

MP2P OLAP: Managed Peer-to-Peer OLAP 을 위한 효율적인 인덱스 구조 및 시스템 구성

김훈동[○] 주길홍 이원석

(주)월비솔루션 기술연구소, 경인교육대학교 컴퓨터교육과, 연세대학교 컴퓨터과학과

spiccato@empal.com, khjoo@ginue.ac.kr, leewo@database.yonsei.ac.kr

MP2P_OLAP: Efficient index structure and system architecture for Managed Peer-to-Peer OLAP

Hoondong Kim[○] Kilhong Joo Wonsuk Lee

Technical Research Center, Will-Be Solution, Inc. Dept. of Computer Education, Gyeongin National University of Education. Dept. of Computer Science, Yonsei University.

1. 서론

OLAP 질의는 데이터 양과 질의의 특성상 질의 처리 시간이 수분에서 수십 분이 걸리면서도 사용자가 이를 감내하며 사용하고 있다. MOLAP 은 빠른 응답을 보이지만, 디스크 공간이 한정되어 모든 조합의 모든 셀을 실체화(Materialized) 하는 데에는 물리적인 한계가 있어, 대규모 데이터의 분석용으로는 적당하지 않다. 뿐만 아니라, 최신 분산 OLAP 연구의 큰 주류인 DHT기반 P2P는 structured P2P 특성상 다차원 범위 질의에 한계가 있었고, Grid OLAP은 인접성 및 시계열 고려가 없었으며, 질의 자체의 서브 셀 조회 알고리즘 연구에 치중되어 있었다. 본 연구는 사용자의 조회 결과가 시계열적 특성으로 미래에 재 사용 가능함에 초점을 두어 P2P 방식을 제안하였으며, 빠른 질의 결과 및 다차원 범위질의를 위해 다단계 Hybrid P2P방식으로 인덱스가 중앙 관리되는 Managed P2P OLAP(MP2P OLAP)을 제안하였다. 이를 위해 큐브 위상관계트리와 인접성 2차원 Quadtree에 시계열 Interval-tree를 접목하여 인덱스를 구축하였으며, Quadtree에 Balance가 유지되도록 선처리 알고리즘을 제안하였다.

2. 본론

Google의 Bigtable은 B+tree와 유사한 인덱스 개념을 다단계의 관리 가능한 분산 시스템으로 확장하여서 높은 효과를 보였다. 본 연구는 이러한 분산 중앙 관리 인덱스 구조에 초점을 두어 OLAP에 다단계의 Hybrid P2P를 접목하였다. 효율적인 인덱스 관리가 제한된 형태로 OLAP에 P2P나 Grid를 도입한 개념은 이미 몇몇 유사(Dehne[1]) 연구가 진행 된 바 있으나, 본 논문에서는 추가로 기존 연구가 간과 하였던 아래와 같은 특성을 전제로 서버에서 범위 질의 및 시계열 관리가 가능한 보다 향상된 알고리즘을 제안하였다.

- (1) 성능 및 범위질의 가능여부 측면에서 기존(Vaisman[2]) DHT P2P 보다 중앙관리 P2P 가 유리하다.
- (2) 실체화 큐브는 위상관계가 존재하여, 서브 셀 포함관계 인덱스 고려가 필요하다.
- (3) P2P 특성상 인덱스에서 물리적 논리적 인접성 고려가 필요하다.
- (4) 데이터의 시계열적 특성을 감안, 과거 큐브 재활용을 위한 인덱스 메커니즘이 필요하다.
- (5) 클라이언트 노드는 보유 큐브에 대하여 자체 메모리 기반 질의 서버 역할을 할 수 있다.

제안하는 MP2P OLAP 아키텍처는 총 3개의 인덱스 Layer를 가지고 있다. 복수개의 Layer가 존재하는 이유는 서로 다른 여러 성능 개선 요소를 모두 해결하기 위함이며, 추후 서버 부하를 고려 하여 Layer 단위로 Hybrid P2P의 깊이를 확장할 수 있는 여지를 부여하기 위함이다. Layer별 특성은 아래 표 1과 같다.

표 1. MP2P OLAP 인덱스 Layer

인덱스 Layer	Layer 특성	기존 연구 간과 특성	역할
Layer1	복잡한 집계 데이터 큐브의 격자를 트리 형태로 정렬하고 있는 인덱스 구조.	(1), (2)	최적의 상위 큐브를 찾아 주는 역할
Layer2	- 논리적 물리적 인접성 정보를 2차원으로 한 Balanced Quadtree 인덱스.(Balance 알고리즘 제안) - 위 인접성 인덱스의 노드에 각 노드를 대표하는 시계열 집계 bit 범위를 Interval-tree 화 한 인덱스	(1), (3), (4)	물리적 인접하고, 논리적 활용도 높은 노드를 시계열적 재귀 호출 깊이가 깊어지지 않는 범위로 최적화 하여 조회 및 갱신 해줌.
Layer3	최 인접하고 시계열 범위를 만족하는 복수 노드가 서브셀 쿼리에 쿼리 서버 역할을 하는 인덱스	(5)	P2P 로 각 노드의 집계 큐브가 쿼리 서버 역할을 함.

큐브 집계단위(Granularity)를 한 단계 올려 큐브 사이즈를 최소화 하고, 기존 ROLAP 을 원천 데이터 화 하여

상세 뷰 요청은 drill through 기법으로 기존 ROLAP 에 요청하도록 하는 경우 상세 row 단위 큐브 전달은 피하면서 상세 뷰 조회도 제공 가능한 시스템을 제공 할 수 있다. 해당 특성을 이용하여 MP2P OLAP은 각각의 노드가 기 집계된 데이터큐브의 P2P 복제자 역할을 할 뿐 아니라, 서버 과생 큐브에 대한 질의를 처리 할 중간 하이브리드 서버 역할을 할 수 있도록 하였다. 또한, 서버 요청은 집계 존재하지 않는 영역으로 한정하고, 복수 클라이언트 노드가 비동기적으로 동시에 서로 데이터를 주고 받을 수 있도록 Legacy ROLAP 을 보완하도록 하였다. 특히 알고리즘 측면에서 물리적, 논리적 인접 성을 고려 하였고, 시 계열적인 특성을 이용하여 서버에 요청하는 데이터의 양을 최소화 하고, 과거의 캐시 데이터가 최대한 미래에도 활용 될 수 있도록 하는데 초점을 두었다. 해당 조건을 만족하는 제안하는 인덱스는 아래 표2와 같다.

표 2. MP2P OLAP 인덱스 알고리즘

인덱스 Layer	인덱스 특성	제안 알고리즘
Layer1	큐브 사이즈가 작은 상위 노드에서의 부분집합의 조회가 성능적으로 유리하므로 상위 노드를 큐브 사이즈가 작은 노드로 한정 하는 경우 트리 형태의 자료 구조	2 ⁿ 개의 큐브 포함관계 Network 격자에서 상위 노드 큐브 사이즈가 작은 경로를 선별한 Tree 구조
Layer2	물리적 인접성은 우편번호 등 1차원 정보임. 부서정보 등 논리적 인접성 + 물리적 인접성 2차원 자료구조에 대한 범위 질의 및 갱신 최적화 인덱스 (Balance 고려 필요)	그림 3과 재귀 결합을 위해 갱신이 빠르고, Split 이 없는 Quadtree가 유리
	위 Quadtree 의 단점인 Balance가 보장 안 되는 점을 보완하기 위한 메커니즘.	[그림 1] 그림 1을 Balanced 후 처리 한 Balanced Quadtree
	인접성을 고려하며, 시계열 범위 연산 시 노드의 Full Scan 을 막는 시계열 범위 Interval Tree 와의 재귀적 결합 인덱스	[그림 2] Balanced Quadtree 의 노드에 재귀적 범위 대표 값을 Interval Tree 화 하여 결합.
Layer3	위 Layer 에서 찾아진 재활용 가능 큐브가 서버 큐브 및 쿼리에 쿼리 서버로 반응 하기 위한 인덱스	VM 차원 메모리 기반 Query 메커니즘 이용.

[1차 축 크기 선처리]

[원본] MaxX:99 MinX:10
MaxY:6999999 MinY:100100



[선처리 후] Max:1000 Min:0
Max:1000 Min:0

$$f(x) = (x - \text{MinX}) * \text{MAX} / (\text{MaxX} - \text{MinX})$$

$$f(y) = (y - \text{MinY}) * \text{MAX} / (\text{MaxY} - \text{MinY})$$

[2차 Balance 선처리]

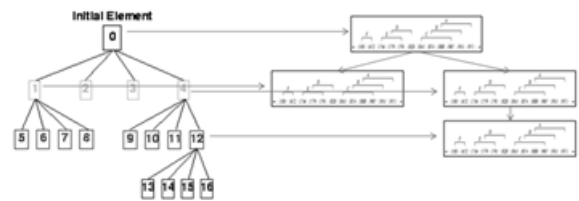
(구간처음 ~ 구간1/2 지점)

$$F'(x) = (\text{count}(\text{Min}, \text{Max}/2) / \text{count}(\text{Min}, \text{Max})) * \frac{1}{2} * f(x)$$

(구간1/2 지점 ~ 구간 끝)

$$F(x) = (\text{count}(\text{Max}/2, \text{Max}) / \text{count}(\text{Min}, \text{Max})) * \frac{1}{2} * f(x) + (1 - (\text{count}(\text{Max}/2, \text{Max}) / \text{count}(\text{Min}, \text{Max}))) * \frac{1}{2} * \text{Max}$$

[Depth 만큼 아래 루틴 재귀 호출]



[인접성 Quadtree]

[시계열 Interval tree]

Balanced Quadtree 인접성 인덱스와 Interval-tree 시계열 인덱스를 재귀적으로 통합한 경우의 시간 복잡도

- 시계열 비 인덱스 수 : M
- 한 하이퍼레카가 가지고 있는 평균 Interval 수가 M 의 1/100 개라고 가정 할 때
- 노드 개수 : N
- Threshold : B

시간복잡도:
- $O(\log_2 N/B + (\log_2 N/B - 1) * (\log_2 M/100))$

그림 1. Balanced Quadtree 후 처리 알고리즘

그림 2. Balanced Quadtree + Interval-Tree 재귀 결합

시물레이션 실험 통해 노드 수에 비례한 부하의 증가가 지수함수 양상을 보이지는 않음을 알 수 있었다. 이로 인하여, 한 개 인덱스에 많은 부하를 동시에 주어 성능 테스트를 하더라도 전체 시스템의 부하 테스트가 가능함을 유추할 수 있었다. 검색 성능은 일반적인 분산 OLAP 접근 방법보다 4800개 노드 이하에 한하여 비슷하거나 약간 개선된 성능을 보였고, 전 구간에 대하여 갱신 성능은 현격한 성능 우위를 확인할 수 있었다. 4800개 노드는 서버 역할을 하는 노드 개수로서 전체 노드의 일부에 해당하므로, 기업이 사용하는 분산 인덱스 서버로 실 세계에 적용 시 충분한 노드 개수라 할 수 있다.

3. 결론

본 연구는 ROLAP의 유연성 및 확장성에, MOLAP의 속도상의 장점을 제공하고자, 클라이언트 큐브 캐시를 관리 가능한 P2P로 공유하는 시스템과 그러한 시스템에 최적화된 인덱스 구조를 제안하였다. 클라우드 컴퓨팅 및 SOA 아키텍처가 각광을 받으면서, 분산 OLAP 의 가능성은 좀더 커지고 있는 추세이다. 본 연구는 그러한 분산환경에서 OLAP 큐브의 인덱스를 다루는 새로운 접근 방법을 제안하였다는 측면에서 의미 있는 연구였다 하겠다. 향후 시물레이션 실험이 아닌 실 운영 환경에서의 테스트를 통해 보다 실증적인 검증을 계속 할 예정이다.

참고문헌

[1] F. Dehne, M. Lawrence and A. Rau-Chaplin. "Cooperative caching for grid-enabled OLAP". *Int. J. Grid and Utility Computing*, 1(2): pp. 169. 2009..

[2] A. Vaisman, M. Espil and M. Parabela, "P2P OLAP: Data model, implementation and case study", *Information Systems*, 34(2): pp. 231-257, 2009.