ISSN:1226-7244 (Print)
ISSN:2288-243X (Online)
논문번호 17-04-04
j.inst.Korean.electr.electron.eng.Vol.21,No.4,353~357,December 2017
http://dx.doi.org/10.7471/ikeee.2017.21.4.353

서버시스템에서의 메모리 불량현상 분석 및 해결방법

Analysis and solution of memory failure phenomenon in Server systems

신 현 승*, 유 승 주*★

Hyunsung Shin*, Sungjoo Yoo*★

Abstract

In order to maintain numerous server systems used in enterprise and data center environments, the most important thing is to prevent the occurrence of UE (Uncorrectable Error) of each server system. With the recent development of cloud services, more memory modules are being used than ever before, while the operating frequency of server systems has increased and the process of developing memory has continued to shrink, making it more likely to fail. In these environments, there is a way to repair memory defects directly in the server system, but there is no currently available guideline to use it effectively. In this paper, we propose a method to effectively prevent memory failure in a server system based on the observation and analysis of memory failure phenomenon in existing system.

요 약

엔터프라이즈 및 데이터센터환경에서 사용되는 수많은 서버시스템을 유지하기 위해서 가장 중요한 것은 각각의 서버시스템에서 UE(Uncorrectable Error)의 발생을 방지하는 것이다. 최근 클라우드 서비스의 발전으로 더 많은 용량의 메모리 모듈이 기존보다 더 많이 사용되고 있는 반면에 서버시스템의 동작 주파수는 높아지고 또한 메모리를 개발하기 위한 공정은 계속해서 축소되어 이전보다 불량이 발생될 확률이 매우 높아졌다. 이런 환경에서 서버시스템에서 직접 메모리 불량을 교체할 수 있는 방법이 제공되고 있지만 이를 효과적으로 사용할 수 있는 가이드라인이 현재 제공되지 않고 있다. 본 논문에서는 기존 시스템에서의 메모리 불량현상을 관찰하고 분석한 결과를 토대로 서버 시스템에서 효율적으로 메모리 불량을 방지하고 대처할 수 있는 방안을 제시하였다.

Key words : Datacenter, Server, Memory, Failure, Memory repair , UE(Uncorrectable Error)

Ⅰ. 서론

클라우드, 소셜네트워크 및 전자상거래 등 다양한 서비스의 발달로 예전보다 훨씬 많은 서버시스템이 사용되고 있으며, 이러한 서비스에서 발생하는 예상하지 못한 시스템 다운타임은 사업에 치명적인 손실을 발생시킨다. 현재 서버 한 개당 최대 24개의 메모리 모듈이 장착되고 2GHz 이상의 매우 높은 동작주파수를 사용하기 때문에 예전보다 불량이 발생될 확률이 매우 높아졌으며,

* Dept. of Computer Science Engineering, Seoul National University
★ Corresponding author
E-mail:sungjoo.yoo@gmail.com, Tel:82-2-880-1481
※ Acknowledgment
Manuscript received Dec. 8, 2017; revised Dec. 20, 2017 ; accepted Dec. 21, 2017

이러한 메모리 불량은 시스템의 다운타임을 발생시키는 가장 중요한 요소이다[1].

기존의 메모리셀 불량 교체는 메모리 제조사에서 생산단계에서만 가능하였으나, 최근에는 다운타임에 의한 재난을 막기 위해서 서버 시스템 제조사에서 새롭게 시스템 자체에서 운용 중에 메모리 불량 셀을 교체할 수 있는 방법이 제공되고 있다[2]. 하지만 메모리 내부의 교체할 수 있는 여분의 셀이 제한되어 있으며, 또한 실제로 시스템에서 셀을 교체 시에 예기치 못한 또 다른 불량이 발생될 소지가 있어서 현재 거의 사용되지 못하고 있다.

본 논문에서는 실제 서버시스템에서 장기간 메모리 불량을 수집하여 분석한 결과를 바탕으로 기존의 서버시스템에서 다운타임을 일으키는 UE(Uncorrectable Error)를 막기 위한 방법과 시스템에서 보다 효율적이고 정확하게 메모리 불량 셀을 교체할 수 있는 방법을 제안하고자 한다.

II. 실험 환경 및 구성

(1) 서버 메모리 구성

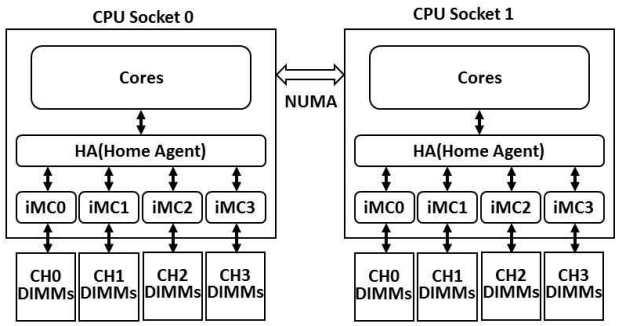


Fig. 1. Server Memory Configuration (INTEL E5 Xeon System)
그림 1. 서버 메모리 구성 (인텔 E5 제온 시스템)

서버시스템은 휴대용 시스템이나 다른 일반적인 시스템과 비교하여 많은 메모리를 사용한다.[3][4] 아래 그림 1은 일반적인 서버시스템의 구성을 나타낸 것으로 현재 가장 많이 사용되고 있는 인텔 E5 제온 시스템[3] 구성도이다. 두 개의 서로 다른 물리적인 CPU가 존재하며 각각 HA(Home Agent)라는 제어장치를 통하여 4개의 iMC(Integrated Memory Controller)에 연결된다. 그리고 하나의 iMC는 서버 메모리에서 하나의 채널(Channel)을 구성하고 각각의 독립적인 채널은 제어 신호와 데이터 신호를 공유하여 여러 개의 모듈이 장착될 수 있다.

그림 2에서와 같이 하나의 채널에는 일반적으로 2개 내지 3개의 메모리 슬롯(SLOT)이 있으며 각각 메모리 모듈인 DIMM(Dual Inline Memory Module)을 장착할 수 있다. DIMM은 내부에 1개, 2개 혹은 4개의 Rank라고 부르는 서로 다른 물리적인 그룹으로 나눌 수 있으며, 하나의 Rank는 독립적인 CS(Chip Select)가 존재하며 일반적인 서버시스템에서는 64비트 버스를 구성하기 위해서 4비트의 데이터버스를 가지는 16개의 메모리 칩(Chip)으로 하나의 Rank를 구성한다.

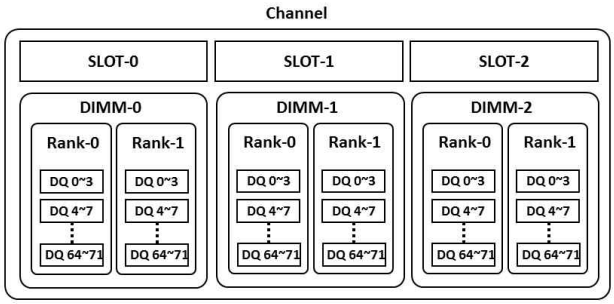


Fig. 2. Memory Channel Configuration (INTEL E5 Xeon System)
그림 2. 메모리 채널의 구성 (인텔 E5 제온 시스템)

또한 4비트의 데이터버스를 가지는 일반적인 메모리 x4 칩의 내부에는 인터리빙(Interleaving)을 효율화하기 위해서 서로 다른 4개의 Bank Group이 있으며, 각각의 Bank Group은 또다시 서로 다른 4개의 Bank로 구성된다. Bank 안에는 4개의 데이터 버스에 대해서 각각 DRAM의 워드라인(Word Line)에 해당하는 Row 주소가 있으며 BL과 대응되는 Column 주소가 존재한다.

(2) 메모리 FAULT, ERROR 그리고 FAILURE

인텔 제온 E5 CPU와 같은 범용 CPU에서는 기본적으로 ECC(Error Correction Code) 기능이 내장되어 있으며[5], 인텔 E5 CPU에는 SDDC(Single Device Data Correction)[6]라는 특별한 기능을 제공함으로써 4비트 버스를 가지는 단일 칩인 x4 메모리 칩이 완전히 망가져서 불량이 발생해도 이를 복구할 수 있다. 이렇게 CPU, 메모리 혹은 서버시스템에서 발생하는 복구가 가능한 불량 현상을

ERROR라고 부르며 불량의 원인에 관계없는 불량의 결과로써 불량에 의해 나타나는 현상을 말한다. 그리고 그러한 ERROR가 계속해서 쌓이거나 혹은 동시에 많이 발생하여 시스템에서 서비스를 할 수 없는 상태가 되면 FAILURE가 발생하게 된다. ERROR와 FAILURE는 불량에 의해 발생된 결과이며 실제 그 불량의 원인은 일반적으로 서버시스템에서 FAULT라고 정의한다.

(3) 불량 수집 데몬(Daemon) 개발

서버 시스템에서 메모리 불량을 수집하기 위해서는 하나의 불량이 발생했을 때 발생한 불량이 어떤 CPU 소켓인지, 어떤 메모리 채널인지, 어떤 메모리 슬롯(혹은 DIMM)인지 그리고 DIMM 내부에서는 어떤 Rank인지, 불량이 발생한 DQ에 해당하는 칩이 16개 중에서 어떤 것인지, 그리고 Bank Group, Bank, Row, Column 주소 등 많은 위치적인 정보와 함께 시간 정보를 제대로 수집해야 한다.

서버시스템은 주로 x86을 기반으로 동작하는 인텔(INTEL)[3] 혹은 AMD[4] 사의 CPU를 사용하고 있다. 그런데 이러한 CPU는 과거부터 현재까지 서버 시스템이 발전하여 세대가 거듭하면서 새로운 기능과 특성이 추가되고 변경되어서 서버 시스템의 CPU는 세대별로 서로 다른 하드웨어 구조와 서로 다른 다양한 레지스터들로 구성이 된다. 인텔 시스템의 경우 서버 CPU에 들어가는 CPU Cores가 점점 많아지고 또한 공정이 바뀌고 설계가 바뀔 때마다 새로운 CPU가 출시되고 이러한 CPU는 MSR(Model Specific Register)이라고 부르는 별도의 레지스터를 통해서만 제어가 가능하다.

메모리 불량을 수집하기 위해서 이러한 레지스터 정보가 필요하지만 실제 공개된 문서가 없기에 기존에 배포된 리눅스 배포 프로그램인 sb_edac[7]를 참조하여 데몬을 개발하였다. 하지만 sb_edac의 경우, 커널 모듈방식의 프로그램으로 메모리 불량을 따로 파일로 수집하기가 용이하지 않아서 기존의 또 다른 공개 프로그램인 mcelog[8]를 수정하여 개발하였다.

mcelog에서는 현재 리눅스 디바이스에 장착된 mcelog 장치를 이용하여 mcelog 장치의 입력 버퍼에 데이터가 수신되면 epoll 함수를 사용하여 멀티플렉싱 기법으로 소프트웨어 인터럽트를 발생하게 하여 데몬을 Trigger하는 방법을 사용한다. 이때 이렇게 Trigger된 mcelog 프로그램에 sb_edac에서 제공되는 불량을 해석하는 코드를 추가하여 PCI 방법으로 불량의 위치를 정확하게 알아낼 수 있도록 데몬 프로그램을 개발하였다.

(4) 실험에 사용한 서버 및 DIMM Spec

실험에 사용한 서버는 E5-2670 SandyBridge-EP 2.6GHz CPU를 사용하는 인텔서버이며, DIMM은 x8 4GB DDR3-1600을 사용하였다.

(5) 서비스 중인 시스템에 설치

서버시스템의 메모리 불량을 수집하기 위해서 인텔 제온(Xeon) E5-2640 v2 CPU[3]를 사용하는 서버 200대에 데몬을 설치하였다. 수집 기간은 약 400일 정도이며 수집기간 동안 워크로드로는 Hspice[9], TCAD[10] 등 반도체 설계 시뮬레이션 프로그램이 계속해서 동작하였다.

III 실험 결과

(1) CE(Correctable Error) 분석

수집된 메모리 불량 정보는 단편적인 불량 셀의 위치와 시간만 나타낼 뿐 실제 불량의 Fault를 정확히 분석하기가 매우 힘들다. 각각의 서버시스템은 각각 독립적인 불량 로그 파일을 가지고 로그 파일에는 두 개의 CPU 소켓에 속한 총 8개의 채널에서 발생하는 메모리 불량 정보를 가지고 있다. 메모리 불량은 채널과 독립적이며 채널과 연관된 불량은 메모리 불량이라기보다 시스템 버스의 불량과 관계된 것이므로 불량 메모리의 정확한 Fault를 분류하기 위해서 발생된 Error를 메모리 DIMM을 기준으로 표 1과 같이 분류하였다.

Table 1. Classification of Memory Fault on Server systems
표 1. 서버시스템에서 메모리 Fault의 분류

Fault Category	Definition
DIMM	Different Rank Errors
Rank	Same Rank Errors But Different Bank Errors
Bank	Same Bank Errors But Different Row/Col Errors
Row	Same Row Address Errors
Column	Same Column Address Errors
Single Bit	Single Bit Errors
Transient	Under 10ea Single Bit Errors

메모리 Fault를 정의하고 이를 기준으로 모든 서버시스템에서 발생된 불량로그를 모두 취합하여 Fault와 Error를 분석하였다. 이때의 Error는 Fault를 기준으로 실제 발생한 횟수를 나타낸 것이며 분류 결과는 아래의 그림 3과 같다.

실제 메모리 불량을 일으키는 원인인 Fault는 한 개의 메모리 셀에서만 불량이 발생한 Single Bit Fault가 28.4%이고 Error가 발생은 되었지만 10개 이하로 매우 드물게 발생한 Transient Fault가 54.1%로 대부분을 차지하였다. 하지만 실제 결과인 Error는 Single Bit Fault가 56.5%이고 DIMM Fault가 35.2%로 매우 높은 비율을 보였다.

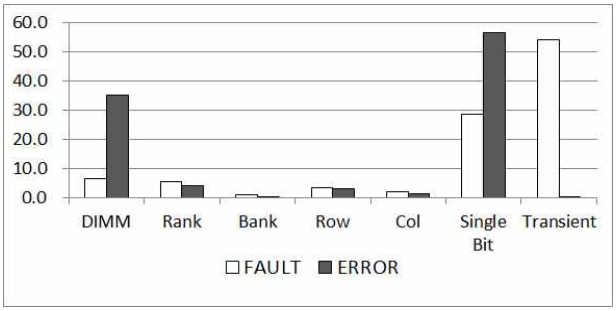


Fig. 3. Server Memory Fault Ratio and Error Ratio
그림 3. 서버 메모리의 Fault 비율과 Error 비율

(2) UE(Uncorrectable Error) 분석

CE(Correctable Error)와 별개로 시스템 다운을 발생시키는 UE의 로그 결과는 사실 실제로는 발생이 거의 되지 않아서 많은 수의 결과를 확보하기가 매우 힘들었다. 전체 서버 중에서 수집 기간 동안 총 5번의 시스템 다운이 발생하였으며, 그 중에서 2번의 경우 메모리 불량 로그가 전혀 없이 발생이 되었으며 나머지 3번의 경우 모두 Rank Fault에 의해서 UE가 발생되었다.

Ⅳ 결론

서버 시스템에서 메모리 불량을 발생시키는 원인은 크게 두 가지가 있다. 실험에서 관측된 DIMM Fault와 Rank Fault와 같이 제어신호 혹은 시스템 버스와 같은 인터페이스에서 발생되는 원인과 실제 메모리 단일 칩 내부에서 발생하는 셀 불량이 있다. 단순히 Error 로그만 가지고는 정확한 Fault의 원인은 알 수 없지만 일정한 시간 동안 축적된 로그들의 상대적인 위치적인 정보를 분석하면 실제 불량의 원인이 인터페이스 불량인지 아니면 메모리 내부의 셀 불량인지를 정확하게 알 수 있다. 이러한 분류에 의해서 서버 시스템에서는 다음과 같은 방법으로 서버시스템의 안정성을 높일 수 있다.

(1) 서버 시스템에서 효과적인 셀 교체(Repair)

DIMM Fault와 Rank Fault와 같은 인터페이스 불량에 의해서 발생되는 Error의 경우에는 Row 주소와 Column 주소에 상관없이 위치가 산발적으로 분산되어 계속해서 발생하기 때문에 만약 서버시스템에서 이를 고려하지 않고 셀을 교체하는 작업을 진행할 경우에는 여유분의 셀을 모두 소진하고도 계속해서 불량이 발생될 것이다. 따라서 일정한 시간 안에 서로 다른 주소의 메모리 불량이 발생될 경우에는 인터페이스 불량으로 간주하고 셀을 교체하는 작업을 진행하지 않고 DIMM 교체나 다른 방법으로 보다 효과적으로 불량을 처리할 수 있다. 그리고 또한 일정한 시간 안에 드물게 발생한 Transient Fault를 구별하여 불필요한 셀 교체를 방지함으로써 제한된 자원의 여유분의 셀을 보다 효과적으로 사용할 수 있다.

(2) UE(Uncorrectable Error)의 방지 방법

UE는 서버시스템에서 가장 영향이 큰 재난 사항으로 실험 결과에 의하면 실제 UE는 메모리 내부의 셀 불량의 의해서 발생되기 보다는 한 개의 메모리 칩의 인터페이스 불량에 의해서 발생된다. 따라서 만약 현재 발생된 메모리 불량이 총 발생된 개수는 적을지라도 위치적인 분석의 결과 Rank Fault인 경우에는 시스템 사용자에게 경고를 발생시키고 메모리를 교체할 수 있도록 조치를 취하면 UE의 발생을 줄일 수 있다.

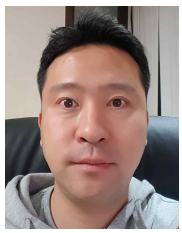
References

[1] Luiz André Barroso, Jimmy Clidaras and Urs Hölzle, "The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second edition," Morgan & Claypool publishers, pp. 154, Jul, 2013.
DOI:10.2200/S00516ED2V01Y201306CAC024)

[2] Charles Slayman, Manny Ma and Scott Lindley, "Impact of Error Correction Code and Dynamic Memory Reconfiguration on High-Reliability /Low-Cost Server Memory" in *Proc of the integrated Reliability Workshop Final Report, 2006 IEEE International,* 2006.
DOI: 10.1109/IRWS.2006.305243
[3] INTEL, "List of Intel Xeon microprocessors", https://en.wikipedia.org/wiki/List_of_Intel_Xeon_microprocessors
[4] AMD, "List of Intel AMD microprocessors", https://en.wikipedia.org/wiki/List_of_AMD_microprocessors
[5] Chin-Lung Su, Yi-Ting Yeh and Cheng-Wen Wu, "An Integrated ECC and Redundancy Repair Scheme for Memory Reliability Enhancement" in *Proc of the 2005 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*. 2005.
DOI: 10.1109/DFTVS.2005.18
[6] SDDC, "Intel x4 Single Device Data Correction", http://www.ece.umd.edu/courses/enee759h.S2003/references/29227401.pdf
[7] Torvalds, "Linux Kernel Drivers for Intel Sandy-Bridge Integrated MC", https://github.com/torvalds/linux/blob/master/drivers/edac/sb_edac.c
[8] Mcelog, "Advanced hardware error handling for x86 Linux", http://www.mcelog.org
[9] Hspice, "Device Level Circuit Simulation", https://www.synopsys.com/verification/ams-verification/circuit-simulation/hspice.html
[10] TCAD, "Technology Computer Aided Design", https://www.synopsys.com/silicon/tcad.html

BIOGRAPHY

HyunsungShin (Member)



2007 : BS degree in Electronic Engineering, Hanyang University.
2007~ : Research Engineer, Samsung Electronics.

SungjooYoo (Member)

1992 : BS degree in Electronic Engineering, Seoul National University.
1995 : MS degree in Electronic Engineering, Seoul National University.
2000 : PhD degree in Electronic Engineering, Seoul National University.
2000~2001: TIMA Research Engineer.
2001~2002: Engineer in Seoul National University.
2002~2004: TIMA Research Engineer.
2004~2008: Research Engineer, Samsung Electronics.
2008~2015: Professor in POSTECH.
2015~ : Professor in Seoul National University.