결과분석

**[Table 1 : Evaluation Result : LUMB(times in second)]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Warm cache | | Cold cache | | Intermediaate Results | |
|  | q2 | q9 | q2 | q9 | q2 | q9 |
| RDF-3X | 26.5 | 37.0 | 28.7 | 43.4 | 424,747,108 | 659,968,158 |
| NO U-SIP | 22.9 | 31.9 | 25.0 | 38.4 | 424,785,330 | 662,615,314 |
| NO U-SIP RP-Rilter | 7.4 | 32.7 | 9.6 | 40.2 | 308,143,082 | 620,276,418 |
| RP-filter | 8.1 | 36.9 | 9.3 | 43.7 | 233,654,645 | 617,592,582 |

Table 1은 실행시간과 각 query의 중간결과의 수를 나타낸다. RDF-3X안에 U-SIP기술을 가지고 query를 측정했다. 그 결과, 실행 계획은 U-SIP의 영향으로 많은 해쉬조인이 포함되기 때문에 U-SIP은 효율적이지 못했다. RP-filter에서 Q2와 Q9은 다른 결과를 보여주고 있다. Q2는 Q9보다도 향상된 결과를 보여준다. 이 이유로는 Q2에서는 선택적인 경로패턴이 존재하기하고 Q9은 없으며, Q2의 중간결과는 Q9에 비해 훨씬 줄어들었기 때문이다. 또한 RP-filter를 사용할 때 query 시간이 길어지는 것을 볼 수 있다. 이 이유는 중간결과를 줄이는 이득보다도 훨씬 더 큰 오버헤드를 생성하기 때문이다.

**[Table 2 : Index statistics]**

|  |  |  |  |
| --- | --- | --- | --- |
|  | size | Number of pattern | Number of visits |
| YAGO2  RP-filter  RG-index | 341 MB  1.1 G | 486,508  82,534 | 486,508  416,497 |
| LUBM  RP-filter  RG-index | 1.4 G  2.4 G | 77  118 | 77  590 |
| SP2B  RP-filter  RG-index | 1.3 G  1.3 G | 68,277  32,436 | 68,277  149,812 |

조금 더 다양한 분석을 위해 세 가지 데이터 세트를 통해 비교 분석해 보았다.(Lehigh University Benchmark(LUBM), Yet Anothere Great Ontology 2(YAGO2), SPARQL Performance Benchmark(SP2B)). LUBM은 도메인이 대학인 벤치마크 데이터 세트이며, YAGO2는 Wikipedia, 2 WordNet 및 GeoNames에서 파생 된 지식 기반이며, SP2B는 DBLP 시나리오를 시뮬레이트하는 벤치마크이다. Table 2는 위의 세 가지 데이터 세트에서의 RG-indecs와 RP-filter 비교분석한 표이다. 차별 비율과 빈도를 사용하여 그래프 패턴을 제거하기 때문에 RG-index는 RP-filter보다 더 적은 패턴을 인덱스한다. 그러나 단일 그래프 패턴이 여러 개의 Visits을 가질 수 있기 때문에 RG-index의 Visits수는 그래프 패턴 수보다 크다.

**[Table 3 : Query statistics]**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Group | A | B | C | D | Total |
| Execution Time(ms) | 0~10 | 10~100 | 100~1000 | 1000~ |  |
| Count(YAGO2) | 638 | 126 | 12 | 0 | 1913 |
| Count(LUBM) | 0 | 5 | 15 | 51 | 71 |
| Count(SP2B) | 178 | 203 | 189 | 7 | 577 |

Table 3은 세 가지 데이터 세트에서의 쿼리 통계치를 보여주고 있다. 쿼리는 RDF-3X에서 실행 시간에 따라 나누어진다. YAGO2는 작고 많은 술어가 있기 때문에 짧은 실행 시간으로 많은 질의를 한다. 그러나 LUBM에는 적은 수의 술어와 많은 수의 트리플이 있기 때문에 쿼리를 평가하는 데 오랜 시간이 걸린다. SP2B는 YAGO2와 LUBM 사이의 중간 특성을 가지고 있다.

**[Table 4 : Query Execution Time(ms)]**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Group | A | B | C | D | Total |
| YAGO2  RDF-3X  RP-filter  RG-index | 5.09  4.22(17%)  3.56(30%) | 28.19  19.38(31%)  14.86(47%) | 122.5  59.91(51%)  43.58(64%) | N/A  N/A  N/A | 21.08  14.58(30%)  11.27(46%) |
| LUBM  RDF-3X  RP-filter  RG-index | N/A  N/A  N/A | 53  50(5%)  50(5%) | 540.8  479.6(11%)  477.2(11%) | 134.490  90.290(32%)  89.587(33%) | 114.385  76.808(32%)  76.209(33%) |
| SP2B  RDF-3X  RP-filter  RG-index | 2.65  2.37(10%)  2.32(12%) | 31.94  25.78(19%)  16.19(49%) | 238.47  168.19(29%)  103.60(56%) | 1361.42  547.28(59%)  99.14(92%) | 106.68  71.53(32%)  41.55(61%) |

Table 4은 평균 실행 시간을 보여준다. RP-filter 및 RG-index 모두 쿼리 성능을 RDF-3X보다 약 30%이상 향상시킨다. 또한 RG-index가 YAGO2 및 SP2B의 RP-filter보다 더 효과적임을 알 수 있다. LUBM에서 RP-filter 및 RG-index는 유사한 효과를 나타냅니다. 이는 LUBM이 상대적으로 구조화 된 데이터 모델을 가지므로, 트리플 필터링에 효과적인 그래프패턴이 소량으로 존재하기 때문이다. 또한 RG-index는 실행 시간이 긴 쿼리의 중간 결과가 더 많아서 효과적으로 줄일 수 있기 때문에 실행 시간이 긴 쿼리에 더 효과적이다.