튜닝 가능한 자원선택 방법론

김 혜 숙* · 오 정 석**

Methodologies to Selecting Tunable Resources

Hyesook Kim* · Jeong Soek Oh**

Abstract

Database administrators are demanded to acquire much knowledges and take great efforts for keeping consistent performance in system. Various principles, methods, and tools have been proposed in many studies and commercial products in order to alleviate such burdens on database administrators, and it has resulted to the automation of DBMS which reduces the intervention of database administrator. This paper suggests a resource selection method that estimates the status of the database system based on the workload characteristics and that recommends tuneable resources. Our method tries to simplify selection information on DBMS status using data-mining techniques, enhance the accuracy of the selection model, and recommend tuneable resource. For evaluating the performance of our method, instances are collected in TPC-C and TPC-W workloads, and accuracy are calculated using 10 cross validation method. comparisons are made between our scheme and the method which uses only the classification procedure without any simplification of informations. It is shown that our method has over 90% accuracy and can perform tuneable resource selection.

Keywords : Workload Characteristics, Tunable Resources, Methodologies

논문접수일 : 2007년 11월 03일 논문게재확정일 : 2008년 03월 19일

* 숭실대학교 전산원 멀티미디어학과 교수, e-mail : hskim@ssuci.ac.kr
** 한국가스안전공사 가스안전연구원 선임연구원, e-mail : dbstar1@paran.com

1. 서 론

데이타베이스 시스템에서 데이타베이스 워크로드 특성은 자원사용(resource usages), 접근 방식(access method), 질의 실행 계획(query execute plan)에 따라 다양하게 변화될 수 있다[Aboylnaga and Chaudhuri, 1999]. 데이타베이스 관리자는 데이타베이스 시스템에서 사용되는 워크로드 특성 변화를 인지해 데이타베이스 시스템의 성능을 적절하게 유지해야 한다. 일정한 성능 유지를 위해 관리자는 데이타 접근 방식, 물리적 구조, 자원 할당량 등을 효과적으로 조율해야 하며 데이타베이스 시스템 관련 지식, 워크로드 정보, 성능 하락을 유도하는 현상 등을 알아야한다[Chaudhuri and Weikum, 2000]. 데이타베이스 시스템의 관리는 관리자에게 많은 시간과 노력을 요구하기 때문에 큰 부담을 줄 수 있다.

데이타베이스 시스템의 관련 연구들과 상용 제품들은 관리자의 부담을 줄이기 위해 효율적으로 데이타베이스 시스템을 관리하거나 튜닝을 수행하는 다양한 원리, 기법, 도구들을 제시해왔다. 다양한 방법 중 연구가 주로 진행된 분야는 물리적 구조의 튜닝, 질의 튜닝, 자원 사용에 대한 튜닝이다. 물리적 구조의 튜닝은 질의 최적기의 질의 비용 계산을 기준으로 질의 수행에 낮은 비용을 산출할 수 있는 구조를 선택하거나 생성하는 문제에 초점을 둔다. 특히, 마이크로소프트는 가상의 물리적 구조와 실제 구조를 비교하여 낮은 수행비용이 산출되는 구조를 선택하고 생성하는 기법을 제시한다[Argrawal et al., 2000; Chaudhuri and Narassayya, 1998; Chaudhuri and Narassayya, 2000; Chaudhuri and Weikum, 2000]. 질의 튜닝은 일반적으로 물리적 구조를 효율적으로 접근하는 질의를 선택하는 문제에 초점을 두며, 오라클은 사용자 질의에 대해 동일한 결과를 얻으면서 낮은 비용을 산출하는 질의로 자동 변환되는 기법에 대해 소개하고 있다[Cyran, 2001]. 자원 사용에 대한 튜닝은 효율적인 자원의 할당 및 사용에 초점을 둔다[Brown et al., 1996; Benoit, 2000; Brown et al., 1994; ,Martin et al., 2000]. 자원 사용에 대한 튜닝은 일반적으로 데이타 버퍼 할당에 관한 연구가 많이 수행되었으며, 작업 메모리(working memory)나 I/O 프로세스 등을 함께 고려하는 다중 자원의 할당에 관한 연구도 수행되었다. [Weikum et al., 1999]은 자원 할당의 자동화에 관한 주제를 중심으로 연구가 진행되었으며 관련된 수식 및 알고리즘이 소개되어 있다.

효율적인 데이타베이스 시스템의 관리에 관한 연구는 지난 10여 년 동안 관리자의 개입을 최소화하여 전체 시스템 관리에 소요되는 비용을 줄이고 다양한 튜닝 방법과 쉬운 관리/튜닝 도구의 제시를 위해 노력해왔으며 데이타베이스 시스템의 자동화로 발전해왔다[Chaudhuri and Narassayya, 1998; Chaudhuri and Weikum, 2000; Elnaffar et al., 2003; Weikum et al., 1999]. 데이타베이스 시스템의 자동화는 DBMS 개발업체와 관련된 연구단체를 중심으로 수행되어 왔으며[Ganek and Corbi, 2003]은 자체 최적화(self-optimizing), 자체 설정(self-configuring), 자체 복구(self-healing), 자체 보호(self-protecting), 자체 구성(self-organizing), 자체 검사(self-inspecting)를 기준으로 현 데이타베이스 시스템의 자동화 현황을 설명하였다.

데이타베이스 시스템의 상태 및 성능에 대한 예측은 수식 모델이나 마이닝 모델을 적용해 근사적으로 접근하는 연구가 진행되고 있다[Elnaffar 2002; Elnaffar et al., 2003; Martin et al., 2000; Weikum et al., 1999]. 수식 모델은 많은 연구에서 수식이 제안되고 검증을 수행하였으나 가설의 정의 및 수식의 전개가 어렵고 비선

형 형태의 그래프로 표현되는 데이타베이스 상태를 해결할 수 있는 수식이 요구된다[Goebel and Gruenwald, 1999]. 마이닝 모델은 데이타베이스 분야에서 현재 워크로드의 자동 분류를 위해 사용되는 정도이고 특정 데이타베이스의 상태를 예측하는 수준까지 진행되지 않았으므로 데이타베이스 상태를 예측할 수 있는 마이닝 모델의 구성 기법이 필요하다.

본 연구는 자동화를 위한 DBMS 구조를 개발하거나 표준적인 인터페이스를 구축하여 현 DBMS의 한계성을 해결하지 않지만 데이타베이스 시스템에서 제공하는 기능과 통계 정보를 이용하여 워크로드 특성에 따른 데이타베이스 시스템의 상태를 판단하고 적합한 자원 추천을 목적으로 한다. 적합한 자원 추천은 데이터 마이닝 기법을 이용한 자원 추천 방법을 통해 수행된다. 본 논문의 자원 추천 방법론은 첫째로, 인스턴스를 수집하고 군집 기법에 의해 추천 정보를 단순화하며 둘째로, 분류 기법에 의해 추천 모델을 구축하고 마지막으로 구축된 모델을 검증하고 자원 추천을 선택한다. 인스턴스는 4가지 자원(데이타 버퍼, 공유 메모리, 개인 메모리, I/O 프로세스)을 기준으로 60개의 할당 크기에 대해 TPC-C[TPC, 2001]와 TPC-W[TPC, 2002]를 수행시켜 총 120번의 실험을 통해 1600여 개를 수집하였다. 본 논문의 결과는 효율적인 데이타베이스 시스템 관리를 위해 데이터베이스에서 기록되었던 상태를 기반으로 적절한 자원의 크기를 추천하며, 워크로드에 따라 자원의 크기를 자동으로 변화시켜 관리자의 개입을 감소시키고 데이타베이스 시스템 자동화 연구 중 자체 검사를 위한 초석을 제시하였다.

본 논문의 구성은 다음과 같다. 제 2장은 데이터 마이닝 기법을 이용한 튜닝 가능한 자원 추천 방법에 대해 설명한다. 제 3장은 분석 기법을 이용한 추천 모델과 3단계 자원 추천 방법론을 이용한 추천 모델을 수행하고 비교한다.제 4장은 연구 결과에 대한 결론을 맺고 향후계획을 제시한다.

2. 튜닝가능한 자원 선택 방법

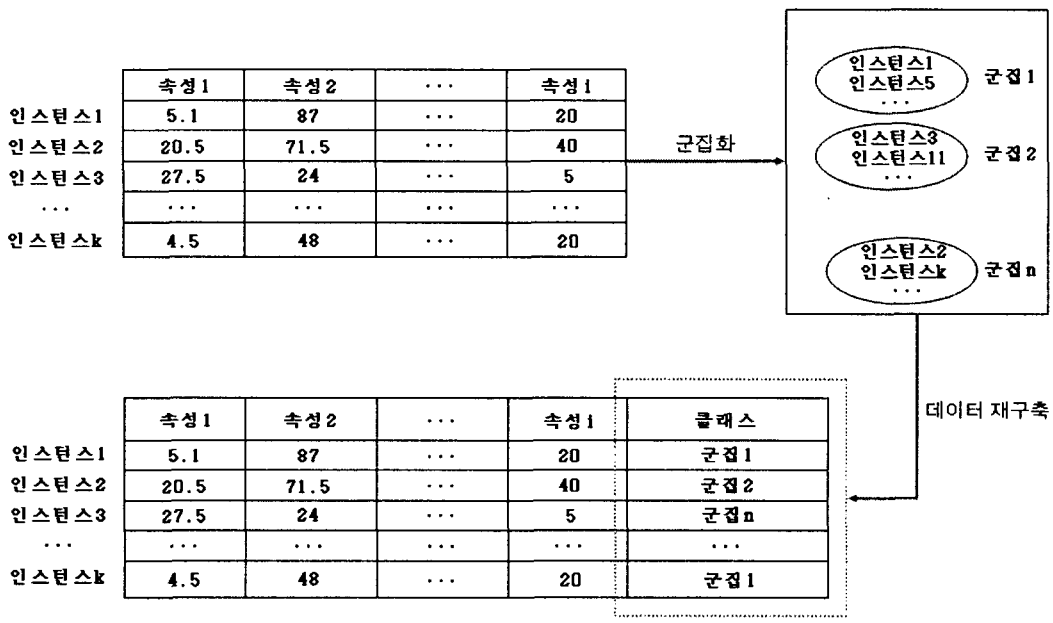
기존에 우리는 TPC-C와 TPC-W 환경에서 자원 할당과 데이타베이스 시스템의 상태에 영향을 주는 요소를 발견하였으며 두 워크로드에 따라 적절한 자원 사용의 지침을 제시할 수 있었지만 워크로드간의 비교와 분석 수행 시간이 많이 걸리고 개인의 지식에 따라 분석 결과가 틀려질 수 있었다. 또한, 유사한 범위를 기록하는 데이터베이스 워크로드 때문에 정확한 데이터베이스 시스템의 상태를 관리하기가 어려워 데이터베이스 튜닝을 위해 튜닝 가능한 자원이 무엇인지 결정하기가 어려웠다.

본 연구는 데이터 마이닝 기법을 이용하여 데이터베이스 시스템 워크로드를 분석하고 데이터베이스 시스템 상태에 따라 튜닝 가능한 자원을 선택하는 방법 연구를 목적으로 한다. 마이닝 기법 적용을 위해 인스턴스 구축 및 클래스 정의를 선행하고 자원선택을 위한 데이터 마이닝 모델 구축 및 검증이 수행되어야 하며 결과로서 현 시스템 상태에 따른 튜닝 가능한 자원의 선택이 이루어져야 한다.

인스턴스 구축은 데이타베이스 시스템 구동 환경에서 상태 데이터의 수집을 의미한다. 상태 데이터는 워크로드 특성의 변화, 수행 시간의 차이, 자원 할당의 차이에 따라 다르게 기록되므로 세 가지 특성 들이 인스턴스를 수집하는 판단기준이 되어야 한다. 즉, 워크로드 특성 변화에서 워크로드는 변화 및 워크로드 식별 정보가 제공되어야 함을 의미한다. 자원 할당 차이에서 인스턴스는 데이타베이스 시스템에 영향을 주는 자원을 선정하여 할당된 자원 크기 변화 정보를 제공해야 한다. 수행 시간 차이의 기

준에서 인스턴스는 동일 자원 및 동일 워크로드를 갖는 인스턴스라도 시간에 따라 다른 값을 제공하기 때문에 시스템 환경에서 수행 시간 별 정보를 제공해야 한다. 세 가지 판단기준은 인스턴스 속성(attribute)들에 의해 기술되고 판단되어진다. 인스턴스 속성의 선별 기준은 데이타베이스에 출현했던 워크로드의 특징을 구분하는 식별성(identities), 데이타베이스 시스템의 구동 중에 쉽게 읽거나 산출될 수 있는 접근성(accessibility), 시스템 인자 (parameter)에 대한 관련성(relevancy)을 중심으로 선별되어야 한다.

클래스 정의는 전체 인스턴스에 대한 분석을 요구하며, 전체 인스턴스 분석은 시스템에 대한 다양한 영향을 고려해야하므로 어렵고 소요 시간이 많이 걸리는 작업이다. 데이타베이스 시스템에서 발생되는 워크로드는 다양하고 복잡하므로 클래스를 과도하게 분류하여 정확성을 하락시킬 수 있다. 과도한 분류를 억제하기 위한 효과적인 클래스 생성은 선택정보의 군집화 과정을 통해 수행될 수 있다. 선택 정보의 군집화는 전체 인스턴스를 몇 개의 작은 군집으로 분할하여 관련성 높은 인스턴스를 동일한 군집으로 재배치한다. <그림 1>은 i개의 인스턴스 속성이 존재하고 k개의 인스턴스가 존재할 때 n개의 군집으로 인스턴스가 재배치되는 예를 보인다. 인스턴스 집합은 군집으로 재배치되며, 생성된 군집은 클래스로서 활용된다. 선택정보의 군집화는 "unsupervised" 마이닝의 군집 기법을 이용해서 수행될 수 있다. 군집 기법은 수행 과정 중에서 전체 인스턴스 분석, 인스턴스의 군집 할당, 추천 정보의 생성에 대해 사람의 개입을 감소시키므로 소요되는 시간을 감소시키고 추천 정보를 효율적으로 생성한다. 데이터 마이닝 모델의 구축은 추천 정보의 군집화를 통해 재구축된 인스턴스를 이용하여 모델을 생성하는 방법에 대해 설명한다. 마이닝의 군집 기법을 이용해 군집화된 정보는 각 인스턴스에 대한 선택 정보로서 재구축 되어야 한다. <그림



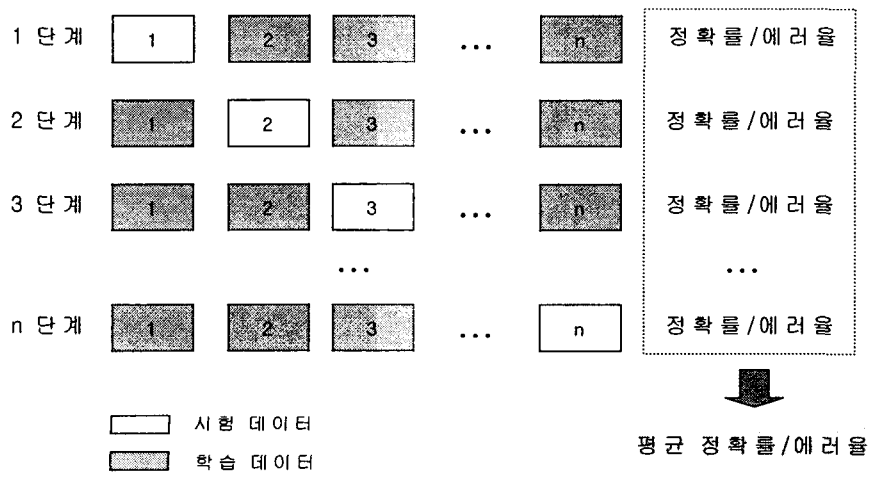
〈그림 1〉 인스턴스 구축 방식

1>은 추천 정보의 군집화로 생성된 n개의 군집에서 k 개의 인스턴스에 대해 자원 추천을 위해 인스턴스가 재구축 되는 예를 보이고 있다.

재구축된 인스턴스를 기반으로 생성되는 선택 모델은 근사(approximation) 기법을 적용하여 클래스에 대한 인스턴스의 분류로서 생성된다. 근사 기법은 가설 및 통계적 수식의 정의가 필요 없는 "supervised" 마이닝을 이용한다. 마이닝 기법은 모델 구축 과정에서 의미 있는 정보를 추출하기 때문에 인스턴스 속성간의 관계나 인스턴스와 추천정보의 관계를 명확히 규명할 필요가 없다.

모델의 검증은 일반적으로 인스턴스 분할을 이용하는 방법을 사용한다. 인스턴스 분할을 이용하는 방법은 추천 모델에 사용되었던 인스턴스를 n개의 조각으로 분할하여 교차 검증을 수행한다. 전체 인스턴스의 분할은 가능한 동일한 인스턴스 개수와 클래스를 가지는 n개의 집합으로 재배치되어야 한다. <그림 2>는 인스턴스를 n개의 조각으로 분할하여 검증을 수행하는 n 교차 검증에 대한 수행과정을 보인다. n 교차 검증은 총 n개의 단계로 구분되어 검증이 수행된다. 각 단계는 클래스가 균등하게 분포된 n개의 조각에서 n-1 조각을 이용해서 추천 모델을 생성하고 나머지 한 조각을 시험 데이타로 이용해서 추천 모델에 대한 정확률(에러율)을 계산한다. 최종 정확률(에러율)은 n개의 정확률(에러율)에 대한 평균을 이용한다.

구축된 선택 모델은 새로운 인스턴스의 입력에 대해 유사한 특성을 가진 인스턴스 집합인 클래스로 결과를 반환한다. 클래스 단위의 결과는 여러 개의 인스턴스 들이 포함되어 복수의 자원 환경을 추천 결과로서 반환하기 때문에 모호하다. 튜닝가능한 자원 선택은 입력된 인스턴스에 대해 가장 근사한 인스턴스를 찾아 수집된 데이터베이스 시스템 환경 기록을 반환해야 하므로 인스턴스 단위의 추천이 수행되어야 한다. 추천 인스턴스는 입력된 인스턴스와 거리를 계산하여 가장 낮은 거리값을 가진 인스턴스로 선택된다. 일반적으로 거리를 구하는 공식은 유클리디안, 맨하탄 등이 적용되지만 이런 거리 방식들은 고차원을 가진 인스턴스에서 거리를 구할 때 값의 범위가 작은 차원에 대하여 편향을 발생시킨다. 본 연구는 고차원에서 특정 차원에



〈그림 2〉 n 교차 검증의 수행 과정

유클리디안 $\sqrt{\sum_{k=1}^{n}|X_{ik}-X_{jk}|^2}$

맨하탄 $\sum_{k=1}^{n}|X_{ik}-X_{jk}|$

변이계수 $\sum_{k=1}^{n}\frac{|X_{ik}-X_{jk}|}{\left(\frac{(X_{ik}+X_{jk})}{2}\right)}$

새로운 인스턴스 집합

27.00, 94.00, 78.30, 24.00, 85000, 230000, 75000, 160900000, 35

클래스에 있는 인스턴스 집합

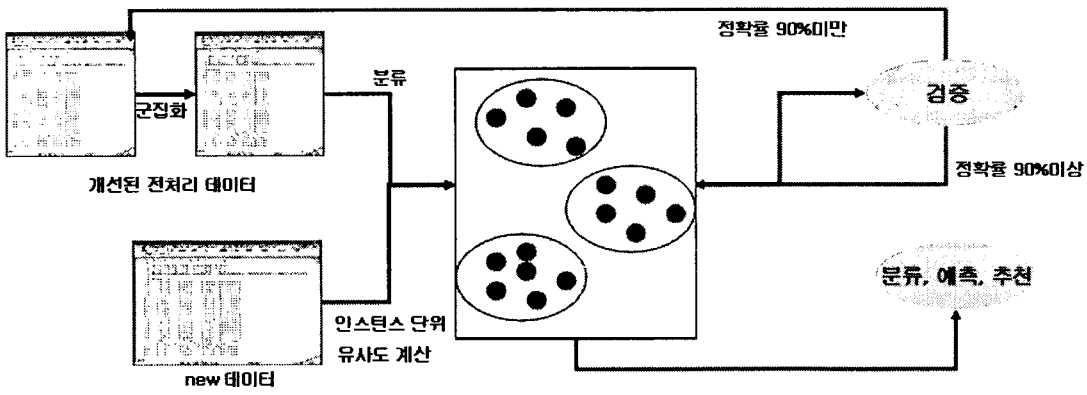
1. 26.69, 89.71, 78.23, 23.03, 80251, 440058, 83052, 159735244, 35
2. 27.06, 92.61, 78.17, 22.51, 77326, 315628, 80720, 162347304, 35
3. 26.84, 95.53, 78.23, 22.86, 67573, 190136, 69487, 159093032, 35

유클리디안에서 산출된 거리값	맨하탄에서 산출된 거리값	변이계수에서 산출된 거리값
인스턴스 1 : 1183582.794	인스턴스 1 : 1387620.37	인스턴스 1 : 0.891
인스턴스 2 : 1449866.418	인스턴스 2 : 1546329.07	인스턴스 2 : 0.574
인스턴스 3 : 1807500.093	인스턴스 3 : 1869774.9	인스턴스 3 : 0.575

〈그림 4〉 인스턴스간의 거리 공식과 계산 예

의해 야기되는 편향을 막기 위해 두 인스턴스 간의 변이계수를 거리로 사용하는 것을 제안한다.

<그림 3>에서 인스턴스는 총 9개의 차원을 가지고 있고, 각 차원의 다른 값의 범위를 지니고 있고 입력 인스턴스 하나와 클래스에 존재하는 인스턴스 세 개를 유클리디안, 맨하탄, 변이계수 거리 공식을 적용하여 산출된 거리를 보여준다. 유클리디안과 맨하탄 공식은 인스턴스 1, 인스턴스 2, 인스턴스 3순으로 낮은 거리값이 계산되어 인스턴스 1을 추천한다. 유클리디안과 맨하탄 공식은 다른 범위를 가지는 각 차원들의 값을 동일하게 취급되어 값의 범위가 적은 전반부 4가지 차원의 변경 영향이 값의 범위가 큰 후반부 차원의 변경 보다 고려되지 못해서 후반부 값이 적은 인스턴스 1이 추천되었다. 변이 계수는 편차를 평균으로 나눔으로써 값의 범위에 따라 발생되는 편향을 감소시켜 인스턴스 2가 추천되었다.



〈그림 4〉 튜닝가능한 자원선택 방법을 위한 과정

튜닝 가능한 자원선택을 요약해보면, <그림 4>처럼 전처리, 모델 구축, 검증, 변이계수를 이용한 인스턴위 단위 탐색, 분류/예측 과정 등으로 도식화될 수 있다.

3. 튜닝가능한 자원 선택을 위한 실험환경

인스턴스 집합은 데이타베이스 시스템 구동 중에 워크로드의 식별, 자원의 종류, 자원의 할당 크기에 대한 정보를 포함해 수집되어야 한다. 이를 위해, 본 연구는 TPC-W와 TPC-C 성능평가를 수행하여 데이타베이스 시스템의 구동 환경을 구축하고, 4개의 자원(데이타 버퍼, 공유 메모리, 개인 메모리, I/O 프로세스)을 이용한다[Morals and Lorentz, 2001]. 인스턴스 수집 실험은 자원 당 15번의 할당 크기를 변경하여 총 120회 (워크로드 수 × 자원의 종류 × 자원 변경회수)를 독립적으로 수행하였다. <표 1>은 자원 할당의 크기 변경을 위한 초기값, 증가값, 최대값을 보여준다. 자원의 초기값은 데이타베이스 시스템이 제공하는 기본값을 이용하고, 증가값은 각 자원의 기본값을 단위로 하여 증가한다. 자원의 최대값은 초기값의 15배의 자원 크기로 확장한다. 중심이 되는 자원이 변경되었을 때 다른 자원들은 초기값을 기본으로 한다. 예를 들어, 데이타 버퍼의 인스턴스 수집은 초기값부터 최대값까지 증가값 단위로 변경하면서 수행되고 다른 자원들은 초기값으로 설정된다.

〈표 1〉 자원의 종류와 변경 범위

자 원	초기값	증가값	최대값
데이타 버퍼	32MB	32MB	480MB
공유 메모리	32MB	32MB	480MB
개인 메모리	20MB	20MB	300MB
I/O 프로세스	1개	1개	15개

수집은 시험동안 5분 단위로 데이타베이스 시스템의 상태를 기술하는 인스턴스 속성의 기록에 의해 수행된다. 인스턴스 속성은 데이타베이스에서 제공되는 통계 정보를 사용하며 TPC-C와 TPC-W 환경에서 워크로드에 따른 자원 사용 형태와 변화를 식별할 수 있었던 8개의 성능 지표를 이용하고 종류는 다음과 같다.

- 데이타 버퍼 적중률 : 검색하는 데이타가 데이타 버퍼에 존재할 확률을 의미한다.
- 공유 메모리 적중률 : 검색하는 질의 정보가 공유 메모리에 존재할 확률을 의미한다.
- 메모리 파싱비율 : 파싱을 수행할 때 공유 메모리에 존재하는 데이타를 읽어 파싱을 수행할 비율을 의미한다.
- 데이타 변경률 : 디스크에서 메모리로 읽은 데이타가 변경될 비율을 의미한다.
- 데이타 버퍼 읽기량 : 디스크에서 데이타 버퍼로 읽는 데이타의 용량을 의미한다.
- 데이타 버퍼 쓰기량 : 데이타 버퍼에서 디스크로 쓰는(write) 데이타 용량을 의미한다.
- 디스크 쓰기량(비체크포인트) : 체크 포인트가 아닌 다른 이유로 디스크로 쓰이는 데이타의 용량을 의미한다.
- 로그 데이타량 : 생성되는 리두(redo) 로그 데이타 용량을 의미한다.

튜닝가능한 자원 추천 방법에서 인스턴스 속성 구조는 군집 기법과 분류 기법에서 약간 차이가 있다. 군집 기법은 <표 2>와 같이 8개의 인스턴스 속성과 측정된 수행시간으로 구성된 인스턴스의 속성 구조를 보여준다. 분류 기법은 <표 2>에 군집 기법의 결과인 추천 정보(클래스)를 매핑시켜 재구축된 인스턴스 속성 구조를 사용하며 <표 3>에서 보인다.

〈표 2〉 인스턴스의 속성 구조(scheme)

데이타 변경률	버퍼 적중률	공유 메모리 적중률	메모리 파싱율	디스크 쓰기량	버퍼 읽기량	버퍼 쓰기량	리두 로그량	수행 시간

〈표 3〉 인스턴스의 속성 구조(scheme)

데이타 변경률	버퍼 적중률	공유 메모리 적중률	메모리 파싱율	디스크 쓰기량	버퍼 읽기량	버퍼 쓰기량	리두 로그량	수행 시간	선택정보 (클래스)

TPC-W의 인스턴스 수집 환경은 데이타베이스, 웹/응용프로그램, 이미지 서버를 별개의 시스템에 장착하였다. 서버에 사용되는 데이타베이스는 오라클 9i버전을 이용하였으며 웹 서버는 웹로직(WebLogic)과 아파치 웹 서버를 이용하였다. TPC-C의 인스턴스 수집 환경은 클라이언트와 데이타베이스 시스템(오라클 9i)이 동일 기계에서 구동된다. 또한 실험 기간 동안 시스템에 최대 부하를 주기 위해, TPC-C는 웨어하우스의 수를 15개로 설정하였고, TPC-W는 EB의 수를 90개로 설정하였다. 수집된 인스턴스 집합은 TPC-C와 TPC-W 환경에서 총 1600여 개를 수집하였다. 또한 각 인스턴스는 부가적으로 워크로드 종류, 자원 종류, 자원의 크기 정보를 식별할 수 있도록 하였다.

마이닝 기법이 적용되는 방법을 위해 마이닝 도구는 공개용 소스인 WEKA(버전3.4)를 이용하였다[Weka]. WEKA는 널리 알려져 있고 대중적인 군집 기법 알고리즘과 분류 기법 알고리즘들을 제공한다. 마이닝의 분류 기법으로 생성된 모델의 검증은 교차 검증 기법을 이용하였으며 널리 사용되고 있는 10교차 검증 기법을 적용하였다. 분류 기법은 WEKA에서 제공하는 J48 알고리즘을 이용하였다. J48은 C4.5 알고리즘을 개선한 알고리즘으로 수치형(numeric) 데이타 타입뿐만 아니라 명목형(nominal) 데이타 타입도 처리하며 의사 결정 트리로 결과 모델이 생성되는 알고리즘이다. 군집 기법은 WEAK에서 제공하는 EM 알고리즘을 이용하였다. 사용된 알고리즘에 대한 구체적인 설명과 동작방식의 원리는 본 논문의 범위를 넘어서 설명하지 않는다.

4. 튜닝가능한 자원선택 방법의 수행 결과

본 절은 자원선택 방법론에 따라 데이터베이스 시스템 환경에서 인스턴스를 수집하고 모델을 구축하여 검증을 수행한다. <그림 5>는 5개의 군집 개수로 설정된 EM 알고리즘의 결과를 클래스로 이용하는 J48 알고리즘 추천 모델을 보여준다. J48 알고리즘은 속성을 값을 비교하여 해당 추천 정보로 분류되는 의사결정 트리로 추천 모델을 구축한다. 모델에서는 다섯 개의 클래스로 분류되기 위해 9개의 속성 값 중 5개의 속성이 이용되었으며 값의 범위와 조건에 따라 클래스로 분류된다. 예를 들어, class[0]으로 분류되는 경우는 공유 메모리 적중률이 80.93이상이고 버퍼 적중률이 70.64이하이면서 53.42 초과인 인스턴스들이 분류된다.

〈표 4〉 EM 알고리즘을 적용한 선택 모델의 검증 결과

분류 알고리즘	군집 개수	정확률
J48	2	99.94%
	5	97.8%
	10	97.1%
	15	97.7%
	19	96.6%

```
DBLHR <= 80.93
|  DBNCHECK <= 88057
|  |  DBRDS <= 154738896: class[2]
|  |  DBRDS > 154738896
|  |  |  DBNCHECK <= 80734
|  |  |  |  ELAPSEDTIME <= 35: class[2]
|  |  |  |  ELAPSEDTIME > 35: class[1]
|  |  |  DBNCHECK > 80734: class[1]
|  DBNCHECK > 88057: class[1]
DBLHR > 80.93
|  DBBHR <= 70.64
|  |  DBBHR <= 53.42: class[0]
|  |  DBBHR > 53.42
|  |  |  ELAPSEDTIME <= 10: class[3]
|  |  |  ELAPSEDTIME > 10: class[0]
|  DBBHR > 70.64
|  |  ELAPSEDTIME <= 40: class[3]
|  |  ELAPSEDTIME > 40: class[4]
```

DBLHR : 공유 메모리 적중률
DBNCHEK : 디스크 쓰기량
DBRDS : 로그 데이터 량
DBBHR : 데이터 버퍼 적중률
ELAPSEDTIME : 수행 시간

〈그림 5〉 J48 알고리즘을 사용한 추천 모델의 예

<표 4>는 군집기법으로 EM 알고리즘과 분류 기법으로 J48 알고리즘을 사용한 추천 모델의 검증 결과를 보인다. 군집은 2개부터 총 19개까지 생성하였고 다섯 개의 군집 개수 단위로 보인다. 생성된 모든 추천 모델은 정확률이 95% 이상 보였다.

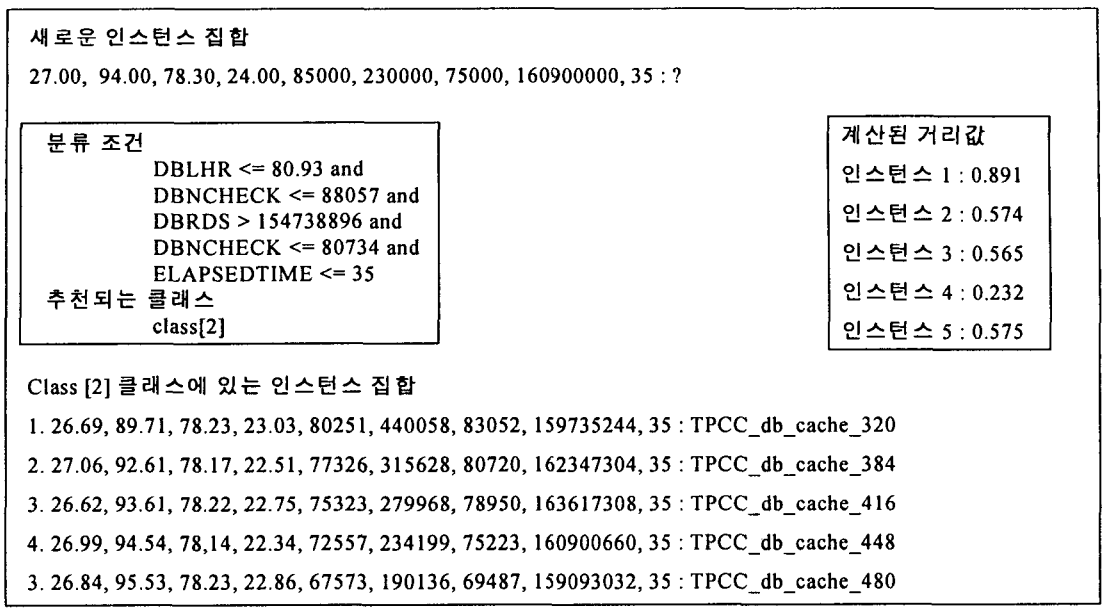
워크로드에 따른 자원 선택이 분류 목적이어서 2개의 워크로드(TPC-C, TPC-W)와 4개의 자원이므로 총 8개의 클래스가 생성된다고 가정했을 때 분류 기법만을 적용한 추천모델의 검증결과는 <표 5>와 같다. 분류기법만을 적용한 결과는 추천 정보의 단순화를 위해 군집 기법을 적용한 선택모델의 검증결과보다 정확률이 낮아서 선택모델로서 사용이 부적합함을 알 수 있었다.

〈표 5〉 분류기법만을 적용한 선택 모델의 검증 결과

분류 알고리즘	클래스 개수	정확률
J48	8	66.4%

튜닝가능한 자원의 선택은 클래스 단위의 추천 정보를 인스턴스 단위의 추천으로 변환하고 추천된 정보에 대응하는 자원의 사용 형태를 얻는 작업을 의미한다. 새로운 인스턴스에 대해 자원 추천을 수행할 때 인스턴스는 구축된 추천 모델의 입력으로 활용되고 추천 모델에 의해 특정 클래스로 분류된다. 분류된 클래스는 유사한 속성의 값을 지닌 인스턴스들의 집합이므로 입력된 인스턴스와 가장 근접한 인스턴스를 선택하는 작업을 수행하고 근접한 인스턴스가 측정된 데이터베이스 시스템의 자원 상태를 획득한다.

<그림 6>은 인스턴스를 입력받아 <그림 5>의 추천 모델에 의해 클래스를 분류하고 변이 계수를 적용하여 인스턴스를 선별하여 인스턴스에 대응된 데이터베이스 환경 정보에 의해 자원을 추천하는 예를 보여준다. 입력된 인스턴스는 추천 모델의 분류 조건에 의해 class[2]로 분류된다. class[2]와 분류 조건에 부합되는 인스턴스 집합은 다섯 개이다. 다섯개의 인스턴스들은 입력된 인스턴스와 변이 계수 거리 공식에 의해 거리값을 계산하고 가장 낮은 거리값을 산출하는 인스턴스 4를 선별한다. 인스턴스 4가 TPC-C 워크로드에서 데이타 버퍼가 448MB에서 측정되었으므로, 입력된 인스턴스의 자원 추



〈그림 6〉 자원 추천의 예

천은 데이타 버퍼의 448MB로 수행되며, 자원 선택 정확률은 97.8%이다.

5. 결론 및 향후 계획

본 논문은 워크로드 특성에 따른 데이타베이스 상태를 판단하고 튜닝가능한 자원을 선택하기 위해 방법론을 제안하였다. 튜닝가능한 자원 선택 방법론은 군집 기법을 이용해서 추천 정보를 단순화하고 분류 기법과 추천 제약 조건을 통해 자원을 추천한다. 방법론의 적합성 검증은 인스턴스를 수집하고 분류기법만을 이용한 추천 모델 구축 방법과 비교하여 수행하였다. 인스턴스 수집은 TPC-C와 TPC-W 환경에서 총 120회의 자원 변경 실험을 통해 8개의 워크로드 속성이 포함된 1600여 개의 인스턴스를 수집하였다. 마이닝 기법이 필요한 방법론을 위해 본 연구는 공개용 소프트웨어인 WEKA에서 제공하는 분류 기법 알고리즘(J48)과 군집 기법 알고리즘(EM)을 이용하였다.

분류 기법만을 이용하여 추천 모델을 구축하는 방법은 TPC-C와 TPC-W환경에서 수집되는 인스턴스 특성을 식별하여 워크로드 추천을 성공적으로 수행할 수 있었다. 그러나 워크로드 특성에 따른 자원 추천과 워크로드 특성에 따른 자원 크기의 추천은 인스턴스 값들이 유사하게 기록되기 때문에 구축된 모델의 검증에서 정확률이 낮았다. 반면에 본 논문의 방법은 추천 정보의 단순화를 위해 군집 알고리즘을 이용하였다. EM 알고리즘은 19개의 군집 개수까지 생성하여 적합한 군집의 개수를 탐색하였고. 추천 모델은 생성된 모든 군집의 개수에 추천을 수행하기 위해 분류 알고리즘(J48)을 적용하여 생성하고 10교차 검증 기법에 의하여 정확률과 에러율을 산출하였다. 생성된 모든 추천 모델의 정확률이 90% 이상을 보였다.

본 연구의 결과인 방법론은 효율적인 데이타베이스 관리와 자율적 튜닝을 위해 효과적으로 사용될 수 있다. 본 연구가 제시한 방법론은 워크로드에 따른 데이타베이스 시스템의 상태를

판단하여 적합한 특정 자원의 크기를 추천할 수 있었으며, 사람이 수행했던 분석 및 판단에 대한 개입을 감소시켜 자율적 데이타베이스 시스템의 분류 중 자체 검사에 대한 초석을 제시하였다. 향후 연구로는 데이타베이스 시스템의 상태 정보를 효과적으로 처리하기 위해 적절한 군집 개수의 탐색 방법과 알고리즘 개발이 요구되며, 수립된 방법론을 적용해서 상이한 워크로드 환경에서 자동으로 자원의 크기를 조절하는 자원 조절 도구를 개발하려 한다.

참 고 문 헌

[1] Aboylnaga, A. and Chaudhuri, S, "Self-Tuning Histograms without Looking at Data", *ACM SIGMOD Conference*, 1999, pp. 181-192.

[2] Argrawal, S., Chaudhuri, S., and V. Narassayya, "Automated Selection of Materialized Views and Indexes of SQL Databases", *VLDB Conference*, 2000, pp. 496-505.

[3] Brown, K. P., Carey, M. J., and Livny, M., "Goal-Oritented Buffer Management Revisited", *ACM SIGMOD Conference*, 1996, pp. 353-364.

[4] Benoit, D., G., "Automated Diagnosis and Control of DBMS Resource", *EDBT Conference*, Ph.D workshop, 2000.

[5] Brown, K. P., Mehta, M., Carey, M. J., and Livny, M., "Towards Automated Performance Tuning for Complext Workloads", *VLDB Conference*, 1994, pp. 72-84.

[6] Chaudhuri, S., and Narassayya, R., "AutoAdmin 'What-if' Index Analysis Utility", *ACM SIGMOD Conference*, 1998, pp. 367-378.

[7] Chaudhuri, S. and Narassayya, R., "Automating Statistics Management for Query Optimizers", *ICDE Conference*, 2000, pp. 339-348.

[8] Chaudhuri, S. and Weikum, G., "Rethingking Database System Architecture : Towards a Self-Tuning RISC-Style Database System", *VLDB Conference*, 2000, pp. 1-10.

[9] Cyran, M., "Oracle 9i : Database Performance Guide and Reference, Release I (9.0.1)", Oracle Corporation, 2001.

[10] Elnaffar, S., "A Methodology for Auto-Recognizing DBMS Workloads", *CACON*, 2002.

[11] Elnaffar, S., Martin, P., and Horman, R., "Automatically Classifying Database Workloads", *CKIM Conference*, 2002, pp. 622-624.

[12] Elnaffar, S., Powely, W., Benoit, D., and Martin, P., "Today's DBMSs : How Automatic are They?", *DEXA Workshop*, 2008, pp. 651-655.

[13] Ganek, A. and Corbi, T., "The Dawning of the Autonomic Computing Era", *IBM Systems Journal*, Vol. 42, No. 1, 2003, pp. 5-13.

[14] Goebel, M. and Gruenwald, L., "A Survey of Data Mining and Knowledge Discovery Software Tools", *ACM SIGKDD Explorations*, Vol. 1, No. 1, 1998, pp. 20-33.

[15] Martin, P., Li, H. Y., Zheng, M., Romanual, K., and Powley, W., "Dynamic Reconfiguration Algorithm : Dynamically Tuning Multiple Buffer Pools", *DEXA Conference*, 2000.

[16] Martin, P., Powley, W., Li, H. Y., and

Romanula, K., "Managing Database Server Performance to Meet QoS Requirements in Electronic Commerce Systems", *International Journal on Digital Libraries*, Vol. 3, No. 4, 2002, pp. 316-324.

[17] Morals, T. and Lorentz, D., "Oracle 9i : Database Reference, Release I (9.0.1)", *Oracle Corporation*, 2001.

[18] "TPC Benchmark C Specification(Revision 5.0)", http://www.tpc.org/tpcc/default.asp, 2001.

[19] "TPC Bemchmark W(Web Commerce) Specification(version 1.8)", http://www.tpc.org/tpcw/default.asp, 2002.

[20] WEKA Public Machine Learning(Mining) Software tool(version 3.4) Homepages, http://www.cs.waikato.ac.nz/weka/index.html.

[21] Weikum, G., Koning, A. C., Krasis, A., and Sinnewell, M., "Towards Self-Tuning Memory Management for Data Servers", *Bulletin of Technical Committee on Data Engineering*, Vol. 22, No. 2, pp. 3-11.

저자소개

김 혜 숙

숭실대학교 전산계산학과에서 학사, 성균관대학교에서 경영학석사, 전북대학교 대학원에서 컴퓨터공학전공으로 박사학위를 취득하였다. 현재 숭실대학교 전산원 멀티미디어학과 교수로 재직 중이며, 주요 관심분야는 Image Processing, Multimedia Database, Database Modeling등 이다.

오 정 석

숭실대학교 컴퓨터학과에서 석사 및 박사학위를 취득하였으며, (주)틸론 기술연구소 책임연구원을 역임하였고, 현재 한국가스안전공사 가스안전연구원 선임연구원으로 재직 중이다. 주요 관심분야는 Autonomous DBMS Tuning, Database Workload, Analysis, Ubiquitous Information and Industrial Technologies이다.

◈ 이 논문은 2007년 11월 03일 접수하여 1차 수정을 거쳐 2008년 03월 19일 게재확정되었습니다.