

효율적인 페이지 사용을 위한 UNISYS NDB 의 CALC 배치 방식 개선 알고리즘

조태환[†], 김영갑[‡]

[†] 고려대학교 컴퓨터과학기술대학원

[‡] 고려대학교 컴퓨터학과 소프트웨어시스템 연구실

e-mail : jthwan@chb.co.kr, ygkim@software.korea.ac.kr

Improved Algorithm of CALC Method in UNISYS NDB for Efficient Page Allocation

Tae-Hwan Jo[†], Young-Gab Kim[‡]

[†] Graduate School of Computer Science & Technology, Korea University

[‡] Software System Lab., Computer Science & Engineering, Korea University

요 약

현재 UNISYS 네트워크 데이터베이스(Network Data Base, NDB)에서는 엔트리 레코드(entry record)를 배치시키기 위한 방법 중 CALC방식의 DMSCALC 알고리즘을 사용하고 있다. 그러나 이 알고리즘의 특성상 하위 레코드가 과도하게 발생하는 업무의 경우 동일 기본저장 페이지(primary page)에 할당되는 다 엔트리 레코드의 저장 영역에 대한 제한을 유발시켜 오우버플로워 페이지(overflow page)로 이동되고 이에 따라 성능 저하를 가져온다. 본 논문에서는 나뉜셈법을 따르는 DMSCALC 알고리즘을 분석 연구하고, 엔트리 레코드가 저장되는 기본저장 페이지의 산출 알고리즘을 개선하여 기본저장 페이지 간 간격을 확보하였으며 개선 전 알고리즘과 비교할 때 기본저장 페이지 사용 율의 경우 최소 22%, 최대 41%에 이르는 성능 향상을 나타내었다. 또한 오우버플로워 페이지 발생률의 경우에도 최소 47%, 최대 67%의 감소 효과를 나타내었다.

1. 서론

UNISYS 네트워크 데이터베이스(Network Data Base, NDB)에서 엔트리 레코드(entry record)를 배치시키는 방법에는 DIRECT 방식, INDEX SEQUENTIAL 방식, CALC 방식으로 나누어지며, 이러한 배치 방식들은 최적화된 엔트리 레코드 배치를 위하여 업무 성격에 따라 가장 적합한 방법으로 선택된다. DIRECT 방식은 가장 빠른 Access를 구현하며 물리적으로 근접배치를 효과적으로 가능케 하고, 예측된 입력 키(input key)를 사용할 때 적합하다. INDEX SEQUENTIAL 방식은 순차적 프로세스(sequential processing)에 유리하며 적은 양의 임의적 접근(random access)에 적합하다. CALC 방식은 앞의 두 가지 배치 방식이 일정 수준의 예측 가능한 입력 키가 제시된다는 것과는 달리 예측이 어려운 입력 키가 제시되는 경우의 업무에 적합하게 사용할 수 있다. 즉 엔트리 레코드 배치에 대한 사용자 관여는 허용되지 않으며 데이터베이스 시스템에서 엔트리 레코드 배치를 전적으로 관장한다.

현재 UNISYS 네트워크 데이터베이스를 사용하는 국내 기관으로는 금융기관 중 국내에서 가장 거래량이 많은 농협은행을 포함하여 조흥은행, 신한은행, 수협은행과 한국증권거래소, 은행연합회 등이 대표 할 수 있다. 금융업무는 발생 거래에 대하여 장기간 유지 관리와 사용자의 조회 요구에 즉시 응대 가능할 수 있는 시스템이 요구되며, 이러한 요구사항은 하나의 엔트리 레코드에 수 개에서 수천 개에

이르는 하위 레코드가 생성되는 요인을 제공한다. 동 레코드 생성 패턴을 기반으로 한 CALC 방식의 사용은 실 업무에서 오우버플로워 페이지(overflow page)의 과다 생성을 유발하여, I/O 성능 저하, 영업점 서비스 지연, 시스템 가용성 저하, 시스템 증설 요인 제공 등 다양한 문제점을 나타낸다. 이때 사용되는 CALC 방식의 엔트리 레코드 배치 알고리즘은 DMSCALC 알고리즘이 사용되며, 이는 유사한 입력 키를 가진 엔트리 레코드들의 근접 배치가 필요한 업무에 적합하다. 보유 데이터베이스 중 적게는 35%, 많게는 60%의 DMSCALC 알고리즘을 사용하는 상기 각 기관에서는 위와 같은 문제점에 대한 해결을 위하여 엔트리 레코드 배치에 필요한 항목 값만을 변화 시켜 최적의 레코드 배치 모형을 구하고 있으나 근본적인 DMSCALC 알고리즘에 대한 개선이 선행되지 않는 한 문제점 해결은 어렵다.

따라서 본 연구에서는 DMSCALC 방식의 근본적인 알고리즘을 개선하여 오우버플로워 페이지의 과다 생성을 최소화 하고, 시스템 성능 향상을 높여 고품질 영업점 서비스 제공과 적정한 경영비용 집행이[1] 가능토록 한다.

2. DMSCALC 알고리즘 기법

UNISYS 네트워크 데이터베이스의 CALC 배치 방식에서의 엔트리 레코드 배치 알고리즘은 DMSCALC 방식과 RANDEENTIAL 방식이 제공된다[2]. 이러한 알고리즘을 수행하게 되면 결과로 엔트리 레코드가 어느 위치에 저장 되

어야 하는지 기본저장 페이지(primary page) 번호와 Calc Chain Header 번호를 반환하게 된다. 여기서 페이지 번호는 일반적으로 블록의 개념을 가지며, Calc Chain Header 번호는 할당된 기본저장 페이지 내에서 엔트리 레코드들에 대한 중복을 피하기 위하여 그룹화를 구성하는 역할을 수행하게 된다.

RANDENTIAL 방식은 입력 키를 단순한 문자열로 삼아 비수학적 수준으로 취급하는 알고리즘이다. 이러한 RANDENTIAL 방식과는 다르게 DMSCALC 방식은 나눗셈법[3]을 이용하는데, 주어지는 입력 키의 값을 2진수 데이터로 삼아 저장 가능한 영역의 총수로 나눗셈을 수행하고, 그 나머지를 출력 파라미터로 하는 방법이다.

이 방식은 네트워크 데이터베이스의 특성과 은행업무의 데이터 관리 특성상 유사성이 있는 엔트리 레코드들을 근접 배치 시켜 온라인 거래, 온라인 일괄 조회 등 실시간 요구에 효율적이고, 신속하게 대응하게 하는 방식으로 알고리즘은 표 1 과 같다.

표1 DMSCALC 알고리즘 기법

단계1) $c = a \times b$
단계2) $k/c = \text{몫} + e/c (0 \leq e < c)$
단계3) $e/b = f + g/b (0 \leq g < b, 0 \leq f < a)$
단계4) 페이지 번호 = $f + 1$
단계5) Calc Chain Header 번호 = g

위 표 1 을 단계별로 서술하면 다음과 같다.

- 단계 1) 영역 내 기본저장 페이지 수(a)와 페이지 내 Calc Chain Header 수(b)를 가지고 저장 가능한 영역 수(c)를 구한다
- 단계 2) 입력 키(k)를 이진수를 변환한 후 정수화 하고 저장 가능한 영역 수(c)로 나누고 나머지(e)를 구한다
- 단계 3) 단계 2)에서 구해진 나머지(e)를 페이지 내 Calc Chain Header 수(b)로 나누어 몫(f)과 나머지(g)를 구한다
- 단계 4) 단계 3)에서 구해진 몫(f)에 1 을 더하여 저장 될 기본저장 페이지를 결정한다.
- 단계 5) 나머지(g)는 Calc Chain Header 번호를 결정한다.

개선 전 알고리즘은 엔트리 레코드에 소속된 하위 레코드가 과다한 경우 동일 기본저장 페이지로 할당되는 타 엔트리 레코드의 저장 영역에 대한 제한으로 오우버플로우 페이지로의 이동에 따른 성능 저하를 가져온다.

3. 개선된 DMSCALC 알고리즘 기법

개선 전 알고리즘의 문제점인 과다한 하위 레코드 생성에 따른 타 엔트리 레코드의 오우버플로우 페이지 이동은 할당된 기본저장 페이지에 선 저장된 엔트리 레코드 및 하위 레코드의 저장 가능 영역에 대한 선 점유 현상에 기인하고, 그 원인으로 기본저장 페이지번호 산출 시 엔트리 레코드간의 할당된 페이지 간격(page gap)이 작다는 것이다.

이와 같은 문제점을 해결하고자 개선된 알고리즘에서는 기본저장 페이지 산출 시 하위 레코드의 과다 생성을 대비

한 엔트리 레코드간 페이지 간격을 일정부분 확보할 수 있도록 제시 하였으며, 알고리즘은 표 2 와 같다.

표2 개선된 DMSCALC 알고리즘 기법

단계1) $c = a \times b$
단계2) $k/c = \text{몫} + e/c (0 \leq e < c)$
단계3) $e/b = f + g/b (0 \leq g < b, 0 \leq f < a)$
단계4) Calc Chain Header 번호 = g
단계5) $h = f + e$
단계6) $h/c = \text{몫} + i/c (0 < i < a)$
단계7) 페이지 번호 = i

위 표 2 를 단계별로 서술하면 다음과 같다.

- 단계 1) 영역 내 기본저장 페이지 수(a)와 페이지 내 Calc Chain Header 수(b)를 가지고 저장 가능한 영역 수(c)를 구한다
- 단계 2) 입력 키(k)를 이진수를 변환한 후 정수화 하고 저장 가능한 영역 수(c)로 나누어 나머지(e)를 구한다
- 단계 3) 단계 2)에서 구해진 나머지(e)를 페이지 내 Calc Chain Header 수(b)로 나누어 몫(f)과 나머지(g)를 구한다
- 단계 4) 단계 3)에서 구해진 나머지(g)를 Calc Chain Header 번호로 결정한다.
- 단계 5) 단계 3)에서 구해진 몫(f)에 단계 2)에서 구해진 나머지(e)를 더하여 최적화 전 페이지 번호(h)를 구한다
- 단계 6) 단계 5)에서 구해진 최적화 전 페이지 번호(h)를 저장 가능한 영역 수(c)로 나누어 나머지(i)를 구한다.
- 단계 7) 구해진 나머지(i)로 기본저장 페이지를 결정한다.

위 개선된 알고리즘은 두 개의 기본저장 페이지 마다 하나의 여유 페이지를 할당하여 과다한 하위 레코드 생성을 대비하고, 엔트리 레코드간 근접배치 기준을 정확히 준수하여, 최대한 영역 낭비가 발생치 않도록 하였으며, 물리적으로 보조기억장치의 효과적 사용과 신속한 접근 원칙[4]을 가능하게 한다

즉, 제시한 알고리즘은 기본저장 페이지 산출에 있어 좀더 유연하게 레코드가 저장 될 수 있도록 기존 알고리즘의 최종 페이지 산출 공식에 입력 키를 저장 가능한 페이지 수로 나눈 나머지를 가산하여 최적화된 페이지 번호 산출이 가능토록 함으로써 페이지 간 간격을 유지토록 하였다.

4. 분석 및 평가

개선된 알고리즘을 평가하기 위하여 표 3 과 같이 실 데이터(real data)를 사용하였다.

표3 평가 대상 업무 기초 정보

업무명	엔트리 레코드 수	하위 레코드 수	SCHEMA 등록 정보	
			기본저장 페이지 수	Calc Chain Header 수
A업무	90,129	1,527,029	80,000	3
B업무	44,822	982,881	80,000	1
C업무	82,733	2,322,007	50,000	4
D업무	62,770	897,119	70,000	5
E업무	161,15	1,854,408	100,00	7

알고리즘의 평가는 다음의 기본 규칙을 정하여 실시 하였다.

- 첫째, 개선 전 알고리즘과 개선 후 알고리즘 비교
- 둘째, 기본저장 페이지의 Load Factor를 100%로 하여 실질적인 기본저장 페이지의 사용율을 측정
- 셋째, DARP(Dynamic Area Record Placement) 방식을 준용하여 기본저장 페이지당 전용의 오버플로워 페이지를 할당
- 넷째, 원장 재편성을 통한 평가 수행

먼저 개선 전 알고리즘과 개선 후 알고리즘을 평가하기 위하여 원장 재편성을 수행하였고, 원장 상황을 진단하기 위하여 업체에서 제공하는 데이터베이스 점검 툴(tool)[5]을 이용하여 결과를 추출하였으며 그 결과는 그림 1, 그림 2, 그림 3을 통하여 확인 할 수 있다.

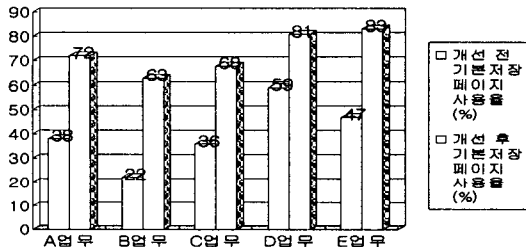


그림1 기본저장 페이지 사용율 비교

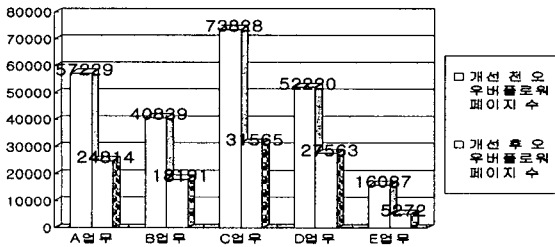


그림2 오버플로워 페이지 수 비교

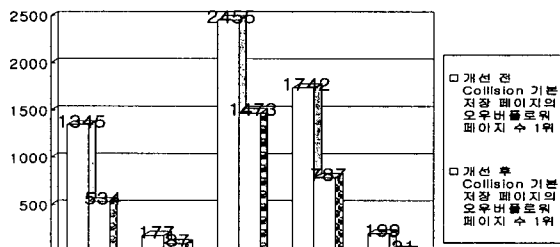


그림3 Collision 기본저장 페이지의 오버플로워 페이지 수 1위 비교

위 평가 결과를 살펴보면 DMSCALC 알고리즘은 원장 최적화를 위한 원장 재편성 수행 시 엔트리 레코드와 하위 레코드를 최대한 근접배치 시키며 하위 레코드가 해당 기본

저장 페이지에 저장 될 수 없는 경우 다음 기본저장 페이지에 저장 된다는 것을 나타내고 있다. 따라서 다음 번에 생성되는 엔트리 레코드의 저장 할 위치는 이전 엔트리 레코드의 하위 레코드 생성 개수에 영향을 받게 된다.

그림 1 기본저장 페이지 사용 율 비교를 살펴보면 개선 후 알고리즘을 수행 한 결과가 높게 나타났으며, 그림 2 오버플로워 페이지수 비교에서도 개선 후 알고리즘 수행 결과가 현저히 적게 나타났다. 또한 그림 3 Collision 기본저장 페이지의 오버플로워 페이지 수 1위 비교 결과도 개선 후 알고리즘을 수행 한 결과가 우수한 것으로 나타나는바, 이는 기본저장 페이지 간 여유 페이지를 할당하여 이전 엔트리 레코드의 하위 레코드 개수에 대한 영향을 감소시킬 경우 기본저장 페이지의 효율적 사용과 오버플로워 페이지로의 이동이 감소 된다는 것을 증명하고 있다.

5. 결론

본 논문에서는 기존의 UNISYS 네트워크 데이터베이스의 엔트리 레코드 배치 방식 중 CALC 배치방식의 DMSCALC 알고리즘이 엔트리 레코드의 기본저장 페이지 할당에 있어 밀집된 형태의 저장을 유도하게 되고 이로 인한 과도한 오버플로워 페이지 생성과 그에 따른 I/O 성능 저하, 영업점 서비스 지연, 시스템 가용성 저하, 시스템 증설 요인 제공 등 문제점 해결을 위하여, 개선된 DMSCALC 알고리즘을 제시하였다.

제시된 알고리즘을 증명하기 위하여 대량의 하위 레코드를 보유하고 있는 5개 업무에 대하여 기존 DMSCALC 알고리즘을 구현하고, 개선된 DMSCALC 알고리즘을 구현 하여 결과를 비교 하였으며, 그 결과 기본저장 페이지 사용율의 경우 최소 22%, 최대 41%에 이르는 성능 향상을 나타내었고, 오버플로워 페이지 발생 율의 경우에는 최소 47%, 최대 67%가 감소하는 향상된 성능을 테스트를 통해 확인 할 수 있었다.

향후에는 개선된 DMSCALC 알고리즘을 업무별 특성에 맞춰 유연하게 페이지 간 간격을 조정할 수 있는 진화된 룰(rule) 개발에 초점을 맞추고, 더불어 UNISYS 에서 제공되는 데이터 베이스 관련 유틸리티 활용에 접목할 수 있는 방안 에 대한 연구가 필요하다.

참고문헌

- [1] 금융감독원 " 감사업무편람(4) - 정보기술(IT) 부문 감사업무" 6200 전산경영 (2000)
- [2] Unisys Corporation in Korea. " Data Base Management System DMS1100 Structure Analysis" ver. ue-014r2 pp. 48-52, 155-160 (1998)
- [3] 황중선, 정영식 " C 언어로 설명한 알고리즘" pp. 174-175 (1996)
- [4] Hector Garcia-Molina, Jeffrey D. Ullman and Jennifer Widom " Database Systems The Complete Book (International Edition)" Chapter 11. Data Storage (2001)
- [5] Unisys Corporation in Korea. " DBE 6R1C 1100/90 L39 (RELEASE)" pp. 101-108 (2000)