과학계산용 클러스터 파일시스템에서의 인터럽트 통합효과 분석*

박석중[O1] 우준[2] 이재국[1] 김형식[1]
[1]충남대학교 컴퓨터공학과 [2]한국과학기술정보연구원
midstone@csal.cnu.ac.kr, wjnadia@kisti.re.kr, empire@csal.cnu.ac.kr, hkim@cnu.kr

Analysis on the interrupt coalescence effect in cluster file systems for scientific computation

Seok-Jung Park[O1], Joon Woo[2], Jae-Kook Lee[1], Hyong-Shik Kim[1]
[1]Dept. of Computer Engineering Chungnam University
[2]Korea Institute of Science and Technology Information

요 약

클러스터 파일시스템은 근거리 또는 원거리에 있는 클러스터 시스템 간에 연구데이터 공유 뿐 아니라, 실시간 계산을 위한 데이터 저장 공간으로 사용되는 네트워크 기반의 파일시스템이다. 고도의 과학계산을 처리할 때 계산노드들은 네트워크를 통해 연결된 클러스터 파일시스템으로부터 대용량의 데이터를 송수신하는 과정에서 CPU의 부하가 생기게 되고 이러한 문제는 계산노드로 하여금 과학계산의 속도를 저하시키는 요인이 된다.

본 논문에서는 패킷 송수신으로 인한 CPU 부하를 줄이고 이를 통하여 계산 성능을 향상시킬 목적으로 계산노드에서 수신하는 패킷들에 대해 인터럽트를 통합할 때 CPU 사용률에 미치는 영향을 분석하였다.

1. 서 론

고도의 계산을 요하는 과학계산을 수행하기 위하여 슈퍼컴퓨터와 같은 고가의 단일 기종을 사용하는 대신 저가의 다수 계산 노드(processing node)들을 고속의 네트워크를 통해 하나로 묶는 클러스터 시스템은 이미 슈퍼컴퓨팅 분야에서 상당한 부분을 차지하게 되었다.

이러한 클러스터 시스템은 LAN에 연동된 내부 이기종 클러스터 간 또는 WAN에 연동된 타기관의 클러스터 간 데이터의 공유를 통한 효율적이고 편리한 슈퍼컴퓨팅 자원 연동을 위해서 클러스터 파일 시스템을 사용한다.

클러스터 파일시스템은 고속의 네트워크로 연결되어 데이터를 전송해준다. 이러한 클러스터 파일시스템은 고도의 과학계산을 위한 데이터를 저장할 뿐 아니라, 과학 응용 계산중에도 임시 중간 데이터를 읽고 쓰므로 대용량의 데이터가 각각의 계산노드들에게 고속으로 전송된다.

그러나 계산노드에서 1Gbps 이상의 고성능 네트워크를 통해 받는 데이터의 양이 증가하면서 패킷 송수신으로 인한 CPU 부하가 증가하여, 계산에 할당해야 할 CPU 자원을 네트워크에서 패킷을 수신하는데 사용하게 된다. 일반적인 상황 하에서는 이 정도의 CPU 사용률은 문제가 되지 않지만, 고도의 과학계산에서는 계산 시간을 증가시키는 요인으로 작용하게 된다. 따라서 네트워크에서 패킷 송수신으로 인한 CPU 사용률을 줄일 필요성이 나타난다.

2절에서는 클러스터 파일시스템 환경에서의 과학계산 특징을 알아보고, 3절에서는 데이터 송수신과 인터럽트의 관계를 보인다. 마지막으로 4절에서는 패킷 송수신으로 인한 CPU 사용률을 감소시키는 방법을 인터럽트 통합을 중심으로 다루도록 한다.

2. 클러스터 파일시스템 환경에서의 과학 계산

클러스터 파일시스템 환경에서의 과학 계산은 다음과 같은 특징을 갖는다.

(1) 대용량 데이터

네트워크로 연결된 클러스터 파일시스템을 이용해서 대용량의 연구데이터를 계산노드로 읽어오고 실시간으로 계산을 수행하며, 이 때 생성되는 임시 중간 데이터를 다시 읽고 쓰는 과정을 되풀이 하므로 대량의 네트워크 트래픽이 발생한다.

(2) 고도의 CPU 사용률

상호 연결된 클러스터 시스템들은 기상데이터 시뮬레이션, 고에너지 데이터 계산 등 고도의 계산과정을 거치므로 최대의 CPU 사용률을 나타낸다.

3. 데이터 송수신과 인터럽트

"본 연구는 지식경제부 및 정보통신연구진흥원의 대학IT연구센터 지원사업의 연구결과로 수행되었음" (IITA-2008-C1090-0801-0016)

3.1 네트워크 패킷 수신 절차

애플리케이션이 네트워크를 통해 데이터를 수신하는 과정은 다음과 같다. NIC에 패킷이 도착하면 NIC 버퍼에 저장을 하고 인터럽트를 발생시킨다. 인터럽트 서비스 루틴(NIC Driver)에 의해 커널메모리로 복사되고 상위 계층 처리를 위해 CPU 큐에 저장된다. 수신된 각각의 패킷은 각 계층 마다 헤더를 처리하는 과정을 거쳐 애플리케이션에 도착하게 된다.

3.2 인터럽트 처리 오버헤드

위에서 설명한 네트워크 패킷 처리 과정은 고속으로 패킷을 전송할 때 더욱 문제가 될 수 있다. NIC에서는 매 패킷이 도착 할 때마다 패킷 헤더 처리를 위한 인터럽트가 발생하게 되는데, 이것은 지나친 컨텍스트 스위칭을 발생시키고 계산 노드에 큰 부하로 작용하게 된다.

3.3 인터럽트 통합

인터럽트 통합은 패킷이 도착할 때마다 인터럽트를 발생시키지 않고 일정한 양의 패킷이 도착하거나 시간이 경과했을 때 인터럽트를 발생시킴으로서 패킷처리를 위한 인터럽트의 수를 줄이는 방법이다.

대용량 데이터를 전송할 때 매 패킷마다 인터럽트를 발생시킴으로 인해 패킷 송수신에 많은 CPU 부하가 발생하여, 애플리케이션에서 사용해야 할 CPU를 패킷처리에 사용하게 되므로 이것을 해결하기 위해 인터럽트 통합을 활용한다.

특히 슈퍼컴퓨팅 계산을 지원하기 위해 사용되는 클러스터 파일시스템은 대용량의 네트워크I/O를 발생시키기 때문에 고도로 CPU를 사용하는 과학계산을 위해 CPU 사용률을 줄일 필요성이 있다.

리눅스 환경에서 CPU 사용률을 감소시키기 위해 사용하는 패킷 수신 인터럽트 통합 변수는 rx-frames, rx-usecs 으로 인터럽트 간격을 프레임수와 시간으로 정의한다.

3.3.1 rx-frames

rx-frames는 몇 개의 패킷마다 인터럽트를 발생시킬 것인지를 지정하는 변수이다. 예를 들어 rx-frames가 6으로 설정되어 있을 경우, 패킷이 6개 도착했을 때 인터럽트가 발생해 도착한 패킷을 처리하게 된다. 이 값은 각 네트워크 인터페이스 카드 드라이버에 기본 값으로 지정되어 있으며, 시험에 사용된 Broadcom사의 NetXtreme II BCM5708 Gigabit Ethernet 드라이버에는 6으로 지정되어 있다.

3.3.2 rx-usecs

rx-usecs는 마지막 패킷이 도착한 후 다음 패킷이 도착할 때까지 기다리는 시간을 정의한다. 예를 들어 rx-usecs가 18로 설정되어 있다면, 어떠한 한 패킷이 도착한 후 다음 패킷이 18μs동안 도착하지 않으면 rx-frames 에 설정된 값만큼 패킷을 받지 않았더라도 인터럽트를 발생시킨다. 이것은 트래픽이 적은 상황에서 인터럽트 통합으로 인한 패킷 지연시간이 지나치게 커지는 것을 방지한다. Broadcom사의 NetXtreme II BCM5708 Gigabit Ethernet 드라이버에는 기본 값 18로 지정되어 있다.

4. 인터럽트 통합

1Gbps 이더넷 환경에서 처리율을 향상시키고CPU 사용률을 줄이기 위해서 점보 프레임을 사용한다. 1500바이트의 기존 MTU보다 큰 크기를 점보 프레임으로 부르며, 최대 9000바이트까지 가능하다[1]. 이 시험에서는 CPU 부하가 가장 적고 처리율이 좋은 9000을 MTU로 사용하였다.

4.1 시험환경

시험환경은 다음과 같다.

- 구간 : KISTI(대전) - 부산대(부산)
- 네트워크 : 1Gbps 전용 네트워크
- 컴퓨터
  - 송신
    - CPU : Xeon 2.8Ghz Dual core
    - 메모리 : 3GB
    - NIC : Broadcom Corporation NetXtreme BCM5703X Gigabit Ethernet
    - OS : Red Hat Enterprise Linux AS (2.6.9-54.ELsmp)
  - 수신
    - CPU : Xeon 2.3Ghz Quad core
    - 메모리 : 2GB
    - NIC : Broadcom Corporation NetXtreme II BCM5708 Gigabit Ethernet
    - OS : Red Hat Enterprise Linux ES (2.6.9-42.ELsmp)

4.2 rx-frames에 따른 영향

CPU 사용량을 줄이기 위해 rx-frames를 증가시키며 시험하였다. rx-usecs는 시스템에 설치된 NIC 드라이버의 기본값인 18로 고정하고 rx-frames 드라이버 기본값인 6의 배수로 증가시키면서 인터럽트의 개수와 이에 따른 처리율에 대한 영향을 살펴보았다. 그림 1은 rx-frames의 증가에 따른 인터럽트 수의 영향을 보여준다.

그림 1은 rx-frames를 조정하면 패킷도착에 따른 인터럽트를 통합하여 인터럽트 개수를 감소시킬 것이라는 예상에서 벗어나는 결과를 보인다. rx-frames와 관계없이 인터럽트의 개수는 계속해서 동일하게 발생하고 있다. 또한 그림 2는 rx-frames의 변화에 따른 처리율의 변화도 없음을 보인다.

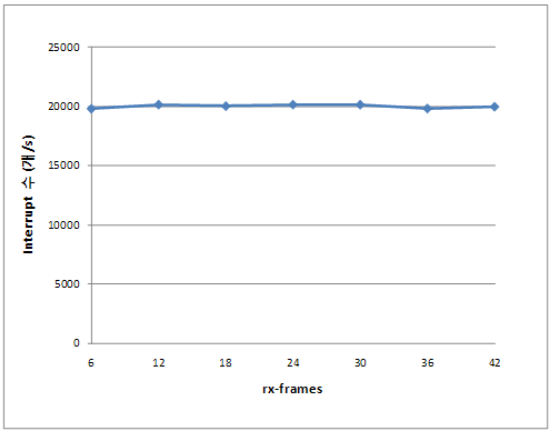

그림 1. rx-frames에 따른 인터럽트 수
(rx-usecs = 18μs)

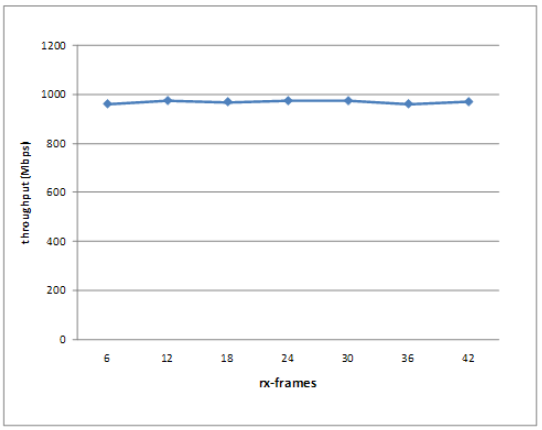

그림 2. rx-frames에 따른 처리율
(rx-usecs = 18μs)

rx-frames에 따라 인터럽트가 통합되지 않고 영향을 미치지 않는 이유는 기본값인 rx-usecs인 18μs가 점보 프레임 환경에서는 인터럽트를 통합하기 위한 부적절한 값이기 때문이다. 4.3절에서 이에 대한 자세한 설명을 하고 4.4절에서 rx-usecs값을 적절한 값으로 재설정 한 후의 rx-frames로 인한 영향을 보인다.

4.3 rx-usecs에 따른 영향

CPU 사용량을 줄이기 위해 rx-usecs를 증가시키며 시험하였다. rx-frames는 시스템에 설치된 NIC 드라이버의 기본값인 6으로 고정하고 rx-usecs를 드라이버 기본 값인 18의 배수로 증가시키면서 인터럽트의 개수와 이에 따른 처리율을 측정하였다. 그림 3은 rx-usecs의 증가에 따른 인터럽트 수의 영향을 보여준다.

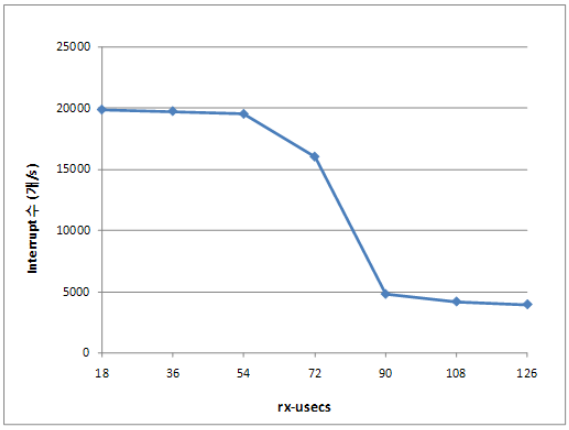

그림 3. rx-usecs에 따른 인터럽트 수

rx-usecs가 증가함에 따라 초당 약 2만번 발생하던 인터럽트가 54μs를 기점으로 인터럽트가 통합되며 초당 약 4000개 수준으로 감사한다. usecs가 90μs 가 넘어서야 인터럽트 통합효과가 크게 나타나는 것은 패킷이 도착하는 시간차와 관련이 있다. 1Gbps 망에서 패킷간의 시간차는 다음과 같이 구할 수 있다.

패킷간의 시간차 = 패킷크기 / 1Gbps

현재 사용되는 패킷 크기는 최고 점보 프레임9000 Byte 이므로 패킷간의 시간차는 9000 Byte / 1Gbps = 72μs로 계산된다. rx-usecs는 마지막 패킷이 도착 한 후 다음 패킷을 기다리는 시간 한계를 규정하는데 rx-usecs가 현재의 패킷 시간차인 72μs 이하로 설정된 구간에서는 마지막 패킷을 받은 후 다음 패킷을 받기 전에 rx-usecs에 의해 인터럽트가 발생하기 때문에 rx-frame이 어떠한 값인지에 관계없이 인터럽트가 통합되지 않는다. 이것은 앞의 그래프 1과 2에서 rx-frames에 의해 인터럽트 통합이 왜 이루어지지 않았는지를 잘 설명해준다.

또한 MTU 1500일 때는 드라이버 기본 설정값인 18이 rx-usecs으로 최적값이지만, 점보 프레임을 사용하면서 패킷간 시간차보다 더 큰 값으로 재설정해야 함을 알 수 있다.

그림 4는 rx-usecs에 의한 처리율의 변화를 보여준다. rx-usecs는 점보 프레임을 사용할 경우에 인터럽트를 통합하기 위한 중요한 변수이지만 처리율에는 큰 영향을 미치지 않는 것을 알 수 있다.

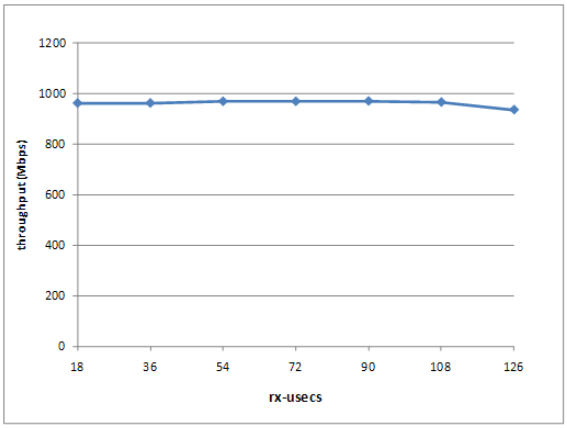

그림 4. rx-usecs에 따른 처리율

4.4 rx-frames의 영향 (rx-usecs = 90㎲)

4.2에서 rx-usecs의 기본값인 18㎲는 기존의 MTU 1500일 때 인터럽트를 통합하기 위한 적정 값으로 현재 클러스터 파일시스템의 점보프레임 환경에서 인터럽트를 통합하기 위해서는 점보프레임의 패킷 시간차보다 큰 값으로 재설정 해야하는 것을 4.2에서 알 수 있었다. 따라서 점보 프레임환경에 맞는 rx-usecs 값으로 설정한 후 rx-frames에 의한 영향을 보기 위해서, 점보 프레임에 대한 적정 값 90㎲를 rx-usecs로 재설정 한 후 rx-frames을 증가시키면서 시험을 하였다

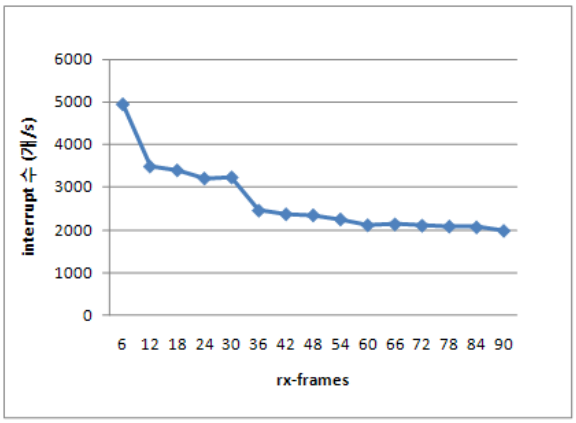

그림 5. rx-frames에 따른 인터럽트 수
(rx-usecs = 90㎲)

그림 5는 점보 프레임에 인터럽트 통합이 일어날 수 있도록 적절하게 설정된 rx-usecs에서 rx-frames에 의한 인터럽트 통합을 다시 보여준다 rx-frames의 값이 증가함에 따라 인터럽트가 적게 발생하는 것을 볼 수 있다.

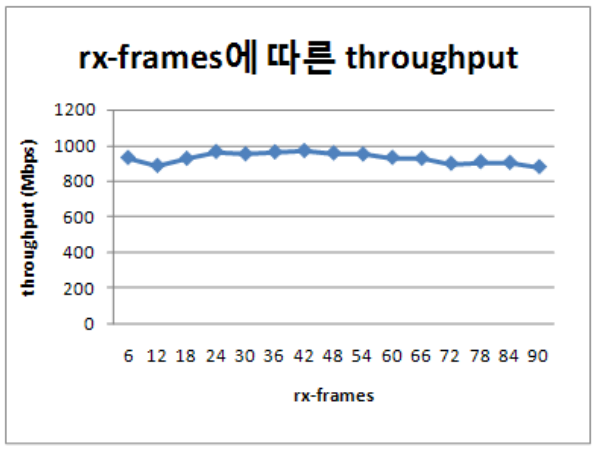

그림 6. rx-frames에 따른 처리율
(rx-usecs = 90㎲)

그림 6은 rx-frames에 따른 처리율을 보여준다. 큰 변화는 없지만 rx-frames 값 42를 지남에 따라 조금씩 떨어지기 시작한다. 이것은 인터럽트의 주기가 너무 길어져서 NIC의 버퍼가 부족해 패킷을 드롭하기 때문에 나타나는 현상으로, 인터럽트와 처리율을 고려해 적정한 값을 선택해야 한다.

4.5 CPU 사용률 비교

그림 7은 점보 프레임을 갖는 클러스터 환경에서 인터럽트 통합 변수가 기본 값일 때와 최적 값 일때의CPU 사용률의 변화를 보여준다. 점보 프레임을 사용함으로 인해 변경된 인터럽트 통합 최적 값은 위의 실험결과를 바탕으로 하여 rx-frames을 42로 rx-usecs을 90으로 설정 하였다. 기존에는 패킷 송수신 처리에 따른 CPU 사용률이 11%이었던 것과 비교해서 최적 값으로 설정했을 때의 CPU사용률이 현저하게 떨어진 것을 볼 수 있다.

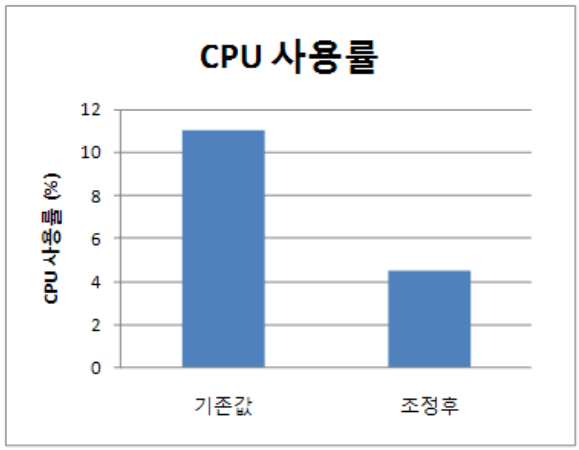

그림 7. 최적화 전후의 CPU 사용률

4.6 지연시간에 대한 영향

rx-usecs과 rx-frames 값을 증가시킴으로써 인터럽트 통합의 효과를 이끌어내고 이에 따라CPU 사용률을 줄일 수 있었다. 그렇지만 rx-usecs와 rx-frames의 곱에 의해 패킷 지연시간이 결정되기 때문에 무조건 큰 값보다는 rx-usecs는 인터럽트 통합이 나타나기 시작하는 최소값으로, rx-frames는 인터럽트가 많이 줄어든 가장 작은 값으로 적절한 값을 정하는 것이 필요하다.

시험에 사용된 시스템에서 점보프레임 환경에서CPU 사용률을 줄이기 위한 적절한 값인 42를 rx-frames으로 90을 rx-usecs으로 설정했을 때, 최악의 경우 지연시간은 42 * 90μs = 3780μs =3.8ms 가 추가되고 기존 Round Trip Time(RTT)가 5~6ms 인 것을 감안 했을 때 이 값은 상당히 큰 값으로 볼 수 있다.

그럼에도 불구하고 대용량 트래픽 전송에서는 짧은 지연시간을 요구하지 않으며, 지연시간의 증가로 인한 처리율의 감소는 TCP 윈도우 크기를 증가시킴으로 해결할 수 있으므로 과학계산을 위한 가용 CPU 자원을 확보하기 위해 rx-usecs와 rx-frames를 증가시키는 것은 효과적이다.

5. 결론

지금까지 1Gbps 고성능 장거리 네트워크를 사용하는 클러스터 파일시스템에서 NIC의 패킷 인터럽트를 통합하여 계산 노드의 CPU 부하를 감소시키는 것을 보였다.

고도의 과학계산을 위한 클러스터파일 시스템은 고대역폭의 네트워크가 필요하다. 1Gbps 네트워크에서는 처리율을 향상시키고 CPU 사용률을 낮추기 위해 점보 프레임을 사용하므로, 본 논문에서는 점보프레임을 전제로 패킷 송수신으로 인한 CPU 부하를 감소시키기 위해 인터럽트를 통합하였으며, 인터럽트 통합을 위한 변수 rx-frames와 rx-usecs의 적정한 값과 상관관계를 보였다.

인터럽트 통합을 통해서 패킷 송수신에 의한CPU 사용률을 기존보다 1/2 이하로 감소시켰으며 이에 따라 과학응용계산 목적으로 CPU 자원을 더 많이 할당할 수 있는 것으로 기대된다.

참고문헌

[1]http://en.wikipedia.org/wiki/Jumbo_frame