-Cache에 이미 적재되어 있을 확률이 높기 때문이다. 이 경우도 최대 블록 수가 큰 경우가 작은 경우보다 패리티 블록 수가 많기 때문에 P-Cache에 해당 패리티 블록이 적재되어 있을 확률이 상대적으로 낮다.

5-2. 입출력 요구당 서비스 시간

그림 5는 하나의 입출력 요구에 대한 평균 디스크 접근 시간을 디스크 접근 시간에 대해 나타낸 것이다. 그림 5에서 ‘RW-x’로 표시한 것은 디스크 입력과 출력 요구의 비를 나타낸 것이다. ‘RW-0.5’는 디스크 블록 입력 요구와 출력 요구의 비가 50:50인 것을 나타내고 있으며, ‘RW-0.7’은 그 비가 70:30, ‘RW-0.9’는 그 비가 90:10인 것을 나타낸 것이다. 또 ‘ACCESS TIME’은 저장 장치의 접근 시간을 나타내고 있으며, ‘BLOCK DISTRIBUTION’ 은 입출력 요구의 대상 블록의 시작 위치에 대한 분포를 나타낸 것이며, 정규 분포는 표준편차의 비가 40%인 값을 표준 편차로 사용하였다.

읽기 요구보다는 쓰기 요구의 처리에 더 긴 처리 시간이 필요함에도 불구하고 시뮬레이션 결과에 의하면 읽기/쓰기 요구의 비율에 따른 서비스 시간의 변동에 큰 차이가 없다. 이러한 결과는 디스크 접근 시간이나 요구 대상 블록의 시작 위치의 분포에 상관 없이 일관된 결과를 나타내고 있으며, 이는 RAID 레벨-5가 쓰기 요구를 처리할 때 P-Cache를 사용함으로 디스크 접근의 수를 최소화하고 디스크 접근을 최대한 지연시켜 패리티 캐시가 P-Cache에 적재되어 있는 동안 더 많은 쓰기 요구를 처리하기 때문인 것으로 판단된다. 또한 P-Cache의 구조로 인해 패리티 블록의 생성을 위한 별도의 배타적 논리합 연산을 수행하지 않는 것도 읽기/쓰기 요구의 비에 상관없이 입출력 요구 당 평균 서비스 시간이 큰 차이가 없는 결과의 한 원인인 것으로 판단된다.

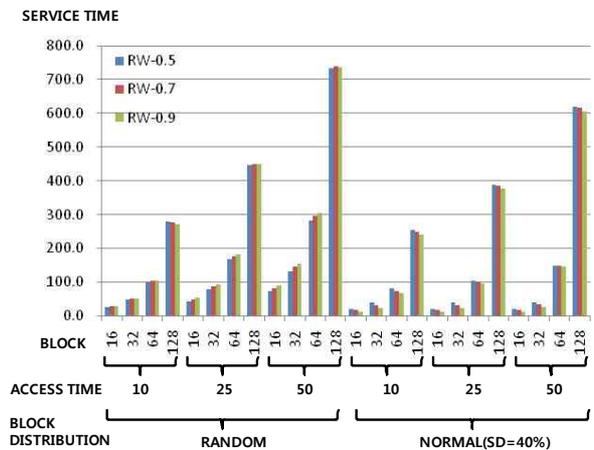


그림 5. 입출력 요구당 서비스 시간  
Fig. 5. Service Time per Disk IO Request

입출력 요구당 평균 서비스 시간도 요구의 대상이 되는 블록의 시작 위치에 대한 분포에 따라 다른 결과를 보이고 있다. 즉, 블록의 시작 위치가 임의적으로 결정되는 경우보다 지역성을 갖는 경우가 더 짧은 서비스 시간을 나타내고 있으며, 이것은 앞서 기술한 캐시 메모리의 적중률과도 관련이 있다.

캐시 메모리의 적중률이 높은 경우는 낮은 경우보다는 디스크 접근의 확률이 낮기 때문에 서비스 시간이 짧아지는 것으로 판단된다. 반면 캐시 메모리의 적중률이 높은 경우와 낮은 경우에 서비스 시간의 차이가 상대적으로 크지 않는 것은 RAID 레벨-5의 저

장 장치가 접근 시간은 매우 짧은 DDR 메모리를 기반으로 한 저장 장치이기 때문이다.

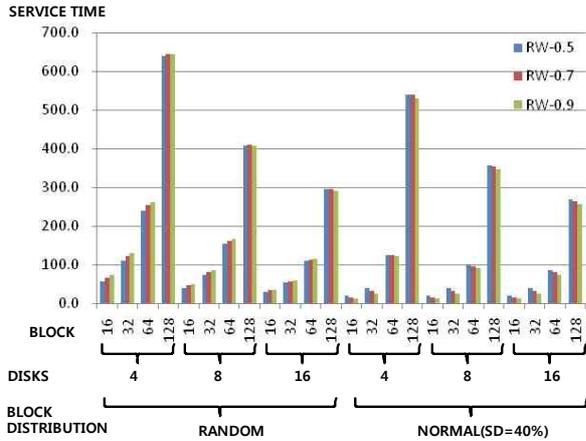


그림 6. 저장장치의 수에 따른 디스크 입출력 요구당 평균 서비스 시간  
Fig. 6. Average Service Time per Disk IO Request for the number of Storage Devices

그림 6은 디스크 접근 시간이 25인 경우에 저장 장치의 수에 따른 입출력 요구당 평균 서비스 시간을 나타내었다. 시뮬레이션 결과에 의하면 앞서 기술한 바와 같이 저장 장치의 수가 증가되면 평균 서비스 시간이 감소됨을 알 수 있다. 이것은 하나의 디스크 입출력 요구를 서비스하기 위해 많은 저장 장치가 동시에 블록 입출력을 서비스하기 때문이다. 이 경우에도 디스크 입출력 비의 변화에 대한 서비스 시간은 앞서 기술한 것과 같이 큰 변동이 없다.

5-3. 블록 당 서비스 시간

디스크 입출력 요구는 일반적으로 복수개의 블록을 대상으로 하고 있으며, 본 논문의 시뮬레이션에서는 최대값(16, 32, 64, 128)을 이용하여 블록 수를 임의로 생성하는 것으로 하였다. 따라서 디스크 입출력 요구 당 평균 서비스 시간에 대한 분석뿐만 아니라 블록 당 평균 서비스 시간을 분석하는 것도 의미가 있다.

그림 7은 블록 당 평균 서비스 시간은 나타낸 것이다. 시뮬레이션 결과에 의하면 D-Cache와 P-Cache 적중률이 높은 정규분포의 경우가 임의적으로 블록 시작 위치를 결정하는 경우보다 블록 당 평균 서비스

시간이 짧다.

또한 그림 7은 디스크 접근 시간이 10인 경우에 임의 분포인 경우에 평균적으로 디스크 접근 시간 대비 약 30~40% 시간만으로 서비스할 수 있으며, 정규분포의 경우에 약20~40%의 시간만으로 서비스할 수 있다. 반면 디스크 접근 시간이 25인 경우에는 각각 디스크 접근 시간 대비 20%~28%, 8%~24%의 시간만으로 서비스할 수 있다. 또 디스크 접근 시간이 50인 경우에는 각각 약 16%~22%, 4%~20%의 시간만으로 서비스할 수 있다. 이것은 디스크 접근 시간이 긴 저장 장치를 이용하는 경우에 본 연구의 RAID 레벨-5가 더 효과적임을 나타내는 것이다. 물론 절대적인 평균 서비스 시간은 디스크 접근 시간이 짧은 저장 장치를 이용할 경우에 더 짧다.

또 정규 분포와 같은 접근의 지역성을 가지며, 최대 블록의 수가 작은 경우(16, 32, 64)에는 디스크 접근 시간과 상관없이 비슷한 수준의 평균 접근 시간을 보였다.

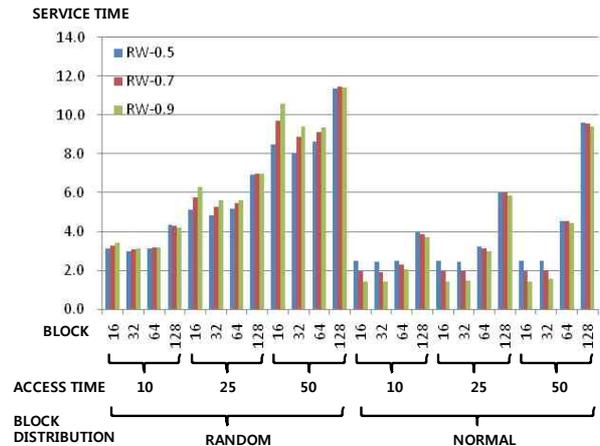


그림 7. 블록당 평균 서비스 시간  
Fig. 7. Average Service Time per Block

그림 8은 디스크 접근 시간을 25로 했을 때 저장 장치의 수에 따른 블록 당 평균 접근 시간을 나타낸 것이다. 서비스 대상 블록의 위치가 임의로 결정될 때에는 저장 장치의 수가 증가할수록 블록 당 평균 서비스 시간은 감소하지만 정규분포에 따른 접근의 지역성을 고려하면 최대 블록의 수가 작은 경우에 저장 장치의 수에 덜 영향을 받음을 알 수 있다.

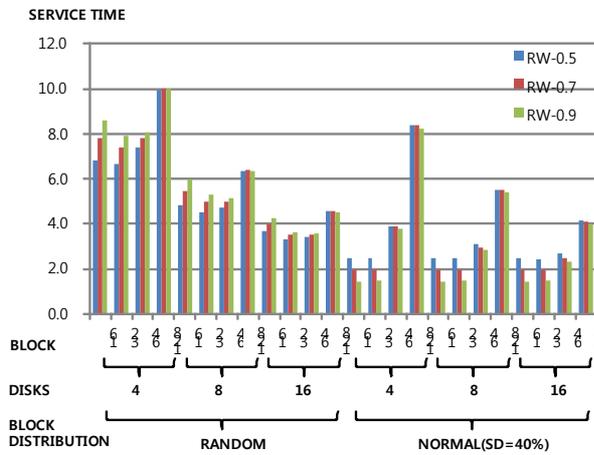


그림 8. 저장장치 수에 따른 블록 당 평균 서비스 시간  
Fig. 8. Average Service Time per Block for the number of Storage Devices

5-4. 자원 사용률

본 논문에서 개발한 시뮬레이터에서는 D-Cache, P-Cache, Disk(저장 장치)라는 자원(resource)을 정의하고, 시뮬레이션을 하는 동안에 각 자원의 사용률을 수집한다.

그림 9는 저장 장치가 8개, 접근 시간을 25로 했을 경우의 자원 사용률을 나타낸 것이다. D-Cache는 읽기/쓰기 비율에 상관없이 비슷한 사용률을 나타내지만, P-Cache는 쓰기 요구의 비율이 높은 경우에 더 높은 사용률을 나타낸다. 이것은 쓰기 요구가 많을 경우에 패리티 블록을 생성, 또는 갱신하는 빈도가 높아지면서 P-Cache의 사용률도 높아졌기 때문이다.

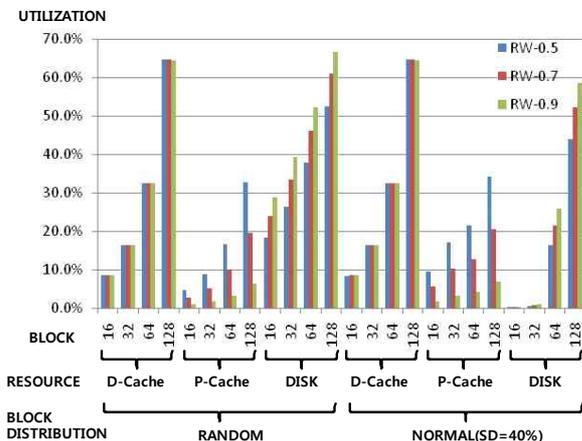


그림 9. 자원 사용률  
Fig. 9. Resource Utilization

Disk 자원의 사용률은 D-Cache의 적중률과 반비례하고 있다. 즉, D-Cache의 적중률이 낮은 경우 Disk 자원의 사용률은 높고, 적중률이 높은 경우에는 상대적으로 저장 장치 접근의 빈도가 낮아지기 때문이다. 또 읽기 요구의 비가 높을수록 Disk 자원의 사용률이 높아지고, 쓰기 요구의 비가 높을수록 Disk 자원의 사용률이 낮아지고 있다. 이것은 P-Cache를 이용한 쓰기 요구 처리가 효율적으로 이루어져 저장 장치에 대한 접근 빈도가 낮음을 나타낸다고 할 수 있다.

5-5. 디스크 입출력 요구 당 접근 빈도

그림 10은 디스크 입출력 요구 당 저장 장치에 대한 평균 접근 빈도를 나타낸 것이다. 이 지표는 저장 장치에 접근했을 때 입출력할 블록 수는 고려하지 않고, 단순히 저장 장치에 대한 접근의 수를 나타낸 것이다.

접근의 지역성을 고려하지 않은 경우가 정규 분포를 이용하여 지역성을 시뮬레이션한 경우보다 상대적으로 많은 저장 장치 접근 빈도를 보이고 있다. 모든 경우에 동일한 조건이라면 쓰기 비율이 높은 경우에 접근 빈도가 낮다. 이것은 앞서 기술한 것과 같이 P-Cache가 효율적으로 동작을 하며, 이로 인해 저장 장치 접근의 수를 감소시키는 것으로 판단된다.

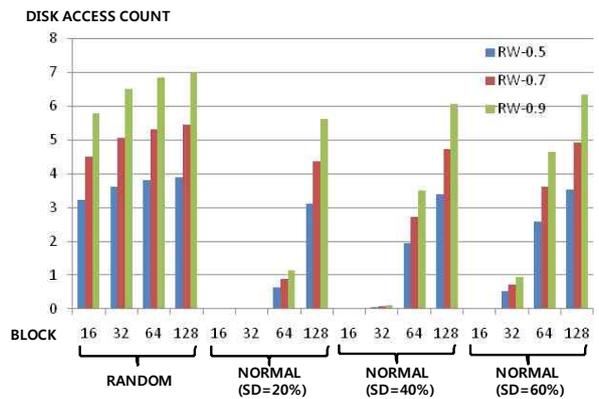


그림 10. 디스크 입출력 요구 당 평균 저장 장치 접근 빈도  
Fig. 10. Average Access Frequency per Disk IO Request

접근의 지역성을 고려할 경우에 블록의 최대 크기가 작으면 저장 장치 접근 빈도가 그림 10에 표시가

되지 않을 정도로 매우 낮다. 이것은 최초 해당 블록에 접근할 때 저장 장치에 접근을 하지만 그 블록이 D-Cache 또는 P-Cache에 적재되어 있는 상황에서 지속적으로 디스크 입출력 서비스를 하기 때문인 것으로 판단된다. 이 결과는 앞서 기술한 캐시 메모리의 적중률과 연관성이 있다.

## VI. 결 론

고성능, 고신뢰성 컴퓨팅 분야에서 성능 향상의 병목이 되는 요소는 스토리지 시스템이다. 스토리지 시스템의 성능 및 신뢰성 확보를 위한 여러 접근법들 중의 하나가 복수개의 저장 장치를 이용하여 접근의 병렬성을 높이며, 여분의 정보를 생성, 이용함으로써 신뢰성을 높이고자 하는 RAID 시스템이 개발, 사용되고 있다. 또한 반도체 메모리를 이용한 새로운 형태의 저장 장치에 대한 연구 개발도 이루어지고 있다.

본 논문에서는 DDR 메모리를 기반으로 한 저장 장치들로 구성된 P-Cache를 사용하는 RAID 레벨-5의 성능을 분석하였다. 이를 위해 본 논문에서는 P-Cache를 이용한 RAID 레벨-5의 동작 흐름을 정의하고 이를 기반으로 시뮬레이션 모델을 제안하였고, 제안된 모델에서 성능 및 특성 분석을 위해 수집할 통계량을 도출하였다.

시뮬레이션을 위해 본 논문에서 제안한 시뮬레이션 모델에 따라 시뮬레이터를 구현, 여러 파라미터를 이용하여 5회의 시뮬레이션을 수행하였다. 시뮬레이션의 결과에 따르면 P-Cache가 저장 장치로의 접근 빈도를 감소시키며, 디스크 입출력 요구에 대한 서비스 시간을 감소시키는 것에 효과가 있는 것으로 판단된다.

향후 연구 과제로는 P-Cache를 구현하여 실제 성능을 평가하고, 또 본 시뮬레이션 결과와 비교함으로써 더욱 정교한 시뮬레이터를 개발하는 것이다.

## 감사의 글

본 논문은 한국전자통신연구원 NGS시스템기술개발사업 위탁연구결과의 일부임.

## 참 고 문 헌

- [1] 정승국, 고대식, “차세대 스토리지 SSD 기술동향”, *정보통신연구진흥원, 주간기술동향* 1369호, pp.23-34, 2008. 10.
- [2] Ko Dae-Sik, Chung Seung-Kook, “A Design of DDR-1 Solid State Drive using PCI-e Interface,” *APCC 2009, accepted, Shanghai China, Oct. 2009*
- [3] Peter Chen, Edward Lee, Garth Gibson, Randy Kats, and David Patterson, R, “RAID : High-Performance, Reliable Secondary Storage,” *ACM Computing Surveys, Vol. 26, Issue 2*, Jun. 1994.
- [4] Pei Cao, Swee Boom Lim, Shivakumar Venkatarman, and Jone Wilkers, “The TickerTAIP Parallel RAID Architecture,” *ACM Tr. On Computer Systems, Vol. 12, No. 3*, pp.236-269, Aug. 1994
- [5] 김종현 역, 컴퓨터시스템 구조론, *사이텍미디어*, 2006
- [6] 구본근, 곽윤식, 정승국, 황정연, “DDR-SSD용 RAID 구현을 위한 RAID 구현 방법 및 저장 매체별 디스크 입출력 대역폭을 이용한 성능 평가”, *한국정보기술학회 논문지*, 제7권 제5호, 2009.
- [7] 구본근, 곽윤식, 정승국, 황정연, “DDR-SSD를 위한 RAID 레벨 5의 고속화 방법”, *한국향행학회 논문지*, 제13권 제5호, 2009.
- [8] <http://www.csee.usf.edu/~christen/tools/toolpage.html>

## 구 본 근 (具本根)



1991년 2월 : 인제대 전산학과(공학사)  
 1993년 2월 : 부산외국어대학교 컴퓨터공학과 (공학석사)  
 1998년 2월 : 경북대 컴퓨터공학과 (공학박사)  
 1998년 3월~현재 : 충주대학교 컴퓨터공학과 교수  
 관심분야 : 컴퓨터구조, 임베디드시스템, 상황인식시스템, 스토리지 시스템

## 곽 윤 식 (丁道令)



1984~6 경희대학교 공학사  
 1994. 3 경희대학교 공학박사  
 1991.5 ~현재 충주대학교 컴퓨터공학과 교수  
 관심분야 : 컴퓨터비전, 인터넷통신 등

### 정 승 국 (丁道令)



2004년 2월:한남대 전자정보통신공학과(박사)  
1985년 ~ 현재 : 한국전자통신연구원  
책임연구원  
관심분야 : 그리드 컴퓨팅, SSD, Utility  
Computing, Storage & Server Virtualization

### 황 정 연 (丁道令)



1993년 2월:중앙대 통계학과(이학석사)  
2006년 2월:충남대 통계학과(이학박사)  
1993년 6월~현재 : 한국전자통신연구원  
책임연구원  
관심분야 : 데이터마이닝, EA/ITA, DDR  
기반의 SSD 시스템, SRM S/W 개발