

성능지수를 고려한 관계형 데이터베이스 시스템의 목표지향형 성능관리 모델

박상용*, 김정동**, 백두권**

*고려대학교 소프트웨어공학과

**고려대학교 컴퓨터·전파통신공학과

e-mail: *roseflavor@paran.com, **kjdvhu@gmail.com, **baikdk@korea.ac.kr

A Goal-Oriented Performance Management Model with DB Performance Index for Relational Database System

SangYong Park*, Jeong-Dong Kim**, Doo-Kwon Baik**

*Dept, of Software Engineering, Korea University

**Dept, of Computer Science & Radio Communications Engineering, Korea University

요 약

관계형 데이터베이스를 사용하는 응용시스템들은 어플리케이션 소스 코드에 내재된 SQL을 통하여 데이터를 액세스하게 되며, SQL을 통한 데이터베이스 액세스 성능은 응용시스템의 성능에 결정적인 영향을 미치기 때문에, 관계형 데이터베이스시스템의 튜닝에 대해서는 많은 연구가 이루어져 왔다. 그러나 데이터베이스시스템 튜닝을 통해 얻은 성능 수준은 응용시스템의 변경이나 데이터 저장량 증가 등 여러 요인에 의해 다시 저하될 수 있으며, 이와 같은 성능 수준의 변화를 정량적으로 관리하면서 최소 비용으로 성능 수준을 유지할 수 있도록 튜닝 시점을 결정하는 방법에 대한 연구는 현재까지 연구된 바가 없다. 이 논문에서 제시하는 DB 성능지수와 이를 이용한 데이터베이스시스템의 성능관리 모델은 특정 벤더에 종속되지 않고 성능 수준을 정량화함으로써 이해관계자들에게 성능 수준의 변화를 가시적으로 보이고 데이터베이스 관리자의 성능관리 및 튜닝 시점 판단을 지원하여 일정 수준의 성능을 유지할 수 있는 방법으로 활용될 수 있다.

1. 서론

1970년 E. F. Codd에 의해 관계형 데이터베이스가 소개된 이후로 다양한 벤더에 의해 상품화되고 보급되어져 현재는 기업에서 사용하고 있는 대부분의 데이터베이스가 관계형 데이터베이스관리시스템을 기반으로 하고 있다[1].

관계형 데이터베이스를 사용하는 응용시스템들은 어플리케이션 소스 코드에 내재된 SQL을 통하여 데이터를 액세스하는데, 이 SQL의 성능이 응용 애플리케이션의 성능을 좌우하게 되고, 이것이 모여서 전체 응용시스템의 성능을 결정하기 때문에, 관계형 데이터베이스시스템 기반의 응용시스템을 구축한 대부분의 사이트들은 SQL 성능을 향상시키기 위해 많은 시간과 비용을 투입하고 있다. 그러나 데이터베이스 튜닝을 통해 얻은 성능 수준은 응용시스템의 변경이나 데이터 저장량 증가와 같은 환경적 변화에 의해 다시 저하될 수 있어 데이터베이스 튜닝은 일회성의 활동이라 할 수 있으며, 이와 같은 환경적 변화에 대한 대응은 SQL 몇 개의 개선이 아닌 더 넓은 범위에 걸친 전반적인 개선이 요구된다.

기업에서의 데이터베이스시스템 성능관리란 일회성의 튜닝 외에도 반복적인 모니터링을 통해서도 이루어지는데[2], 이러한 방법은 데이터베이스시스템에 문제가 발생하고 있는지의 여부는 감지할 수 있지만, 시스템 전반의 성능을 평가하여 객관적이고 가시적인 수단으로 이해관계자 간에 의사소통을 원활하게 하거나 데이터베이스 시스템 전반에 대한 재튜닝 여부를 판단하는 데에는 적합하지 않다. 현재 성능관리와 관련

한 연구는 데이터베이스 벤더들이 제시하는 방법을 기반으로 하는 튜닝[3]이나 성능관리를 위한 서비스 관점에서의 접근[4], 성능평가[5] 등이 주를 이루고 있기 때문에, 보다 적은 비용으로 데이터베이스시스템의 성능 수준을 유지하고, 객관적인 수단을 통하여 체계적·지속적으로 성능관리를 할 수 있는 방법에 대한 연구가 필요하게 되었다. 이 논문에서는 기존의 문제를 해결하기 위해 DB 성능지수라는 새로운 분석지표를 도입하여 이를 통한 성능목표 설정 및 성능관리에 대한 방법을 제안하고 있으며, 논문의 구성은 다음과 같다.

제 2장에서는 성능관리와 관련한 기존의 연구 방법에 대해 기술하고, 제 3장에서는 제안 모델인 성능지수를 이용한 성능관리 접근방법에 대해 정의한다. 제 4장에서는 제안 모델의 효과에 대해 검증하고, 제 5장에서는 결론 및 향후 연구에 대해 제시한다.

2. 관련 연구

현재 데이터베이스시스템의 성능관리와 관련한 연구는 데이터베이스 벤더들이 제시하는 방법에 기초한 DB튜닝[3], 서비스와 프로세스 관점에서 성능관리를 접근하는 ITSM[4], 벤치마크 기반의 성능평가[5] 등이 주를 이룬다.

DB튜닝(Performance Tuning)은 오라클이나 마이크로소프트, IBM 등 DBMS 벤더들이 많은 방법들을 정형화하여 제공하고 있으며[6], 데이터베이스 튜닝 전문 업체에서 제시하는 방법과 온-오프라인 상에서의 많은 저술들이 활용되고 있다. DB튜닝은

메모리 사용 효율화, DBMS 파라미터 변경, SQL 재구성, 힌트 사용 등 다양한 접근을 통해 SQL의 응답속도를 향상시키도록 하고 있으며[7], 주로 시스템 안정화나 Trouble Shooting 측면에서의 해결책으로 접근하고 있다. DB튜닝은 데이터베이스시스템의 응답속도 향상이라는 목적은 달성할 수 있으나 일회성이고, 일단 향상된 성능을 유지·관리하는 방법에 대한 제시는 없다.

ITSM(IT Service Management)은 정보시스템 사용자가 만족할 수 있는 서비스를 제공하고 지속적인 관리를 통해 서비스의 품질을 유지, 증진시키기 위한 일련의 활동·기법이다. 고객과 서비스 제공자 간 계약인 서비스 수준 관리(SLA: Service Level Agreement)의 품질 수준을 만족시키기 위한 모든 기법을 지칭하는 것으로, SLA의 내용에는 서비스 수준을 측정할 수 있는 세부 서비스 요소(SLO: Service Level Object)들이 포함된다. 이러한 IT 서비스 관리를 제공하기 위한 모든 과정 및 프로세스를 정리해 놓은 것이 ITIL(IT Infrastructure Library)이다[4]. ITSM은 서비스 관점의 접근에 성능 모니터링(Performance Monitoring) 기법을 내재하여 성능 문제에 대한 감시와 원인 규명에 활용 가능하지만 결과적으로 좁은 범위에서의 문제 해결 중심이고, 데이터량 증가와 같은 환경적 변화에 대응하거나 시스템 전반의 성능 수준 목표를 가지화하여 관리하는 구체적인 방법에 대한 제시는 없다.

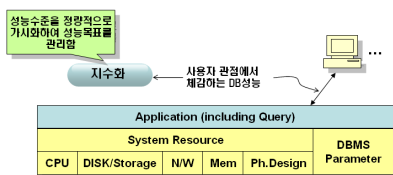
성능평가(Performance Evaluation)에 대한 연구는 주로 DBMS 자체의 성능에 대해 평가하는 연구가 주를 이루고 있으며, Wisconsin 벤치마크나 TPC 계열의 벤치마크가 널리 사용된다. 벤치마크를 통해서 질의최적화의 성능이나 연산 기능, 트랜잭션 처리 능력 등 DBMS 자체의 기능과 성능을 평가하는 것이 목적이므로, 복잡·다양한 질의를 많이 사용하며, 동일 업종이라도 서로 다른 비즈니스를 구현하는 비즈니스 애플리케이션들로 구성된 응용시스템의 성능을 가지화하여 평가하기에는 적합하지 않다.

그 외에도 데이터베이스 활용 관점에서 성능평가나 튜닝 등을 적용하는 연구가 일부 있으나 일회성의 방법적 접근을 연구한 것들이고, 성능수준을 가지화·객관화하여 지속적으로 관리할 수 있는 방법적인 부분에 대한 연구는 이루어지지 않고 있다.

3. 제안 모델

3.1 성능지수

데이터베이스시스템의 성능은 SQL의 성능으로 나타나며, 여기에 영향을 미치는 요소는 CPU, 디스크/스토리지 구성, 네트워크, 메모리, 데이터베이스 설계, DBMS 파라미터, SQL 사용형태 등 다양한 팩터들이 존재하지만 결국 사용자가 체감하는 성능은 <그림 1>에서와 같이 SQL을 포함한 사용자 접점에 있는 애플리케이션으로 집약된다.



<그림 1> 사용자 체감 응답속도에 대한 영향 요소

즉, 여러 요소들이 복합적으로 작용하여 SQL의 성능을 결정하게

되고, SQL이 내재된 애플리케이션은 SQL에 의해 성능이 결정된다 할 수 있기 때문에, 결국 SQL을 통한 데이터베이스 액세스 성능을 수치화함으로써 시스템의 성능 수준을 가지화할 수 있다.

성능지수는 성능 목표에 대한 성능 측정값의 구성비로 정의할 수 있으며, 수식 1과 같이 산출한다.

$$\text{성능지수} = \frac{\text{성능 측정값}}{\text{성능 목표값}} \times \text{가중치} \dots\dots\dots (1)$$

수식 1에서 사용된 가중치는 성능을 측정하는 성능지표 항목의 튜닝 우선순위에 따른 구성비로, 성능지표 항목별로 튜닝의 효과성을 고려하여 결정하며, 응용시스템의 목적, 용도, 시스템 구성 등 여러 요소가 반영된다.

성능지표와 가중치가 결정되면, <표 1>과 <그림 2>에서 보는 바와 같은 성능평가 모형에 대입하여 성능지표별로 성능지수를 산출하고 수식 2에 따라 총성능지수를 계산할 수 있다.

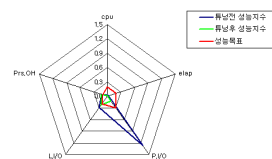
총성능지수는 각 지표항목에 대한 성능지수를 토대로 하여 시스템의 성능을 하나의 수치로 단일화한 것이며, 성능목표치에 대한 총성능지수를 1로 했을 때의 상대적인 성능 수준을 의미한다. 즉, 총성능지수가 1보다 클수록 성능이 나쁘고, 1보다 작을수록 성능이 좋음을 의미한다.

$$\text{총성능지수} = \sum_i \left(\frac{\text{지표값}_i}{\text{성능목표값}_i} \times \text{가중치}_i \right) \dots\dots (2)$$

이와 같이 산출한 총성능지수는 성능에 대한 가시적 표현으로써 이해관계자 간의 의사소통 수단으로 활용될 수 있다.

<표 1> 성능지수에 의한 성능평가 사례

성능지표	단위	가중치	성능목표	튜닝전		튜닝후		
				지표값	성능지수	지표값	성능지수	
응답속도	CPU Time/exec.	초	0.2	0.02	0.002	0.022	0.0	0.016
	Elapsed Time/exec.	초	0.2	0.05	0.005	0.019	0.0	0.015
I/O 량	Physical I/O per exec.	blocks	0.3	0.5	2.1	1.3	0.2	0.1
	Logical I/O per exec.	blocks	0.2	30	44.5	0.3	32.2	0.2
기타	Parse Overhead	%	0.1	10	12.4	0.1	11.9	0.1
총성능지수				1		1.7		0.5



<그림 2> 튜닝 전·후 성능지수 변화 비교

3.2 성능지표 항목의 선정

성능지표 항목은 SQL의 성능을 잘 표현할 수 있는 항목들로 선정하는데, SQL의 성능은 I/O량에 좌우되므로, SQL의 처리에 소요되는 I/O량과 이에 따른 응답속도를 반영하는 항목으로 선정한다. 성능지표 항목은 DBMS의 종류에 따라 다를 수 있으며, <표 2>는 오라클과 MS SQL Server의 경우에 측정 가능한 성능지표 항목의 예다.

<표 2> DBMS별 성능지표 항목의 예

	SQL 응답속도	I/O 량	기타
오라클	CPU Time/exec Elapsed Time/exec	Physical I/O (Disk) Logical I/O (Memory)	Parse Overhead
MS SQL Server	CPU Time/exec Duration/exec	Page I/O 또는 Page Read/Write	-

성능지표 항목의 측정은 각 DBMS 벤더에서 제공하는 SQL 성능측정 유틸리티S/W 사용 결과로부터 산출하며, 유틸리티 S/W의 예로는 오라클의 SQL Trace나 Trace Analyzer, MS SQL Server의 Profiler나 Read80 Trace 등이 있다. 아래 <표 3>은 오라클에서 SQL Trace를 통해 얻은 SQL 성능자료이며, <그림 3>과 같이 성능지표 항목의 측정값을 얻을 수 있다.

<표 3> 오라클 SQL Trace로 얻은 SQL 성능자료 사례

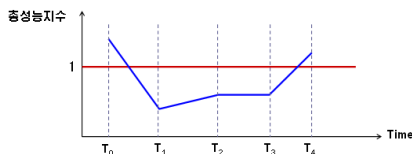
Call	Count	Cpu	Elapsed	disk	query	current	rows
Parse	12092223	7023.19	6961.74	507	9462	475	0
Execute	19264438	8682.46	12948.95	9875552	36823910	19149866	5677215
Fetch	16026098	30430.2	35679.86	2219173	1166674567	13034015	42507352
Total	47389759	46135.85	54990.55	12095232	1203501939	32184356	48184567

$CPU\ Time/exec. = Total\ CPU\ Time \div SQL\ 총\ 실행수$
 $= 46135.85 \div 19264438 = 0.0024\ (초)$
 $Physical\ I/O\ per\ exec. = Total\ Disk\ I/O \div SQL\ 총\ 실행수$
 $= 12095232 \div 19264438 = 0.63\ (blocks)$
 $Logical\ I/O\ per\ exec. = Total\ Memory\ I/O \div SQL\ 총\ 실행수$
 $= (query\ total + current\ total) \div SQL\ 총\ 실행수$
 $= (1203501939 + 32184356) \div 19264438 = 64.1\ (blocks)$

<그림 3> 성능지표 측정값 산출 사례

3.3 목표지향형 성능관리 프로세스

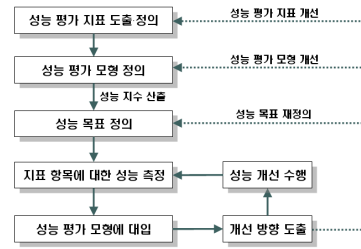
성능지수에 의한 성능평가 모형에 따라 성능목표를 설정하고 주기적으로 총성능지수를 측정하면 <그림 4>와 같이 성능수준의 변화를 추적해 볼 수 있으며, 이를 통해 지속적인 목표관리 및 성능 유지를 위한 튜닝 시점의 판단이 가능해진다.



<그림 4> 총성능지수의 변화 추이 추적 사례

<그림 4>의 T₀ 시점에 측정된 총성능지수는 튜닝 전의 성능 수준을 표현한 것으로, 시스템의 성능이 매우 좋지 않기 때문에 1 보다 높은 수치로 나타나고 있다. 여기에 튜닝을 수행하고 T₁ 시점에 총성능지수를 다시 측정한 결과 1 보다 적은 수치로 나타나 성능 수준이 향상되었음을 나타내며, 수치의 변화 정도는 튜닝의 효과성을 계량하는 지표로 사용될 수 있다. 같은 방법으로 T₂ ~ T₄ 시점에 총성능지수를 측정하여 성능수준의 변화를 객관적으로 입증하고 시스템 전반에 걸친 튜닝이 필요한지 여부를 판단하는 근거 자료로 활용할 수 있다.

<그림 5>는 <표 1>과 <그림 2>, <그림 4>에 따라 성능지표를 설정하고 성능평가 모형을 정의하여 지표 항목에 대한 측정값으로부터 성능 목표에 근접하는 정도를 주기적으로 관리하면서 성능을 유지해 나가는 절차를 표현한 것으로, 목표지향형 성능관리 프로세스라 정의한다. <그림 5>의 프로세스에서 성능목표의 정의는 전체적인 성능관리 수준을 결정하는데 중요한 역할을 하기 때문에 초기에는 유사 업종의 평균적인 성능수준을 고려하여 결정하고, 성능관리를 수행하면서 점차 현실화하거나 경험을 반영한 의욕치를 적용할 수 있다. 성능목표는 대상 시스템의 OLTP와 Batch 처리 비중에 따라라도 다를 수 있고, 같은 업종의 같은 목적을 가진 시스템일지라도 데이터베이스



<그림 5> 목표지향형 성능관리 프로세스 수행 절차

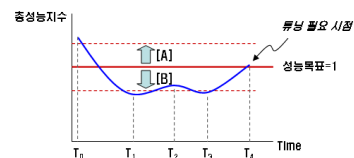
시스템의 환경, 사용현황, 보유기술수준 등과 같은 여러 요인에 의해 다르게 설정될 수 있다. <표 4>는 한국표준산업분류에 따른 업종 분류를 토대로 OLTP 위주로 구성된 9개 업종의 26개 사이트에서 측정된 튜닝 후의 성능지수에 대한 실측 데이터의 평균값이며, 이를 참고로 초기의 성능목표를 설정할 수 있다.

<표 4> OLTP 위주 시스템에서 측정된 튜닝 후 성능지수 평균값

업종	오라클				MS SQL Server		
	CPU Time /exec	Elapsed Time /exec	Physical I/O per exec	Logical I/O per exec	CPU Time /exec	Duration /exec	Page Read/Writes /exec
금융업	0.0115	0.0349	3.7	132.2			
컴퓨터 프로그래밍, 시스템 통합 및 관리업	0.0016	0.0025	0.5	63.7			
정보서비스업	0.0025	0.0054	2.5	37.7	0.0063	0.0083	79.2
교육서비스업	0.0020	0.0111	0.03	41.79			
공공행정, 국방 및 사회보장 행정업	0.0088	0.0161	5.9	93.9			
전자부품, 컴퓨터, 영상, 음향 및 통신장비 제조업	0.0013	0.0033	1.6	73.7			
소프트웨어, 컴퓨터, 영상, 음향 및 통신장비 제조업	0.0023	0.0058	1.7	24.9			
항공 및 운송관련 서비스업	0.0041	0.0049	0.4	67.7			
항공, 가스, 증기 및 공기조절 공급업	0.0038	0.0056	2.7	31.6			
연세 평균	0.0044	0.0100	2.4	64.4	0.0063	0.0083	79.2

3.4 튜닝 시점의 판단

본 논문의 제안 모델은 주기적으로 측정된 총성능지수의 변화 추이로부터 최소 비용으로 성능 유지가 가능한 튜닝 시점의 판단을 지원한다. <그림 6>은 총성능지수에 대한 추이 변화를 표현한 것으로, 성능 목표치의 총성능지수를 1로 보았을 때 1보다 커질수록 성능 문제가 심각함을 의미한다.



<그림 6> 총성능지수의 변화 추이와 성능목표 조정 사례

빈번한 대량 데이터 처리나 사용자-데이터량 급증 등과 같은 환경적 변화로 다수의 SQL이 성능 문제를 유발하게 되면 T₄ 시점과 같이 총성능지수가 목표값을 넘어서게 되고, 시스템 전반에 대한 재튜닝이 필요하게 된다. 이때 조기에 튜닝에 착수할수록 상대적으로 적은 노력과 비용으로 해결이 가능하기 때문에 성능 목표를 넘어서는 T₄ 시점을 재튜닝이 필요한 시점으로 판단할 수 있다. 이와 같은 판단은 튜닝 필요성에 대한 근거가 되고 이해관계자간에 공감대를 형성하여 이들의 협조를 이끌어내는 객관적이고 정량적인 기초 자료가 될 수 있다.

<그림 5>에 따른 성능관리 수행 시 보유 기술력과 관리 수준을 고려하여 <그림 6>의 Case [A]나 Case [B]와 같이 목표 수준을 올리거나 낮춰 성능 목표를 조정할 수 있다. Case [A]와 같이 성능목표를 높이면 상대적으로 적은 노력과 비용으로 목표 달성이 가능하며,

보유 기술력이 높지 않거나 빈번한 대량 데이터 처리, Batch 처리 비중이 높은 환경에서 고려될 수 있다. 반대로, Case [B]와 같이 성능 목표를 낮추면 목표 달성을 위해 더 많은 노력과 기술력이 요구될 수 있으나 높은 사용자 만족도와 시스템 활용도 향상을 이끌어 낼 수 있다. 성능목표 달성 후 성능목표 이내로 유지하는 데에도 상당한 기술력이 요구되며, 데이터베이스 관리자는 물론이고 데이터베이스시스템 개발·유지보수 담당자들의 SQL 사용 수준이 높은 조직에서 적용 가능하다. 소규모의 잦은 튜닝이 발생할 수 있으나 높은 보유 기술력을 감안하면 사용자 만족도가 매우 높은 성능 수준을 유지할 수 있기 때문에 의욕치를 반영한 성능관리 형태로 고려될 수 있다.

4. 검증 및 평가

제안 모델이 데이터베이스시스템의 성능 수준을 수치화하여 용이하게 표현할 수 있는지를 검증하기 위해 오라클 DBMS를 사용하는 사이트에서 튜닝 전·후의 성능지표 항목에 대한 측정값을 산출하여 성능평가 모형에 적용하였다.

<표 5>는 제조업종 S전자에서 오라클 SQL Trace 데이터를 이용하여 산출한 튜닝 전·후의 성능지표 측정 결과이고, 이를 성능평가 모형에 대입하여 <표 6>과 같이 성능지수를 산출하였다. 산출한 성능지수를 <그림 7>과 같이 표현함으로써 추가적인 튜닝이 필요한 지표 항목의 존재 여부를 설명할 수 있고, <그림 8>와 같이 총성능지수에 대한 변화 정도를 분석해 봄으로써 성능 수준의 향상도를 평가할 수 있다.

<표 5> 제조업종 S전자, 튜닝 전·후 성능지표 측정결과

call	count	cpu	elapsed	disk	query	current	rows
Parse	425972	51.22	51.95	0	147	0	0
Execute	601509	315.82	417.9	24667	8001516	3380609	300620
Fetch	340893	961.49	2432.59	1109552	15414505	0	232738
total	1368374	1328.53	2902.45	1130319	23416168	3380609	53358

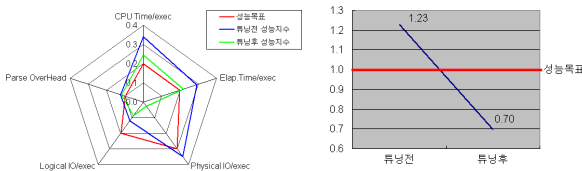
1회 SQL 실행당 평균 CPU Time (초) : 0.0022 초
 1회 SQL 실행당 평균 Elapsed Time (초) : 0.0048 초
 1회 SQL Execution당 평균 논리IO/block I/O : 1.88 Blocks
 1회 SQL Execution당 평균 물리IO/block I/O : 44.88 Blocks

call	count	cpu	elapsed	disk	query	current	rows
Parse	590606	71.03	71.88	0	0	0	0
Execute	599395	408.79	602.91	26415	1714030	3291271	195146
Fetch	514350	474.24	1499.17	61228	14288748	0	280281
total	1704351	954.06	2173.97	87643	16002778	3291271	435427

1회 SQL 실행당 평균 CPU Time (초) : 0.0016 초
 1회 SQL 실행당 평균 Elapsed Time (초) : 0.0036 초
 1회 SQL Execution당 평균 논리IO/block I/O : 0.16 Blocks
 1회 SQL Execution당 평균 물리IO/block I/O : 32.19 Blocks

<표 6> S전자 성능지표 측정값의 성능평가 모형 대입 결과

성능지표항목	기준치	성능목표	튜닝전	튜닝전 성능지수	튜닝후	튜닝후 성능지수
CPU Time/Exec	0.2	0.0013	0.0022	0.34	0.0016	0.24
Elap.Time/Exec	0.2	0.0033	0.0048	0.29	0.0036	0.22
Physical IO/Exec	0.3	1.6	1.88	0.35	0.15	0.03
Logical IO/Exec	0.2	73.7	44.55	0.12	32.19	0.09
Parse OverHead	0.1	10	12.4	0.12	11.9	0.12
총성능지수	1			1.23		0.70

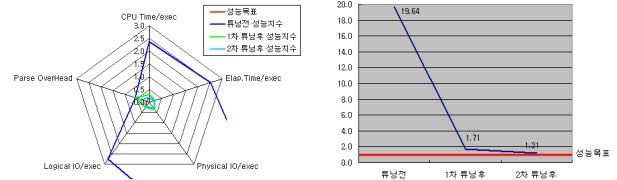


<그림 7> S전자 튜닝 전·후 성능지수 변화 비교

제안 모델에 따른 프로세스를 수행하면서 <그림 8>에 나타난 총 성능지수 변화를 계속적으로 추적한다면 현실적인 관리 수준에 근

접한 성능목표치를 도출하여 성능평가 모형을 재정의하고 <그림 5>의 프로세스에 따라 지속적인 성능관리를 수행할 수 있을 것이다.

<그림 9>와 <그림 10>은 시스템 통합 및 관리업종의 A사에서 같은 방법으로 오라클 SQL Trace 데이터를 이용하여 3차례의 성능수준 변화를 분석한 결과이며, 튜닝을 통해 성능수준이 목표에 근접하고 있음을 잘 보여주고 있다. A사의 경우에도 제안모델에 따른 성능관리가 튜닝의 효과성을 입증하고 성능목표 및 성능수준 관리를 용이하게 함을 알 수 있다.



<그림 9> A사 튜닝 전·후 성능지수 변화 비교

5. 결론

이 논문은 관계형 데이터베이스시스템의 성능관리에 대해 성능지수를 이용한 성능평가 모형과 이에 기반한 목표지향형의 성능관리 프로세스 모델을 제시하며, 이를 통해 이해관계자 간의 의사소통 활성화와 지속적인 성능관리 및 관리 수준 향상에 대한 방법을 제시하였다.

제안한 모델은 오라클 데이터베이스를 사용하는 사이트의 실제 튜닝 사례에 적용하여 제안 모델이 성능 수준을 수치화하고 성능관리에 적용될 수 있음을 증명하였으며, 오라클 외의 다른 DBMS에 대해서도 성능지표 항목을 정의하고 성능평가 모형에 반영하여 동일한 방법으로 성능관리를 수행할 수 있음을 보였다.

이 논문에서 제시하는 목표지향형 성능관리 모델의 범용성과 활용성을 향상시키기 위해 향후 업종별 유형에 따른 성능지수의 기준 및 목표 수준의 정의에 대해 보다 많은 데이터를 통해 검증해 봄으로써 성능 목표의 객관성을 향상시키는 연구를 진행할 예정이다.

참고문헌

- [1] "Forecast, Database Management Systems Software, Worldwide, 2005-2010", Update, Gartner, 2006.
- [2] 임선영, "대형 서버 기반 복합 솔루션 시스템을 위한 모니터링 시스템 구현", 서강대 정보통신대학원, 2008.
- [3] Richmond Shee, Kirtikumar Deshpande, K Gopalakrishnan, "Oracle Wait Interface: A Practical Guide to Performance Diagnostics & Tuning", McGraw-Hill, 2004.
- [4] 서한준, itSMF NL, "ITIL 기반의 IT 서비스 관리 ITSM", Nemo Books, 2006.
- [5] J. Gray, "The Benchmark Handbook for Database and Transaction Processing Systems", 2nd ed. Morgan Kaufmann, 1993.
- [6] Oracle, "Oracle Database Performance Tuning Guide 11g Release 1", Oracle, 2008.
- [7] 이화식, "새로운 대용량 데이터베이스 솔루션 1", (주)엔코아 컨설팅, 2005.