저전력 정렬 알고리즘 설계

김동승*, 최성운**, 윤성로*
고려대학교 *전기전자전파공학부/ **모바일솔루션학과
e-mail: dkimku@korea.ac.kr

Design of Energy-Aware Sorting Algorithms

Dongseung Kim*, Sungwoon Choi**, Sungroh Yoon*
*School of Electrical Engineering/Dept. of Mobile Solutions, Korea University

요 약

저전력 운전을 위해 정렬 알고리즘을 대상으로 알고리즘 복잡도와 에너지 소모 관련성을 분석하고 전력 효율을 향상시키도록 알고리즘을 수정 개발하여 실험하였다. 얻어진 결과는 향후 그린컴퓨팅 실현에 활용하고자 한다.

1. 서론

컴퓨팅 분야에서도 에너지 사용량을 줄여 녹색 환경을 지키려는 노력의 일원으로 green computing이 강조되면서 종전의 최단시간에 주어진 계산을 완료하는 성능위주의 컴퓨팅 방식에서 저전력 컴퓨팅이 중요 설계 요구사항으로 등장하게 되었다 [1-3] 문헌 [4]에 따르면 CPU 외에도 DRAM이 전력소모의 비중이 크고, 데이터 이동/접근의 빈도를 낮추는 것은 프로세서에서 연산횟수를 줄이는 것보다 훨씬 에너지 절감효과가 큰 것으로 알려져 있다. 이를 감안해서 전력 절감 컴퓨팅을 위한 기초연구로 핵심 연산인 정렬 알고리즘에 대한 실행시간 및 에너지/파워 소모와 그 효율을 분석한 연구결과를 보고한다.

2. 기존 정렬 알고리즘의 에너지 소모 분석 실험

정렬(sort)은 컴퓨터 연산 중 기본적이고 핵심적인 알고리즘에 속한다. 주어진 자료를 key의 순서에 따라 배치하는 결과를 제공하는 소트는 다양한 방식과 고유한 연산 복잡도를 갖는다. 대부분의 소트의 기본연산은 비교(compare)와 자리바꿈(swap)으로 볼 수 있고, radix sort 등을 제외하고는 비교횟수로 알고리즘의 복잡도를 결정한다. 허나 green computing 측면에서 에너지 소모 면에서 정렬연산의 복잡도 해석에 재검토가 필요하다. 이에 성능이 뛰어나고 복잡도가 O(NlogN)로 동일한 퀵정렬, 머지정렬, 힙정렬 (quicksort, mergesort, heapsort)을 대표적으로 분석 실험하고 소요시간 및 소요 에너지를 측정하였다.

컴퓨터의 연산 전력모델은 크게 두 요소 즉 무부하전력(idling power)과 부하전력으로 구성된다. 부하전력은 가동 중 실연산 과정에서 추가되는 전력으로 프로세서, 캐쉬 메모리 RAM 접근, 버스 등 데이터 전송장치, 하드디스크 등 주변장치의 소요 전력을 통합한 값을 측정해서 환산한다.

다양한 CPU 플랫폼의 전력소모특성을 확인하기 위해 Intel의 고성능 마이크로프로세서부터 저전력용 넷북 컴퓨터까지 대표적인 CPU에서 실험하였고, 무작위로 생성된 정수를 10M개 ($M=2^{20}$)부터 주 메모리가 포화될 때 까지(160M) 실험하였다. 실험에는 저전력 노트북용에서부터 고사양 멀티코어 프로세서를 탑재한 경우를 포함시켰다. 또한 병렬처리에 의한 정렬 효과를 확인하기 위해 PSRS(parallel sort by regular sampling) [5]을 멀티코어 프로세서에서 실험하였다. 사용한 컴퓨터의 상세사양을 표1에 보인다.

그림1-3은 정렬을 각기 다른 프로세서에서 실행시킨 후 측정한 소요시간 및 에너지 효율을 그래프로 보인다. 데이터 규모가 증가하면 실행시간은 물론 소요에너지도 따라서 증가한다. 공통적으로 실행이 빠른 알고리즘이 에너지 소모량도 작다. 단 효율면에서는 사용한 CPU 타입에 따라 각기 다른 특성을 보인다. 동일한 프로세서에서는 퀵정렬, 머지정렬, 힙정렬 순으로 에너지 효율이 좋은 것으로 관측되었고, 데이터 규모가 커질수록 에너지 효율이 증가했다. 비록 최악의 시간 복잡도는 동일하지만 연산횟수는 물론 메모리 사용 정도, 캐쉬 적중률 등이 달라 차이를 보이므로 향후 에너지 최적화에 중요한 고려대상이 된다. 프로세서별로는 최근의 고성능

CPU일수록 계산성능은 좋아지지만, 에너지 효율 면으로는 노트북/모바일용 CPU 칩을 장착한 시스템이 효율이 높이 관찰되는 것은 예측 가능 한 사실이다.

최근의 멀티코어 칩도 성능은 물론 에너지 효율도 좋은 것을 알 수 있고, 이는 병렬실험에서도 확인된다. 그림 3의 왼쪽 3개 알고리즘 각각을 1개 코어에서 실험한 결과를 오른쪽 PSRS 알고리즘을 다 코어에서 실험한 결과와 비교했을 때 다수 코어의 결과가 모두 우수함을 보인다. 그 이유로는 내부 코어 프로세서를 제외한 공통적 자원, 즉 버스, 캐쉬 등에서 소비되는 에너지의(오버헤드) 비율이 단일 코어 칩형 프로세서에 비해 멀티코어를 동작시킬 때 큰 감소효과를 얻기 때문이다.

3. 에너지 효율 증대를 위한 퀵정렬 개선

n개 키에 대한 퀵정렬 및 머지정렬의 실행의 소요시간을 t_{QK}, t_{MG} 로 재귀적으로 표시하면

$$t_{QK}(n) = nT_C + FT_S + t_{QK}(ln) + t_{QK}((1-l)n)$$

$$t_{MG}(n) = GT_C + nT_S + 2\,t_{MG}(n/2)$$

단, $F, G \leqq n$, $0 \leqq l \leqq 1.0$ 이고 T_S 와 T_C는 각각 단위 swap, 단위 비교의 소요연산 시간을 표시한다. 정렬연산은 진행과정에서 캐쉬 메모리를 통한 데이터 재사용 빈도가 높지 않아 메모리 접근 속도 개선 효과가 낮다. 대신 데이터 교환(swap) 빈도가 총 소요시간에서 큰 비중을 차지하게 된다. 특히 고성능 프로세서에서는 내부 산술연산 비용/시간에 비해 메모리 접근/(특히 쓰기동작)에 수십내지 수백배의 지연이 있게되어($T_s \gg T_C$)[6], 알고리즘에서 얼마나 많은 계산(산술연산 등)을 줄이는가 보다 swap 횟수를 얼마나 줄이는가가 전체 실행시간과 전력소모량을 절감하는가의 관건이 된다. 실험적으로도 보면, 데이터규모가 커질수록 머지정렬이 퀵정렬에 비해 소요시간이 매우 커지게 되어 퀵정렬의 에너지 절감법을 연구대상으로 한다

퀵정렬은 재귀적 진행과정에서 피봇(pivot)을 기준으로 균등하게 분할(partition)될수록 성능향상 및 에너지 소모가 감소한다. 하지만 적절한 피봇을 찾기 어렵기 때문에 본 연구에서는 PSRS에서의 샘플기법을 도입하여 분할의 균등성을 향상시켰다. 저전력 퀵정렬 (Energry-aware quicksort)은 초기 수 단계 동안 피봇산출에 공들여 다음과 같이 진행된다.

⑴ 현재의 배열 D[1:M]에서 균등한 간격으로 k=log M 개의 샘플을 확보하고 정렬하여 S[1:k]로 저장한 후 k/p 간격으로 working pivot V[1], V[2] .... V[p-1]을 설정한다

⑵ $S[\frac{p-1}{2}]$로 퀵정렬의 피봇을 정한 후 D[1:M]을 분할하고 동시에 working pivot을 경계점으로 하여 M 키의 histogram을 구한다.

⑶ 분할 결과 좌우의 키 배치의 불균등화 정도가 한계치 q (예를 들어, $q = 0.2$) 이내이면 histogram을 참조하여 working pivot 중 최적치를 피봇으로 선택하여 다음 레벨의 분할을 행한다. 그렇지 않으면 규모가 큰 분할영역 (크기가 M')을 ⑴⑵의 과정을 통해 M' 개의 배열을 분할하여 진행한다.

실험적으로는 위 방식을 레벨 5까지 적용한 결과 표2와 같이 최대 6% 정도의 실행시간 단축효과와 에너지 절약효과를 얻었다

4. 결론

green computing 실현을 위한 초기연구로 정렬알고리즘을 검토 수정하여 실험하였다. 퀵정렬의 분할과정을 개선하여 절전효과를 얻었고, 그 원리는 일반적인 알고리즘 설계에 적용될 수 있다. 한편, 추후 병렬알고리즘으로 확장하여 절감효과를 극대화할 계획이다. 적용 알고리즘의 전력소모에 대한 해석적 모델링이 향후 과제이다.

참고문헌

[1] 지식경제부, Weekly IT Brief 2009-12,13호, 2009.4/5

[2] 임철수, 그린컴퓨팅 기술동향 및 추진방안, 2009.3 전자신문사, http:// www.itconferences.com

[3] K.W. Cameron, The Road to greener IT pastures, *IEEE Computer*, May 2009, pp. 87-89.

[4] W. Feng, The importance of being low power in high-performance computing, *CTWatch Quarterly*, August 2005.

[5] X. Li, P. Lu, J. Schaeffer, J. Shillington, P.S.Wong, and H. Shi, On the versatility of parallel sorting by regular sampling, *Parallel Computing:* 19, pp. 1079-1193, 1993.

[6] W. Dally, Power Efficient Supercomputing, *LBLWorkshop on Accelerator- based Computing and manycore,* Nov. 2009.

[표1] 실험 대상의 컴퓨터/CPU 상세 사양

CPU	i5-750	E7200	Z270	L2500
CPU clock freq. (GHz)	2.67	2.53	1.6	1.83
Number of cores	4	2	2	2
RAM /cache capacity	4GB/ 8M	4GB/ 3M	1GB/ 0.5M	1GB/ 2M
Aux. storage type	HD	HD	HD	SSD
Idle to peak power, Watts	49-135	49-62	12-17	13-21

[표2] 저전력 퀵정렬의 실행시간 (단위: 초)

Input Size	10M	20M	40M	80M	160M
Original Quicksort	3.21	6.64	13.91	28.79	60.45
Energy-aware Quicksort	3.02	6.29	13.12	27.31	56.66
Improvement (%)	5.9%	5.3%	5.7%	5.1%	6.3%

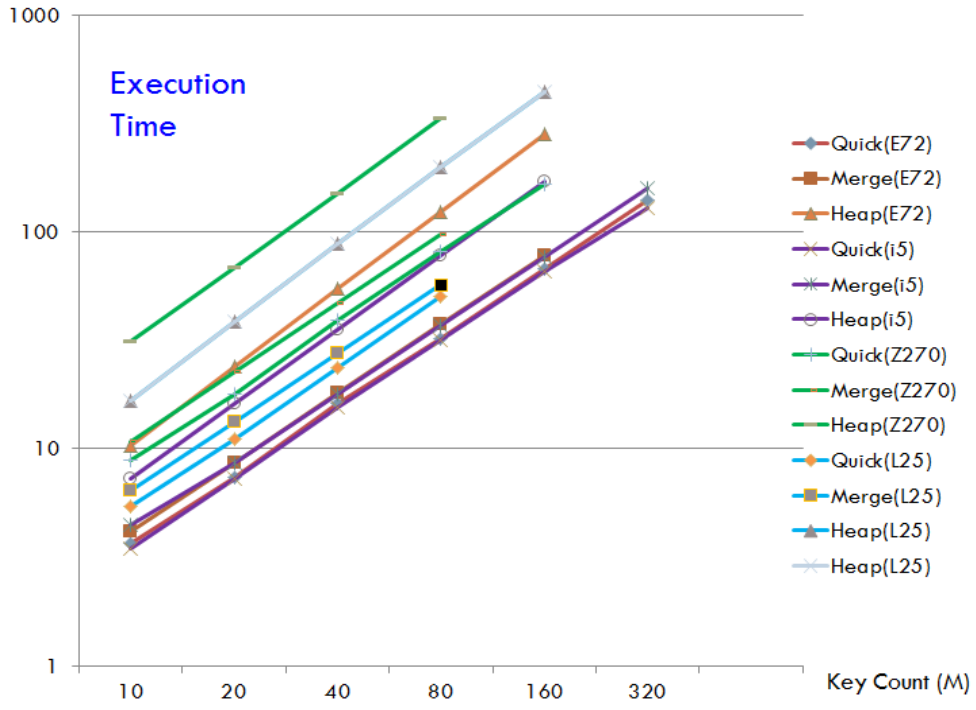

그림 1 대상 정렬 알고리즘의 입력 규모에 따른 수행시간

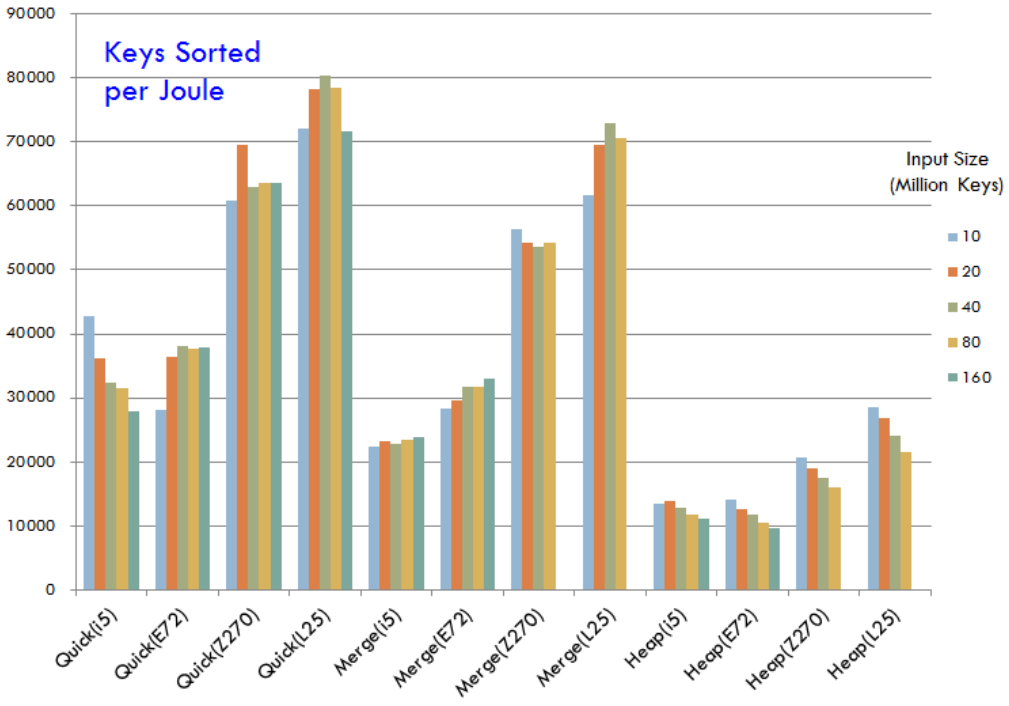

그림 2 정렬 알고리즘 별 에너지 효율 - 단위 Joule 당 정렬 키 갯수

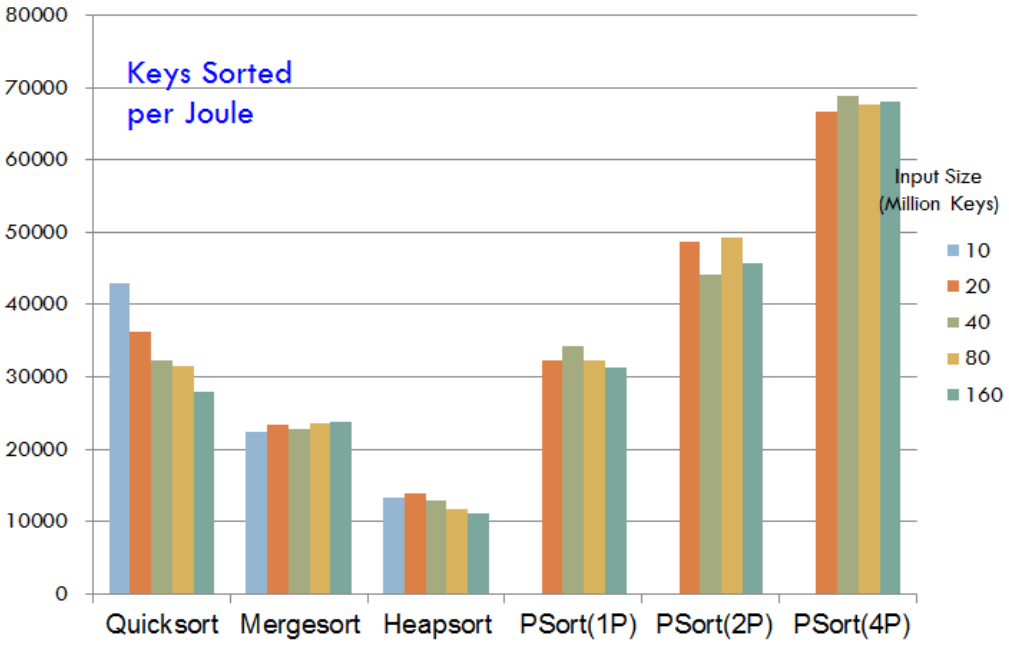

그림 3 멀티코어 (i5-750) CPU의 정렬연산 에너지 효율