

주전산기 CPU 사용량 감소를 위한 개선 대상 프로그램 선정 모델에 관한 연구

신종민*, 민성기**

* 고려대학교 컴퓨터정보통신대학원 정보통신공학과

** 고려대학교 컴퓨터학과

e-mail : jmshin@dongbu.com*, sgmin@korea.ac.kr**

A Study on Model to Select Tuning Programs for Reducing CPU Usage

JongMin Shin* Sung-Gi Min**

* Computer Engineering of Computer Information and Technology, Korea University,

** Dept of Computer Science and Engineering, Korea University

요 약

부하를 분산 하여 시스템 자원의 효율성을 제고하여 시스템의 안정성을 향상하고자 하는 연구는 부하분산 에이전트 이용, 네트워크 트래픽의 동적 감시를 통한 부하 분산, 클러스트 환경을 이용한 부하분산, Queue 를 통한 트랜잭션을 제어에 의한 부하 분산 등이 있다. 또한 어플리케이션 부문은 프로그램 로직 단순화 및 SQL 최적화 등을 실시하여 부하를 경감하는 연구가 진행 되고 있다. 그러나 어플리케이션 부문의 튜닝을 실시하기 위한 대상 프로그램의 선정 및 튜닝 실시 후 사후 관리 방법에 대한 연구는 활발히 이루어 지지 않았다. 따라서 본 논문은 시스템 부하 개선을 위한 튜닝 대상 프로그램 선정 모델을 제시하고, 이를 실 업무에 적용하여 본 연구 모델의 실용성을 검증 하였다

1. 서론

IT 의 급격한 발달과 더불어 금융산업은 정보산업으로 대표 될 만큼 빠른 속도로 IT 기술을 적용 하고 있다. 금융기관에서는 멀티미디어와 네트워크 기술을 이용한 무인자동화지점(CD/ATM)을 확대 하고 있으며, 인터넷의 등장으로 고객들은 개인용 컴퓨터(PC)에서 금융기관 홈페이지에 접속하여 금융거래를 수행 할 수 있게 되었다. 뿐만 아니라 인터넷을 이용하여 각종 공과금이나 요금청구서를 확인하고 지불까지 한번에 할 수 있으며, 기업과 개인간(B2C), 기업과 기업간(B2B)의 전자적인 거래도 할 수 있게 되었다.[2]

이러한 환경은 금융업무가 정보기술의 지원 하에 24 * 365 서비스(365 일 24 시간 정보 시스템 가동 환경)를 가능하게 하고 있다.

이처럼 IT 와 밀접하게 접목된 업무환경에서의 시스템의 과부하에 의한 장애는 곧 기업의 비즈니스와 결부되어 심각한 금전적 손실이나, 기업의 이미지 실추에 영향을 미치게 된다. 따라서 시스템 부하를 최소화하여 시스템 안정성을 제고하고자 하는 많은 연구가 진행되고 있다.

시스템에 가해지는 부하는 네트워크 부하와 실 업무 시스템(이하 주전산기)에서 발생하는 부하가 있다. 이중 주전산기 시스템에서 발생하는 부하는 ①CPU 위주 부하, ②메모리 위주 부하, ③입출력 위주 부하로

나누어 진다. 이중 CPU 위주 부하를 유발하는 주요인은 사용량 폭주와 CPU 사용량 과다 프로그램(이하 Heavy-TR) 이다. [3]

사용자 증대 등으로 인한 사용량 폭주에 의한 과부하는 CPU 를 증설하여 해소하여야 하나, CPU 사용량 과다 프로그램으로 유발된 과부하는 그 원인이 되는 프로그램의 제거를 고려하여야 한다.

일부 연구에서는 프로그램 및 SQL 튜닝을 실시하여 Heavy-TR 을 제거하고 이로 인해 CPU 부하를 감소시킨 실제 성과를 내고 있다. 또한 IBM 기종을 사용하는 기업은 시스템 최적화 대안의 하나로 Heavy-TR 제거 권고를 받고 있다.[11]

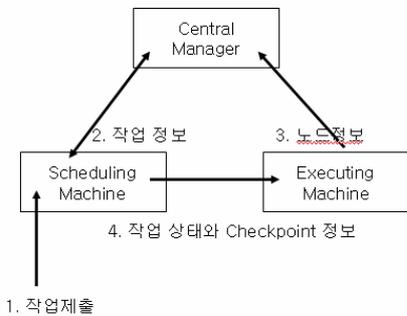
하지만 ‘CPU 를 과다하게 점유한다’는 기준이 없기 때문에 개선 대상인 Heavy-TR 을 선별하는 것이 쉬운 과제는 아니다. 따라서, 본 연구는 주전산기 CPU 를 과다하게 사용하는 Heavy-TR 을 선정하는 방법을 제시 하고자 한다.

논문의 구성은 제 2 장에서는 부하 분산에 관한 기존 연구 결과를 고찰 하고, 제 3 장에서는 기존 연구 결과를 적용한 환경에서 CPU 과부하 개선을 위한 개선 대상 트랜잭션 효과적인 선정 모델을 제시 하고, 제 4 장에서는 제시된 모델의 실 적용사례를 소개하고, 마지막으로 제 5 장에서 결론과 향후 연구방향에 대해 기술 하고자 한다.

2. 관련 연구 및 기존 개선 사례

한정된 시스템 자원을 효율적으로 사용하여 시스템을 안정적으로 유지하게 하는 연구는 지속적으로 진행되고 있다. 이중 대표적인 연구인 네트워크 부하 분산, 발생된 부하의 배분, 클러스터링, 프로그램 튜닝에 관해 고찰해 보고자 한다.

네트워크 부하에 관한 연구는 네트워크의 부하를 동적으로 모니터링 하여 과부하 링크의 트래픽을 동적으로 대안 경로를 찾아 분산 시키는 것이다. 이러한 동적인 트래픽 부하 분산으로 트래픽 관리를 위한 관리자의 부담을 덜어주고 실시간으로 변화하는 트래픽 과부하 상황을 동적으로 제어할 수 있도록 하였다. [10].



(그림 1) LoadLeveler Job cycle

발생된 부하의 배분에 관한 연구로, IBM의 경우 작업관리시스템(WLM: Workload Manager)을 이용하여 다수 사용자의 요구에 따라 자원을 효율적으로 배분 관리 하는 시스템을 제공 하고 있다. WLM은 사용자 프로그램을 실행하기 위해 명시한 요구자원을 시스템 자원으로부터 할당 받아 처리 한다. 사용자가 작업(Job)을 제출(Submit) 하면 작업은 일단 WLM 큐에 대기하다가 해당 작업의 처리조건이 맞는 클래스를 찾아 실행 한다(그림 1). 하지만 프로그램 실행 필요한 메모리의 정의에 따라 최소 20 배 이상 실행시간의 지연이 발생한다. [6].

클러스터링은 분산되어 있는 자원을 이용하여 부하 분산하고자 하는 연구는 이다. 현재 클러스터링에 관한 연구에서도 작업의 특성을 고려하여 CPU를 많이 사용하는 작업과 그렇지 않은 작업으로 분류하고 이를 통해 작업을 효율적으로 분배할 수 있는 알고리즘이 연구과제로 제시 되고 있고, 클러스터링 환경에서 성능을 최대한 활용하기 위해서는 소프트웨어의 최적화 필요성을 향후 과제로 제시하고 있다. (CPU 위주의 부하를 유발하는 트랜잭션에 대한 최적화 필요성에 대한 언급). CPU 위주 부하는 발생된 트랜잭션이 CPU로 하여금 많은 계산을 수행하게 하는 일을 말한다. CPU 위주 부하는 여러 개의 프로세스가 CPU로 하여금 계산을 수행하도록 연산 요청을 할 때만 들어진다.[3][8][9]

앞서 기술된 연구들은 주전산기 시스템 부하의 크

기와 상관 없는 부하 배분에 관한 연구이며, 이 외에도 발생하는 부하의 크기를 작게 하여 부하를 줄이고자 하는 연구로 프로그램 튜닝에 관한 연구가 있다[7]. SAP ERP 트랜잭션 튜닝의 경우 실행 시간을 단축(CPU 사용량 절감)시키기 위해 SQL 튜닝 등에 의한 전반적인 점검과 개선을 실시하고 있다[7]. 하지만 개선 대상 트랜잭션의 선정 방안은 제시 하지 못하고 있다.

일부 기업에서는 Heavy TR을 프로그램 튜닝을 통하여 개선 하고 있다. L사 및 D사의 트랜잭션의 모니터링을 통하여 주단위로 CPU 사용량이 가장 큰 트랜잭션 5~10 개를 선정 하여 로직 및 SQL 튜닝을 실시 하고 있다.

3. 제안 모델

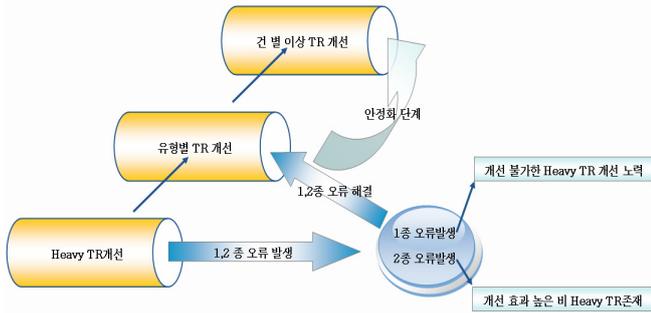
본 연구 논문은 주전산기 시스템에서 수행되는 트랜잭션의 크기(CPU 사용량)가 동일하지 않기 때문에 발생하는 과부하 문제를 해결하기 위하여, 과부하의 원인이 되는 Heavy-TR을 효과적으로 선별하여 제거할 수 있도록 개선 모델을 제시 하고자 한다. 실 운영 환경에서 조직의 학습 곡선 및 성숙도 등을 고려할 때 효율적인 개선 방안으로 아래와 같이 3 단계 모델을 제시 하고자 한다.<그림 4>

단계	수행 내역	개선 대상 선정 범위
1단계	전체 트랜잭션 대상 산포 개선 - 발생한 모든 트랜잭션에서 개선 대상 건을 선정 한다	Heavy TR기준점 (3초) 개선대상 (21%)
2단계	트랜잭션 유형별 산포 개선 - 1 단계 완료 후 시행. - 트랜잭션을 유형으로 분류 후 유형별 개선 대상 건을 선정 한다	2단계 개선대상 1단계 개선대상
3단계	개별 트랜잭션 산포 개선 - 2 단계 완료 후 시행. - 트랜잭션 건 별로 대상 건을 선정 한다.	개선대상 범위 비대상 TR 개선대상 TR

(그림 4) 개선대상 프로그램 선정 모델

1 단계는 개선대상 트랜잭션을 처음 선정하는 단계로 조직의 경험이 전무하기 때문에 단순한 선정방법을 사용한다. 즉 전체 트랜잭션을 대상으로 CPU 사용량 상위 20%(파레토 기법 활용)에 해당하는 트랜잭션을 개선대상(Heavy-TR)으로 선정한다.

1 단계의 경우 개선대상 트랜잭션을 제거할 경우 개선 효과가 크고 매우 빠르게 개선이 진행 된다. 하지만 6 시그마의 관점에서 보면, 1종 오류(양품을 불량으로 판정하는 오류: 개선이 불가능한 트랜잭션을 개선 대상으로 선정한 오류) 및 2종 오류(불량을 양품으로 판정하는 오류: 개선이 필요한 트랜잭션을 개선 대상으로 선정하지 못 하는 오류)를 내포하고 있다.(그림 5).



(그림 5) 1,2 중 오류 발생 유형

조직이 1 단계에서 선정된 대상을 개선하면서 노후가 축적되어 1,2 중 오류를 제거하는 정교한 개선 대상의 선정을 요구하게 된다.

2 단계는 트랜잭션 유형별 산포 개선 단계로 1 단계의 선정 오류를 제거하기 위하여 프로그램을 일정한 유형으로 나누고 각 유형별로 파레토 기법을 적용하여 개선대상을 선정 한다.

프로그램의 유형을 나누는 방법은 각 사의 상황에 따라 다르지만 D 보험사의 사례를 보면 프로그램 복잡도, 사용 Table 개수, 프로그램 종류(단순조회, 다건조회, 복잡 조회, 단순 입력, 다건 입력, 복잡 입력, 단순 출력, 다건 출력, 복잡 출력 등) 등을 고려하여 분류하고 있다.

2 단계를 진행 하며, 1 단계에서 발생한 1,2 중 오류가 보완되어 정교한 개선 대상 트랜잭션이 선정된다. 1 단계의 1 종 오류로 인해 선정 되었던 트랜잭션은 2 단계의 개선 대상 건에서는 제외 되고, 2 중 오류로 인해 未 선정 되었던 건은 2 단계에서는 개선 대상에 추가 선정 된다. 즉 1 단계 보다 좀더 정교한 개선대상을 선정하게 된다.

3 단계는 개별 트랜잭션 산포 개선 단계로 산포를 트랜잭션 별로 측정하여 사전에 설정된 선정기준에 부합되면 해당 트랜잭션을 개선 대상으로 선정한다.

선정기준: '기준값 * 8' 이 산포 최대값 보다 클 경우
 - 기준값 * 8 은 D 사의 경험치 임. 각 사의 정책에 따라 다르게 적용 할 수 있음.
 기준값 = 산포의 평균값 - 산포 최소값

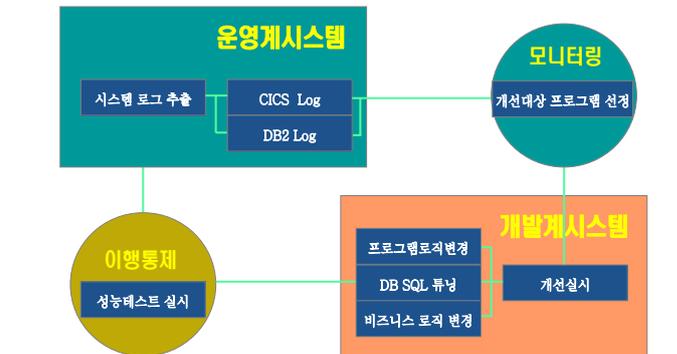
4. 제안 모델의 검증

본 연구 모델의 특징은 트랜잭션 별 CPU 사용량 산포를 분석 하여 개선대상으로 선정하는 것이다. CPU 사용량 평균에 의한 대상 선정 방법은 개별 트랜잭션의 이상현상(6 시그마의 관점의 이상 원인, 예를 들면 과도한 조회기간 및 비정상 조회조건 입력 등 일상 업무처리 방법에서 벗어난 행위로 발생 되는

현상)을 보이는 트랜잭션을 개선 대상으로 선정 할 수 없다. 따라서 선정된 개선 대상은 업무처리 로직이 복잡하거나, 데이터베이스의 I/O 가 많은 트랜잭션이 주로 대상이 된다.

그러나 CPU 사용량 평균이 아닌 트랜잭션 건 별 산포를 개선대상 선정 방법으로 하면 이상현상을 발생시키는 트랜잭션(Heavy TR)을 선별하여 개선대상으로 선정 할 수 있게 된다.

본 논문의 연구 모델을 증명하기 위해 6 시그마 기법인 DMAIC(Define, Measure, Analyze, Improve, Control) 방법론을 적용하여[3] 국내 D 손보사에서 2007 년 3 월부터 12 월까지 프로젝트를 진행 하였다. <그림 6>은 6 시그마 Improve 단계에서 도출된 실행 방안으로 연구 모델 1,2,3 단계의 개선 대상을 선정 하기 위한 프로세스이다. 모든 트랜잭션의 CPU 사용량을 모니터링 하여 매일 개선 대상이 선정된다. 6 시그마 프로젝트로 수립된 관리 방안은 파일럿 적용 후, 운영 업무의 정규 프로세스로 정착 되어 시행되고 있다.



[개선대상 프로그램 리스트]

NO	일자	실행시간	프로그램명	프로그램 유형	소요시간			대상 선정 사유
					CICS	DB2	합계	

(그림 6) 상시 모니터링 프로세스

제안 모델의 1,2 단계 개선대상 트랜잭션의 선정 결과는 <표 2>와 같다. 제안 모델의 1 단계인 전체 트랜잭션을 대상으로 선정기준 'CPU 사용량 3 초 이상인 트랜잭션'을 적용하여 225 건이 선정 되었다. 2 단계인 유형별 분류에 의한 개선을 적용하여 22 건을 추가로 도출 하였다. 즉, 1 단계 개선대상 선정 건수보다 2 단계를 추가 수행 함으로 개선대상 건수를 증대 할 수 있다.

$$\Sigma X_1 \leq \Sigma X_1 + \Sigma X_2 + \Sigma X_3 \quad \text{---식 1)}$$

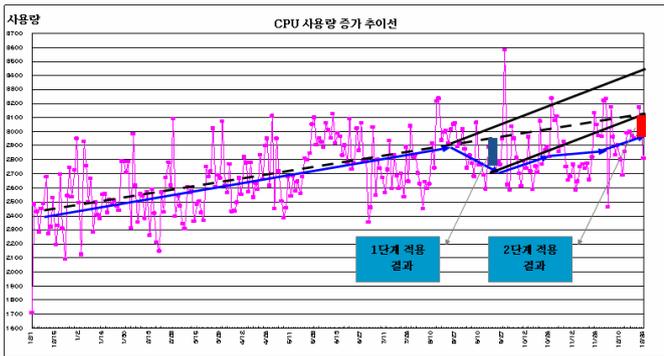
(X₁: 1 단계 건수, X₂: 2 단계 건수, X₃: 3 단계 건수)

<표 2> 선정 기준에 따른 개선대상 선정 결과

구분	선정 기준	선정결과
1 단계	CPU 사용량을 3초 이상 점유하는 트랜잭션	225 건
2 단계	각 유형별 CPU 점유량 상위 5%인 트랜잭션	22 건

1 단계 적용은, 전체 트랜잭션을 모니터링 하여 CPU 3 초 이상(1 초: CPU 사용량 약 40 MIPS) 점유 건을 개선 대상으로 선정 하였다. 3 개월 동안 트랜잭션을 지속 개선한 결과 3 초 이상 트랜잭션이 일 평균 371 건에서 92 건으로 약 250 % 감소 하였다. 개선 트랜잭션의 일 평균 CPU 점유 비를 비교하면 개선 전 대비 74 %의 개선 효과가 있음을 알 수 있다. (개선 전 1,441 초 → 개선 후 375 초).

2 단계를 진행하기 위해 트랜잭션을 기능적인 측면인 DBMS 처리기능(조회, 입력, 삭제, 갱신), 화면 처리기능(조회, 입력, 출력)을 고려하여 최종(3 X 3) 9 개 유형으로 분류 하였다. 또한 트랜잭션 대상은 전체 발생 하는 3,500 종의 트랜잭션 중 파레토의 법칙을 적용하여 사용량이 많은 상위 700 건(상위 20 %)을 프로그램 유형 분류의 대상으로 하였다. 2 단계 적용은, 1 단계 적용 후 감소된 Heavy TR 의 추가 발생이 억제 되었으며, 발생된 Heavy TR 의 존재 기간도 짧아졌다. (그림 7)은 1,2 단계 적용 개선으로 주전산기의 CPU 의 자연 증가추이를 증가 추이선과 실제 사용량을 측정하여 실제로 완화 시키는 효과를 증명 하였다.



(그림 7) 적용 결과

D 사에서는 1,2 단계의 실행 방안으로 상시 모니터링 체계를 구축하여 개선 대상을 선정 하고 각 담당자에게 해당 트랜잭션을 개선 하도록 하였다. 또한, 3 단계는 1,2 단계의 성숙 모델로 6 시그마 Control 단계로 진행 된다. 각 업무 담당자가 상시 모니터링을 수행하여 개선대상을 선정하고, 개선을 수행 하게 된다. (본 논문에서는 실적에서 제외 함)

5. 결론

본 논문은 “주전산기 CPU 사용량 감소를 위한 개선 대상 프로그램 선정 모델에 관한 연구”로 튜닝 대상 프로그램 선정을 위한 3 단계 모델 제시하였고, 이를 실 업무에 적용하여 실용성을 입증 하였다..

본 논문에서 제시한 모델은 주전산기 부하 분산과 관련된 기존 논문에서 언급되지 않은 CPU 자체 부하를 경감 할 수 있도록 튜닝 대상이 되는 프로그램 선정에 대한 구체적인 모델이다.

향후 연구로는 2 단계의 트랜잭션의 유형 분류에 대한 표준 수립에 대한 연구를 진행하고자 한다. 또

한 본 연구 모델의 3 단계는 각 Process Owner 별로 개선이 진행되어 모델의 검증이 미흡했다. 향후 각 단계의 결과를 추적 관리 하여 본 논문에서 제시한 개선 모델에 대한 철저한 검증과 트랜잭션 유형에 대한 표준 수립을 위한 연구를 지속 하겠다.

참고문헌

- [1] 김이영, “전자고지. 결제(EBPP)서비스의 현황 및 전망”, 정보통신정책 제 13 권, 제 18 호,2001, 10.
- [2] 신미향,김갑수,서영희[2003]. “웹 기반환경하에서의 종합금융 시스템에 관한 사례연구”.동명대학
- [3] 김기욱, 정희진, 이상호.”데이터베이스 시스템 벤치마크를 위한 부하 생성기 설계”, 2004 한국정보처리학회 춘계학술발표대회
- [4] 김한샘,한혁수,”CMMI 기반의 프로세스 개선을 위한 6 시그마 활용사례”,2006 상명대학교
- [5] 이해영,최호진,백종문,”6 시그마 현황과 소프트웨어 6 시그마 향후전망”, 2005.12 정보과학회지 제 23 권
- [6] 이영주,우준,성진우,이진훈,임경빈,박찬열,”IBM 시스템에서 WLM 을 이용한 LoadLever 최적 환경 구현”, 2007 년 한국정보처리학회 춘계 학술대회
- [7] 이정재,송관호,최진영,”튜닝을 통한 User Response Time 감소 기법”,2007 한국정보처리학회 춘계 학술대회
- [8] 김호중,”클러스터 시스템의 성능 측정”2003.7.28 한국과학기술원
- [9] 박지은,”부하 분산시스템의 확장”,2004.8.9 한국과학기술원
- [10] 김희진, 박상근, 최덕재,” 네트워크 트래픽 감시를 통한 동적인 트래픽 부하 분산에 관한 연구”. 1999 전남대학교
- [11] www.ibm.com