



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년11월17일
(11) 등록번호 10-2179769
(24) 등록일자 2020년11월11일

(51) 국제특허분류(Int. Cl.)
H04L 12/869 (2013.01) H04L 12/861 (2013.01)
H04L 12/865 (2013.01)
(52) CPC특허분류
H04L 47/58 (2013.01)
H04L 47/6275 (2013.01)
(21) 출원번호 10-2019-0136628
(22) 출원일자 2019년10월30일
심사청구일자 2019년10월30일
(56) 선행기술조사문헌
KR1020150113783 A

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
이원준
서울특별시 강남구 선릉로 221, 104동 2005호(도곡동, 도곡렉슬아파트)
김규영
서울특별시 성북구 안암로 145, 고려대학교 우정정보통신관 507A (안암동5가)
(74) 대리인
윤귀상

전체 청구항 수 : 총 13 항

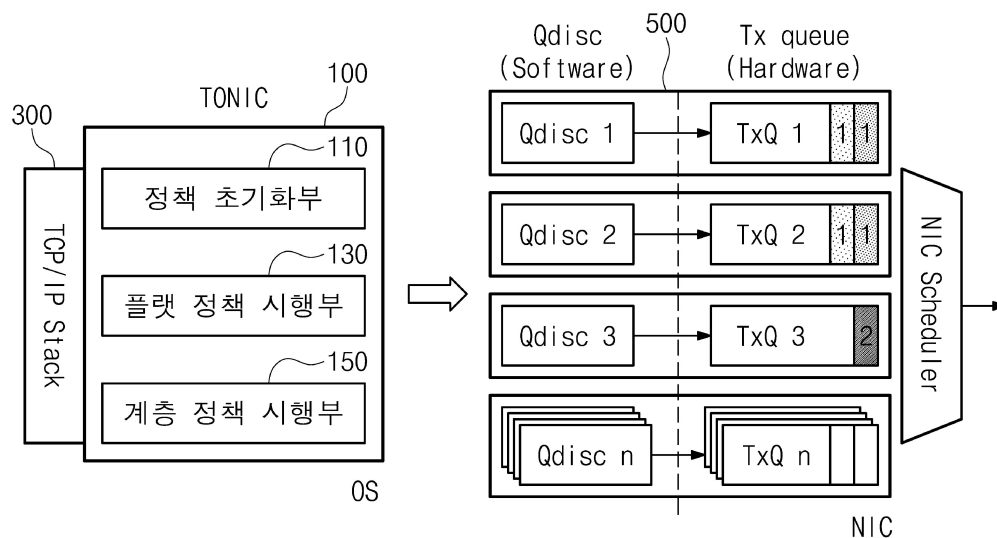
심사관 : 김대성

(54) 발명의 명칭 다중 큐를 지원하는 네트워크 정책 적용 방법, 이를 수행하기 위한 기록 매체 및 장치

(57) 요약

다중 큐를 지원하는 네트워크 정책 적용 방법은, 네트워크 관리자가 제출한 네트워크 정책 내용을 기초로, 모듈 내부의 테넌트 매핑 배열, 테넌트 가중치 배열, 시작 큐 인덱스 배열 및 포트 번호 배열과 변수들을 초기화하는 단계; 초기화된 배열로부터 획득한 테넌트 ID, 테넌트 가중치, 테넌트의 시작 큐 인덱스 및 현재의 큐를 추적하는 인덱스를 기초로, 각 테넌트에게 할당된 가중치만큼의 네트워크 인터페이스 카드(Network Interface Cards, NIC) 큐를 할당하여 테넌트 간 네트워크 격리를 보장하는 단계; 및 테넌트 내 응용 간의 우선순위를 다르게 하기 위해 우선순위가 높은 응용 패킷을 Qdisc의 머리에 삽입하여 테넌트 내 응용 간의 네트워크 격리를 보장하는 단계;를 포함한다. 이에 따라, 새로운 하드웨어 개발에 필요한 비용 및 교체 시간 없이, 현존하는 상용 MQ-NIC만으로도 여러 패킷 스케줄러를 사용하여 서버 내 네트워크 정책을 적용할 수 있다.

대표도 - 도1



(52) CPC특허분류

H04L 49/9068 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	2019027027
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	(원천)차세대정보컴퓨팅기술개발사업
연구과제명	Web 응용을 위한 Seamless 자율 전송 네트워킹 제어 프레임워크
기 여 율	1/1
과제수행기관명	고려대학교
연구기간	2019.03.01 ~ 2019.12.31

명세서

청구범위

청구항 1

네트워크 관리자가 제출한 네트워크 정책 내용을 기초로, 모듈 내부의 테넌트 매핑 배열, 테넌트 가중치 배열, 시작 큐 인덱스 배열 및 포트 번호 배열과 변수들을 초기화하는 단계;

초기화된 배열로부터 획득한 테넌트 ID, 테넌트 가중치, 테넌트의 시작 큐 인덱스 및 현재의 큐를 추적하는 인덱스를 기초로, 각 테넌트에게 할당된 가중치만큼의 네트워크 인터페이스 카드(Network Interface Cards, NIC) 큐를 할당하여 테넌트 간 네트워크 격리를 보장하는 단계; 및

테넌트 내 응용 간의 우선순위를 다르게 하기 위해 우선순위가 높은 응용 패킷을 Qdisc의 머리에 삽입하여 테넌트 내 응용 간의 네트워크 격리를 보장하는 단계;를 포함하는, 다중 큐를 지원하는 네트워크 정책 적용 방법.

청구항 2

제1항에 있어서, 상기 테넌트 간 네트워크 격리를 보장하는 단계는,

테넌트의 현재의 큐를 추적하는 인덱스에 해당하는 값이 해당 테넌트의 가중치 이상인 경우, 현재 큐 인덱스에 해당하는 값을 0으로 초기화하는, 다중 큐를 지원하는 네트워크 정책 적용 방법.

청구항 3

제2항에 있어서, 상기 테넌트 간 네트워크 격리를 보장하는 단계는,

현재 처리 중인 패킷이 속한 소켓의 큐 매핑 메타데이터를 상기 테넌트의 현재의 큐를 추적하는 인덱스에 해당하는 값과 시작 큐 인덱스를 합한 값으로 갱신하는, 다중 큐를 지원하는 네트워크 정책 적용 방법.

청구항 4

제3항에 있어서, 상기 테넌트 간 네트워크 격리를 보장하는 단계는,

다음 패킷을 위해 현재의 큐를 추적하는 인덱스를 갱신하는 단계를 더 포함하는, 다중 큐를 지원하는 네트워크 정책 적용 방법.

청구항 5

제3항에 있어서, 상기 테넌트 내 응용 간의 네트워크 격리를 보장하는 단계는,

현재 사용 중인 패킷의 포트가 포트 번호 배열에 포함되어 있다면 현재 처리 중인 패킷이 속한 소켓의 우선순위 메타데이터 값을 1로 설정하여 높은 우선순위를 부여하고, 그 이외의 경우 우선순위 메타데이터 값을 0으로 설정하여 낮은 우선순위를 부여하는, 다중 큐를 지원하는 네트워크 정책 적용 방법.

청구항 6

제5항에 있어서,

상기 현재 처리 중인 패킷이 속한 소켓의 큐 매핑 메타데이터 값과 상기 우선순위 메타데이터 값을 기초로 현재 사용 중인 패킷의 인큐 여부를 결정하는 단계를 더 포함하는, 다중 큐를 지원하는 네트워크 정책 적용 방법.

청구항 7

제1항에 있어서,

상기 테넌트 매핑 배열은 현재 처리 중인 패킷이 어느 테넌트 소유의 패킷인지 구분할 수 있도록 CPU 코어 별로 할당되어 있는 테넌트의 인덱스를 표기하고,

상기 테넌트 가중치 배열은 테넌트 별 네트워크 스케줄링 가중치를 저장하고,

상기 시작 큐 인덱스 배열은 테넌트 가중치 배열 값을 기반으로 각 테넌트 별로 점유하는 NIC 큐 인덱스를 계산하여 저장하고,

상기 포트 번호 배열은 특정 테넌트가 높은 우선순위를 부여하고 싶은 응용이 있다면 해당 응용이 사용하는 포트 번호를 배열 형태로 저장하는, 다중 큐를 지원하는 네트워크 정책 적용 방법.

청구항 8

제1항 내지 제7항 중 어느 하나의 항에 따른 상기 다중 큐를 지원하는 네트워크 정책 적용 방법을 수행하기 위한 컴퓨터 프로그램이 기록된 컴퓨터로 판독 가능한 기록 매체.

청구항 9

네트워크 관리자가 제출한 네트워크 정책 내용을 기초로, 모듈 내부의 테넌트 매핑 배열, 테넌트 가중치 배열, 시작 큐 인덱스 배열 및 포트 번호 배열과 변수들을 초기화하는 정책 초기화부;

상기 정책 초기화부로부터 획득한 테넌트 ID, 테넌트 가중치, 테넌트의 시작 큐 인덱스 및 현재의 큐를 추적하는 인덱스를 기초로, 각 테넌트에게 할당된 가중치만큼의 네트워크 인터페이스 카드(Network Interface Cards, NIC) 큐를 할당하여 테넌트 간 네트워크 격리를 보장하는 플랫폼 정책 시행부; 및

테넌트 내 응용 간의 우선순위를 다르게 하기 위해 우선순위가 높은 응용 패킷을 Qdisc의 머리에 삽입하여 테넌트 내 응용 간의 네트워크 격리를 보장하는 계층 정책 시행부;를 포함하는, 다중 큐를 지원하는 네트워크 정책 적용 장치.

청구항 10

제9항에 있어서, 상기 플랫폼 정책 시행부는,

테넌트의 현재의 큐를 추적하는 인덱스에 해당하는 값이 해당 테넌트의 가중치 이상인 경우, 현재 큐 인덱스에 해당하는 값을 0으로 초기화하고,

현재 처리 중인 패킷이 속한 소켓의 큐 매핑 메타데이터를 상기 테넌트의 현재의 큐를 추적하는 인덱스에 해당하는 값과 시작 큐 인덱스를 합한 값으로 갱신하고,

다음 패킷을 위해 현재의 큐를 추적하는 인덱스를 갱신하는, 다중 큐를 지원하는 네트워크 정책 적용 장치.

청구항 11

제10항에 있어서, 상기 계층 정책 시행부는,

현재 사용 중인 패킷의 포트가 포트 번호 배열에 포함되어 있다면 현재 처리 중인 패킷이 속한 소켓의 우선순위 메타데이터 값을 1로 설정하여 높은 우선순위를 부여하고, 그 이외의 경우 우선순위 메타데이터 값을 0으로 설정하여 낮은 우선순위를 부여하는, 다중 큐를 지원하는 네트워크 정책 적용 장치.

청구항 12

제11항에 있어서,

상기 현재 처리 중인 패킷이 속한 소켓의 큐 매핑 메타데이터 값과 상기 우선순위 메타데이터 값을 기초로 현재 사용 중인 패킷의 인큐 여부를 결정하는, 다중 큐를 지원하는 네트워크 정책 적용 장치.

청구항 13

제9항에 있어서,

상기 테넌트 매핑 배열은 현재 처리 중인 패킷이 어느 테넌트 소유의 패킷인지 구분할 수 있도록 CPU 코어 별로 할당되어 있는 테넌트의 인덱스를 표기하고,

상기 테넌트 가중치 배열은 테넌트 별 네트워크 스케줄링 가중치를 저장하고,

상기 시작 큐 인덱스 배열은 테넌트 가중치 배열 값을 기반으로 각 테넌트 별로 점유하는 NIC 큐 인덱스를 계산하여 저장하고,

상기 포트 번호 배열은 특정 테넌트가 높은 우선순위를 부여하고 싶은 응용이 있다면 해당 응용이 사용하는 포트 번호를 배열 형태로 저장하는, 다중 큐를 지원하는 네트워크 정책 적용 장치.

발명의 설명

기술 분야

[0001] 본 발명은 다중 큐를 지원하는 네트워크 정책 적용 방법, 이를 수행하기 위한 기록 매체 및 장치에 관한 것으로서, 더욱 상세하게는 패킷 데이터 처리 방법에 관한 것으로, 다중 큐를 지원하는 네트워크 인터페이스 카드(Network Interface Cards, NIC)를 장착한 컴퓨터 서버 장치에서 다양한 패킷 스케줄링 방법을 표현하는 기술에 관한 것이다.

배경 기술

[0003] 클라우드 데이터 센터의 서버는 다양한 응용을 사용하는 다수의 테넌트가 공유한다. 서버는 데이터 센터 네트워크의 첫 번째 병목 지점이므로 테넌트는 테넌트 간 및 테넌트 내 응용 간 네트워크 격리를 필요로 한다.

[0004] 네트워크 격리를 보장하기 위해 운영자는 테넌트 사이의 가중치를 지정하고 테넌트 내에서 서로 다른 성능 요구사항을 갖는 응용 프로그램 간 우선 순위를 지정하는 등의 네트워크 정책을 명세하며, 이러한 네트워크 정책은 일반적으로 패킷 스케줄러에 의해 적용된다.

[0005] 한편, 오늘 날의 10Gbps 이상의 고속 네트워크 인터페이스 카드(Network Interface Cards, NIC)는 다중 큐(Multiqueue)를 지원한다. 단일 큐를 지원하는 NIC에서는 패킷 스케줄링을 통한 네트워크 정책 적용이 서버 내 qdisc에서 일어났으나, 다중 큐 지원 NIC(MQ-NIC)가 장착된 서버에서는 NIC가 최종적인 네트워크 정책 적용 지점이 된다.

[0006] 그러나, 오늘날의 MQ-NIC는 큐당 공정성을 보장하기 위한 라운드 로빈(Round-Robin, RR) 패킷 스케줄러만을 지원한다. 다양한 패킷 스케줄러를 지원하는 NIC 하드웨어를 설계할 수도 있으나, 이는 근본적으로 과도한 교체 비용이 발생하며, 실제 배치까지 상당한 시간이 필요하다는 단점이 있다.

선행기술문헌

특허문헌

[0008] (특허문헌 0001) KR 10-1489414 B1

(특허문헌 0002) KR 10-0817676 B1

(특허문헌 0003) US 9225659 B2

발명의 내용

해결하려는 과제

- [0009] 이에, 본 발명의 기술적 과제는 이러한 점에서 착안된 것으로 본 발명의 목적은 하드웨어 수정 없이 소프트웨어 방법만으로 다중 큐 NIC를 장착한 네트워크 서버 내에서 다양한 네트워크 정책 적용 방법을 제공하는 것이다.
- [0010] 본 발명의 다른 목적은 상기 다중 큐를 지원하는 네트워크 정책 적용 방법을 수행하기 위한 컴퓨터 프로그램이 기록된 기록 매체를 제공하는 것이다.
- [0011] 본 발명의 또 다른 목적은 상기 다중 큐를 지원하는 네트워크 정책 적용 방법을 수행하기 위한 장치를 제공하는 것이다.

과제의 해결 수단

- [0013] 상기한 본 발명의 목적을 실현하기 위한 일 실시예에 따른 다중 큐를 지원하는 네트워크 정책 적용 방법은, 네트워크 관리자가 제출한 네트워크 정책 내용을 기초로, 모듈 내부의 테넌트 매핑 배열, 테넌트 가중치 배열, 시작 큐 인덱스 배열 및 포트 번호 배열과 변수들을 초기화하는 단계; 초기화된 배열로부터 획득한 테넌트 ID, 테넌트 가중치, 테넌트의 시작 큐 인덱스 및 현재의 큐를 추적하는 인덱스를 기초로, 각 테넌트에게 할당된 가중치만큼의 네트워크 인터페이스 카드(Network Interface Cards, NIC) 큐를 할당하여 테넌트 간 네트워크 격리를 보장하는 단계; 및 테넌트 내 응용 간의 우선순위를 다르게 하기 위해 우선순위가 높은 응용 패킷을 Qdisc의 머리에 삽입하여 테넌트 내 응용 간의 네트워크 격리를 보장하는 단계;를 포함한다.
- [0014] 본 발명의 실시예에서, 상기 테넌트 간 네트워크 격리를 보장하는 단계는, 테넌트의 현재의 큐를 추적하는 인덱스에 해당하는 값이 해당 테넌트의 가중치 이상인 경우, 현재 큐 인덱스에 해당하는 값을 0으로 초기화할 수 있다.
- [0015] 본 발명의 실시예에서, 상기 테넌트 간 네트워크 격리를 보장하는 단계는, 현재 처리 중인 패킷이 속한 소켓의 큐 매핑 메타데이터를 상기 테넌트의 현재의 큐를 추적하는 인덱스에 해당하는 값과 시작 큐 인덱스를 합한 값으로 갱신할 수 있다.
- [0016] 본 발명의 실시예에서, 상기 테넌트 간 네트워크 격리를 보장하는 단계는, 다음 패킷을 위해 현재의 큐를 추적하는 인덱스를 갱신하는 단계를 더 포함할 수 있다.
- [0017] 본 발명의 실시예에서, 상기 테넌트 내 응용 간의 네트워크 격리를 보장하는 단계는, 현재 사용 중인 패킷의 포트가 포트 번호 배열에 포함되어 있다면 현재 처리 중인 패킷이 속한 소켓의 우선순위 메타데이터 값을 1로 설정하여 높은 우선순위를 부여하고, 그 이외의 경우 우선순위 메타데이터 값을 0으로 설정하여 낮은 우선순위를 부여할 수 있다.
- [0018] 본 발명의 실시예에서, 다중 큐를 지원하는 네트워크 정책 적용 방법은, 상기 현재 처리 중인 패킷이 속한 소켓의 큐 매핑 메타데이터 값과 상기 우선순위 메타데이터 값을 기초로 현재 사용 중인 패킷의 인큐 여부를 결정하는 단계를 더 포함할 수 있다.
- [0019] 본 발명의 실시예에서, 상기 테넌트 매핑 배열은 현재 처리 중인 패킷이 어느 테넌트 소유의 패킷인지 구분할 수 있도록 CPU 코어 별로 할당되어 있는 테넌트의 인덱스를 표기하고, 상기 테넌트 가중치 배열은 테넌트 별 네트워크 스케줄링 가중치를 저장하고, 상기 시작 큐 인덱스 배열은 테넌트 가중치 배열 값을 기반으로 각 테넌트 별로 점유하는 NIC 큐 인덱스를 계산하여 저장하고, 상기 포트 번호 배열은 특정 테넌트가 높은 우선순위를 부여하고 싶은 응용이 있다면 해당 응용이 사용하는 포트 번호를 배열 형태로 저장할 수 있다.
- [0020] 상기한 본 발명의 다른 목적을 실현하기 위한 일 실시예에 따른 컴퓨터로 판독 가능한 저장 매체에는, 다중 큐를 지원하는 네트워크 정책 적용 방법을 수행하기 위한 컴퓨터 프로그램이 기록되어 있다.
- [0021] 상기한 본 발명의 또 다른 목적을 실현하기 위한 일 실시예에 따른 다중 큐를 지원하는 네트워크 정책 적용 장

치는, 네트워크 관리자가 제출한 네트워크 정책 내용을 기초로, 모듈 내부의 테넌트 매핑 배열, 테넌트 가중치 배열, 시작 큐 인덱스 배열 및 포트 번호 배열과 변수들을 초기화하는 정책 초기화부; 상기 정책 초기화부로부터 획득한 테넌트 ID, 테넌트 가중치, 테넌트의 시작 큐 인덱스 및 현재의 큐를 추적하는 인덱스를 기초로, 각 테넌트에게 할당된 가중치만큼의 네트워크 인터페이스 카드(Network Interface Cards, NIC) 큐를 할당하여 테넌트 간 네트워크 격리를 보장하는 플랫폼 정책 시행부; 및 테넌트 내 응용 간의 우선순위를 다르게 하기 위해 우선 순위가 높은 응용 패킷을 Qdisc의 머리에 삽입하여 테넌트 내 응용 간의 네트워크 격리를 보장하는 계층 정책 시행부;를 포함한다.

[0022] 본 발명의 실시예에서, 상기 플랫폼 정책 시행부는, 테넌트의 현재의 큐를 추적하는 인덱스에 해당하는 값이 해당 테넌트의 가중치 이상인 경우, 현재 큐 인덱스에 해당하는 값을 0으로 초기화하고, 현재 처리 중인 패킷이 속한 소켓의 큐 매핑 메타데이터를 상기 테넌트의 현재의 큐를 추적하는 인덱스에 해당하는 값과 시작 큐 인덱스를 합한 값으로 갱신하고, 다음 패킷을 위해 현재의 큐를 추적하는 인덱스를 갱신할 수 있다.

[0023] 본 발명의 실시예에서, 상기 계층 정책 시행부는, 현재 사용 중인 패킷의 포트가 포트 번호 배열에 포함되어 있다면 현재 처리 중인 패킷이 속한 소켓의 우선순위 메타데이터 값을 1로 설정하여 높은 우선순위를 부여하고, 그 이외의 경우 우선순위 메타데이터 값을 0으로 설정하여 낮은 우선순위를 부여할 수 있다.

[0024] 본 발명의 실시예에서, 상기 다중 큐를 지원하는 네트워크 정책 적용 장치는, 상기 현재 처리 중인 패킷이 속한 소켓의 큐 매핑 메타데이터 값과 상기 우선순위 메타데이터 값을 기초로 현재 사용 중인 패킷의 인큐 여부를 결정할 수 있다.

[0025] 본 발명의 실시예에서, 상기 테넌트 매핑 배열은 현재 처리 중인 패킷이 어느 테넌트 소유의 패킷인지 구분할 수 있도록 CPU 코어 별로 할당되어 있는 테넌트의 인덱스를 표기하고, 상기 테넌트 가중치 배열은 테넌트 별 네트워크 스케줄링 가중치를 저장하고, 상기 시작 큐 인덱스 배열은 테넌트 가중치 배열 값을 기반으로 각 테넌트 별로 점유하는 NIC 큐 인덱스를 계산하여 저장하고, 상기 포트 번호 배열은 특정 테넌트가 높은 우선순위를 부여하고 싶은 응용이 있다면 해당 응용이 사용하는 포트 번호를 배열 형태로 저장할 수 있다.

발명의 효과

[0027] 이와 같은 다중 큐를 지원하는 네트워크 정책 적용 방법에 따르면, 새로운 하드웨어 개발에 필요한 비용 및 교체 시간 없이, 현존하는 상용 MQ-NIC만으로도 여러 패킷 스케줄러를 사용하여 서버 내 네트워크 정책을 적용할 수 있다.

도면의 간단한 설명

[0029] 도 1은 본 발명의 일 실시예에 따른 다중 큐를 지원하는 네트워크 정책 적용 장치를 포함한 네트워크 서버의 개념도이다.

도 2는 본 발명의 플랫폼 정책 시행(Flat policy enforcement)의 의사코드(pseudocode)의 예시를 보여주는 도면이다.

도 3은 본 발명의 계층 정책 시행(Hierarchical Policy Enforcement)의 의사코드(pseudocode)의 예시를 보여주는 도면이다.

도 4는 본 발명의 패킷 인큐 결정의 의사코드(pseudocode)의 예시를 보여주는 도면이다.

도 5는 본 발명이 종래 기술에 비하여 각 테넌트에게 할당해주는 대역폭을 보여주는 실험 결과이다.

도 6은 본 발명이 종래 기술에 비하여 높은 우선순위를 가져야 하는 트래픽의 지연시간을 보여주는 실험 결과이다.

발명을 실시하기 위한 구체적인 내용

[0030] 후술하는 본 발명에 대한 상세한 설명은, 본 발명이 실시될 수 있는 특정 실시예를 예시로서 도시하는 첨부 도면을 참조한다. 이들 실시예는 당업자가 본 발명을 실시할 수 있기에 충분하도록 상세히 설명된다. 본 발명의 다양한 실시예는 서로 다르지만 상호 배타적일 필요는 없음이 이해되어야 한다. 예를 들어, 여기에 기재되어 있는 특정 형상, 구조 및 특성은 일 실시예에 관련하여 본 발명의 정신 및 범위를 벗어나지 않으면서 다른 실시예로 구현될 수 있다. 또한, 각각의 개시된 실시예 내의 개별 구성요소의 위치 또는 배치는 본 발명의 정신 및 범

위를 벗어나지 않으면서 변경될 수 있음이 이해되어야 한다. 따라서, 후술하는 상세한 설명은 한정적인 의미로서 취하려는 것이 아니며, 본 발명의 범위는, 적절하게 설명된다면, 그 청구항들이 주장하는 것과 균등한 모든 범위와 더불어 첨부된 청구항에 의해서만 한정된다. 도면에서 유사한 참조부호는 여러 측면에 걸쳐서 동일하거나 유사한 기능을 지칭한다.

- [0031] 이하, 도면들을 참조하여 본 발명의 바람직한 실시예들을 보다 상세하게 설명하기로 한다.
- [0032] 도 1은 본 발명의 일 실시예에 따른 다중 큐를 지원하는 네트워크 정책 적용 장치를 포함한 네트워크 서버의 개념도이다.
- [0033] 본 발명에 따른 다중 큐를 지원하는 네트워크 정책 적용 장치(10, 이하 장치)는 다중 큐를 지원하는 네트워크 인터페이스 카드(Network Interface Cards, NIC)를 장착한 컴퓨터 서버 장치에서 다양한 패킷 스케줄링 방법을 표현하는 패킷 데이터 처리 장치이다.
- [0034] 도 1을 참조하면, 본 발명에 따른 장치(100)는 네트워크 서버(1) 내에서 TCP/IP 스택(300)과 Qdisc(500) 사이에서 넷필터(Netfilter) 모듈의 형태로 작동한다. Qdisc(500)는 네트워크 인터페이스 카드 내 큐(queue)들과 일대일 매핑하여 작동한다.
- [0035] 본 발명에 따른 장치(100)는 정책 초기화(Policy initialization), 플랫 정책 시행(Flat policy enforcement) 및 계층 정책 시행(Hierarchical Policy Enforcement)의 3 단계를 수행하며, 각각 정책 초기화부(110), 플랫 정책 시행부(130) 및 계층 정책 시행부(150)를 통해 수행할 수 있다.
- [0036] 본 발명의 상기 장치(100)는 다중 큐를 지원하는 네트워크 정책 적용을 수행하기 위한 소프트웨어(애플리케이션)가 설치되어 실행될 수 있으며, 상기 정책 초기화부(110), 상기 플랫 정책 시행부(130) 및 상기 계층 정책 시행부(150)의 구성은 상기 장치(100)에서 실행되는 상기 다중 큐를 지원하는 네트워크 정책 적용을 수행하기 위한 소프트웨어에 의해 제어될 수 있다.
- [0037] 상기 장치(100)는 별도의 단말이거나 또는 단말의 일부 모듈일 수 있다. 또한, 상기 정책 초기화부(110), 상기 플랫 정책 시행부(130) 및 상기 계층 정책 시행부(150)의 구성은 통합 모듈로 형성되거나, 하나 이상의 모듈로 이루어 질 수 있다. 그러나, 이와 반대로 각 구성은 별도의 모듈로 이루어질 수도 있다.
- [0038] 상기 장치(100)는 이동성을 갖거나 고정될 수 있다. 상기 장치(100)는, 서버(server) 또는 엔진(engine) 형태일 수 있으며, 모듈(module), 디바이스(device), 기구(apparatus), 단말-terminal), UE(user equipment), MS(mobile station) 등 다른 용어로 불릴 수 있다.
- [0039] 상기 장치(100)는 운영체제(Operation System; OS), 즉 시스템을 기반으로 다양한 소프트웨어를 실행하거나 제작할 수 있다. 상기 운영체제는 소프트웨어가 장치의 하드웨어를 사용할 수 있도록 하기 위한 시스템 프로그램으로서, 본 발명에서는 리눅스 계열일 수 있다.
- [0040] 상기 정책 초기화부(110)는 네트워크 관리자가 제출한 네트워크 정책 내용을 기초로, 모듈 내부의 테넌트 매핑 배열, 테넌트 가중치 배열, 시작 큐 인덱스 배열 및 포트 번호 배열과 변수들을 초기화한다.
- [0041] 상기 테넌트 매핑 배열은 현재 처리 중인 패킷이 어느 테넌트 소유의 패킷인지 구분할 수 있도록 CPU 코어 별로 할당되어 있는 테넌트의 인덱스를 표기한다.
- [0042] 상기 테넌트 가중치 배열은 테넌트 별 네트워크 스케줄링 가중치를 저장하고, 상기 시작 큐 인덱스 배열은 테넌트 가중치 배열 값을 기반으로 각 테넌트 별로 점유하는 NIC 큐 인덱스를 계산하여 저장한다.
- [0043] 상기 포트 번호 배열은 특정 테넌트가 높은 우선순위를 부여하고 싶은 응용이 있다면 해당 응용이 사용하는 포트 번호를 배열 형태로 저장한다.
- [0044] 상기 플랫 정책 시행부(130)는 상기 정책 초기화(110)부로부터 획득한 테넌트 ID, 테넌트 가중치, 테넌트의 시작 큐 인덱스 및 현재의 큐를 추적하는 인덱스를 기초로, 각 테넌트에게 할당된 가중치만큼의 네트워크 인터페이스 카드(Network Interface Cards, NIC) 큐를 할당하여 테넌트 간 네트워크 격리를 보장한다.
- [0045] 이를 위해, 상기 플랫 정책 시행부(130)는 테넌트의 현재의 큐를 추적하는 인덱스에 해당하는 값이 해당 테넌트의 가중치 이상인 경우, 현재 큐 인덱스에 해당하는 값을 0으로 초기화한다.
- [0046] 또한, 상기 플랫 정책 시행부(130)는 현재 처리 중인 패킷이 속한 소켓의 큐 매핑 메타데이터를 상기 테넌트의 현재의 큐를 추적하는 인덱스에 해당하는 값과 시작 큐 인덱스를 합한 값으로 갱신하고, 다음 패킷을 위해 현재

의 큐를 추적하는 인덱스를 갱신한다.

- [0047] 상기 계층 정책 시행부(150)는 테넌트 내 응용 간의 우선순위를 다르게 하기 위해 우선순위가 높은 응용 패킷을 Qdisc의 머리에 삽입하여 테넌트 내 응용 간의 네트워크 격리를 보장한다.
- [0048] 상기 계층 정책 시행부(150)는 현재 사용 중인 패킷의 포트가 포트 번호 배열에 포함되어 있다면 현재 처리 중인 패킷이 속한 소켓의 우선순위 메타데이터 값을 1로 설정하여 높은 우선순위를 부여하고, 그 이외의 경우 우선순위 메타데이터 값을 0으로 설정하여 낮은 우선순위를 부여한다.
- [0049] 본 발명에 따른 장치(100)는 상기 플랫폼 정책 시행부(130)로부터 전달받은 상기 현재 처리 중인 패킷이 속한 소켓의 큐 매핑 메타데이터 값과 상기 계층 정책 시행부(150)에서 전달받은 상기 우선순위 메타데이터 값을 기초로 현재 사용 중인 패킷의 인큐 여부를 결정한다.
- [0050] 상기 정책 초기화부(110), 상기 플랫폼 정책 시행부(130) 및 상기 계층 정책 시행부(150)가 각각 수행하는 정책 초기화(Policy initialization), 플랫폼 정책 시행(Flat policy enforcement) 및 계층 정책 시행(Hierarchical Policy Enforcement)의 구체적인 단계는 이하 자세하게 설명한다.
- [0051] 1단계인 상기 정책 초기화(Policy initialization)단계는 네트워크 관리자가 제출한 네트워크 정책 내용을 바탕으로 상기 장치(100) 내부의 배열과 변수들을 초기화한다.
- [0052] 상기 장치(100)는 테넌트 매핑 배열, 테넌트 가중치 배열, 시작 큐 인덱스 배열 및 포트 번호 배열의 4개 배열을 사용한다.
- [0053] 상기 테넌트 매핑 배열은 CPU 코어 별로 할당되어있는 테넌트의 인덱스를 표기한다. 소켓버퍼(SKB)의 sender_cpu 메타데이터를 통해 현재 처리중인 패킷을 담당하는 CPU 코어 인덱스를 알 수 있다. 따라서, 상기 장치(100)가 관리하는 테넌트 매핑 배열 값을 참조하면 현재 처리중인 패킷이 어느 테넌트 소유의 패킷인지 구분할 수 있다.
- [0054] 상기 테넌트 가중치 배열은 테넌트 별 네트워크 스케줄링 가중치를 저장하는 배열이다.
- [0055] 상기 시작 큐 인덱스 배열은 테넌트 가중치 배열 값을 기반으로 각 테넌트별로 점유하는 NIC 큐 인덱스를 계산하여 저장한다. 예를 들어, 두 테넌트 0과 1이 각각 2와 1의 가중치를 가지고 있다면 시작 큐 인덱스 배열의 값은 {0,0,1}이 된다.
- [0056] 상기 포트 번호 배열은 특정 테넌트가 자신이 사용하는 응용 중 높은 우선순위를 부여하고 싶은 응용이 있다면 해당 응용이 사용하는 포트 번호를 배열 형태로 저장한다.
- [0057] 2단계인 플랫폼 정책 시행(Flat policy enforcement) 단계에서는 테넌트 간 네트워크 격리를 보장한다. 상기 장치(100)는 각 테넌트에게 할당된 가중치만큼의 NIC 큐를 할당한다. 이를 통해, 우회적으로 가중치 기반 패킷 스케줄러를 표현할 수 있다.
- [0058] 도 2는 이 단계의 의사코드를 나타낸 것이다. 현재 처리 중인 패킷의 큐 번호를 결정하기 위해서는 테넌트 ID, 테넌트 가중치, 테넌트의 시작 큐 인덱스, 그리고 현재의 큐 인덱스를 추적하는 인덱스가 필요하다(도 2의 줄 1-5). 이러한 값들은 1단계에서 초기화한 배열들에 의해서 얻어질 수 있다(도 2의 줄 7-8).
- [0059] 참고로 c_i 는 1단계에서 0으로 초기화된다. 패킷들을 할당된 큐들에 고르게 분배하기 위해서 c_i 는 해당 값이 w_i 를 초과할 때마다 초기화된다(도 2의 줄 11-13).
- [0060] 상기 장치(100)는 현재 처리 중인 패킷이 속한 소켓의 queue_mapping 메타데이터를 본 단계에서 얻어진 인덱스 값을 시작 큐 인덱스와 합한 값으로 갱신하며(도 2의 줄 14), 다음 패킷을 위해 추적용 인덱스 역시 갱신한다(도 2의 줄 15).
- [0061] 3단계인 계층 정책 시행(Hierarchical Policy Enforcement) 단계에서는 테넌트 내 응용 간 네트워크 격리를 보장한다. 테넌트 내 응용 간의 우선순위를 달리하기 위해서 상기 장치(100)는 우선순위가 높은 응용 패킷을 qdisc의 꼬리가 아닌 머리에 삽입한다.
- [0062] 참고로, 상기 장치(100)에서는 qdisc과 NIC 큐들이 일대일 매핑되어 동기화되어 있다. qdisc의 머리에 삽입하면 현재 qdisc에 버퍼되어 있는 다른 패킷들을 건너뛰게 되므로 우선순위 스케줄링 없이도 우선순위 스케줄링을 우회적으로 표현할 수 있다.

- [0063] 도 3은 계층 정책 시행 과정의 의사코드를 나타낸다. 상기 장치(100)는 현재 사용 중인 패킷의 포트가 포트 번호 배열에 포함되어 있다면 SKB의 priority 메타데이터 값을 1로 설정하여 높은 우선순위를 부여한다(도 3의 줄 5-6). 그렇지 않다면 priority 메타데이터 값은 0으로 설정되어 낮은 우선순위를 부여받게 된다(도 3의 줄 7-8).
- [0064] 도 4는 상기 장치(100)가 어떻게 패킷 인큐 결정을 내리는지를 보여준다. 상기 장치(100)를 거치면서 처리된 값을 바탕으로 qdisc는 우회적으로 다양한 패킷 스케줄링을 표현하게 되고, qdisc의 패킷들은 NIC로 내려감으로써 목표는 달성된다.
- [0065] 다중 큐 스케줄링을 지원하는 multiq qdisc는 도 2에서 얻어진 NIC 큐 인덱스값을 queue_mapping 메타데이터를 참조하여 구한다(도 4의 줄 5). 또한, 도 3에서 얻어진 priority 메타데이터 값을 구한다(도 4의 줄 6).
- [0066] 이를 통해 qdisc는 priority 값에 따라 꼬리 인큐 혹은 헤드 인큐를 수행하게 되며 패킷은 항상 queue_mapping에 해당되는 큐에 삽입된다(도 4의 줄 5-8).
- [0067] 본 발명의 유효성을 평가하기 위해 리눅스 상에서 상기 장치(100)를 넷필터(Netfilter) 모듈로 구현하고 Intel 82599 10GbE NIC를 장착한 서버 두 대를 10Gbps 스위치에 연결한 테스트베드를 구성하여 iperf와 sockperf를 사용한 실험을 진행하였다. 비교 대상은 현재 리눅스 커널에서 지원하는 NIC 큐 할당방법인 XPS를 그 대상으로 삼았다.
- [0068] 도 5는 테넌트 3개가 존재하는 경우를 가정한 것으로 각 테넌트 1,2,3은 순서대로 1,2,3의 가중치를 갖는다. 각 테넌트는 0초부터 5초마다 각각 8, 16, 32개의 플로우를 발생시킨다.
- [0069] 도 5를 참조하면, 상기 장치(100)가 각 테넌트에게 할당한 가중치만큼 대역폭을 보장해줄 수 있다. 반면 XPS는 많은 수의 플로우를 가지는 테넌트에게 더 많은 대역폭을 할당해 줄 수 있다. 이는 XPS는 테넌트별 트래픽을 격리하지 못하기 때문이다.
- [0070] 도 6은 두 테넌트가 동일한 가중치를 가지고 iperf로 두 테넌트 모두 8개의 플로우를 발생시켰을 때, 높은 우선순위를 가져야하는 sockperf 트래픽이 겪은 지연 시간을 나타낸 것이다. 상기 장치(100)는 헤드 인큐를 통해 테넌트 내 응용간 네트워크 격리를 보장하므로 XPS보다 낮은 지연을 보임을 알 수 있다.
- [0071] 또한, 도 6의 오른쪽은 각 테넌트가 대역폭을 얼마나 균등하게 공유해주는지 보여주는 Jain's fairness index를 나타낸 것으로 TONIC은 1에 해당하는 완벽한 공정성을 보여줄 수 있다.
- [0072] 이와 같은, 다중 큐를 지원하는 네트워크 정책 적용 방법은 애플리케이션으로 구현되거나 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다.
- [0073] 상기 컴퓨터 판독 가능한 기록 매체에 기록되는 프로그램 명령어는 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 분야의 당업자에게 공지되어 사용 가능한 것일 수도 있다.
- [0074] 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다.
- [0075] 프로그램 명령어의 예에는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드도 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.
- [0076] 이상에서는 실시예들을 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

산업상 이용가능성

- [0078] 본 발명은 리눅스 운영체제를 사용하는 네트워크 서버에 적용 가능하며, 나아가 유무선 네트워킹/통신 장비를 활용하는 ICT 시장 전반에 걸쳐 적용이 가능하다.

부호의 설명

1: 네트워크 서버

100: 다중 큐를 지원하는 네트워크 정책 적용 장치

110: 정책 초기화부

130: 플랫 정책 시행부

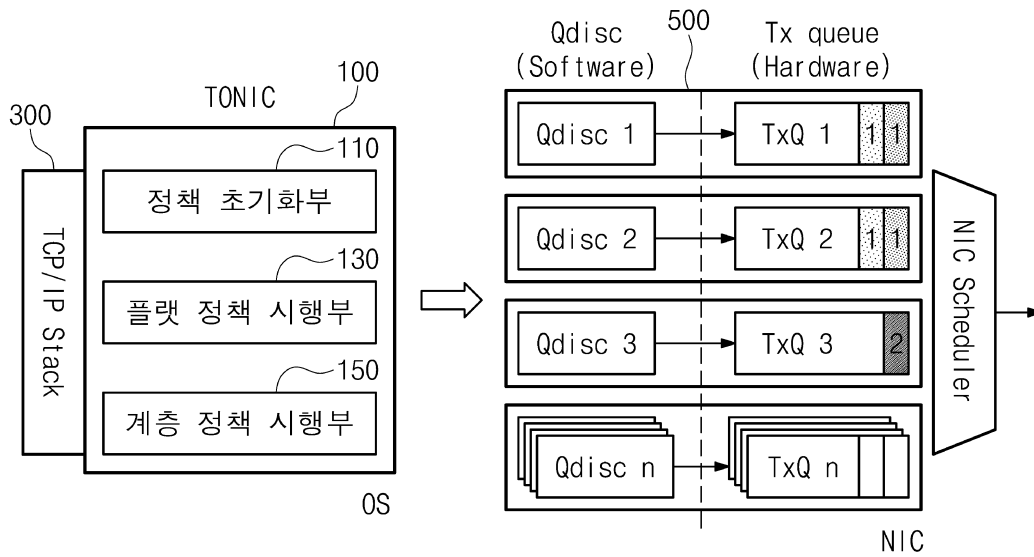
150: 계층 정책 시행부

300: TCP/IP 스택

500: Qdisc

도면

도면1



도면2

Algorithm 1 Tx Queue Assignment in TONIC

```

1: Inputs:
2:  $i$ : An integer indicating the ID of a tenant
3:  $w_i$ : The weight of tenant  $i$ 
4:  $s$ : The index of the first Tx queue assigned to tenant  $i$ 
5:  $c_i$ : The current Tx queue index of tenant  $i$ 
6:
7:  $i \leftarrow \text{TenantMappingArray}[\text{skb} \rightarrow \text{sender\_cpu}]$ 
8:  $w_i \leftarrow \text{WeightArray}[i]$ 
9:  $s \leftarrow \text{StartIdxArray}[i]$ 
10:
11: if  $c_i \geq w_i$  then                                ▷  $c_i$  exceeds the index range of tenant  $i$ 
12:    $c_i \leftarrow 0$  ▷ Initialize the index to circulate Tx queue index based on
   the assigned index range for tenant  $i$ 
13: end if
14:  $\text{skb} \rightarrow \text{queue\_mapping} \leftarrow s + c_i$            ▷ Assign Tx queue index
15:  $c_i \leftarrow c_i + 1$                                ▷ Update current index

```

도면3

Algorithm 2 Traffic Prioritization in TONIC

```

1: Inputs:
2:  $p_s$ : The source port number in TCP header
3:  $p_d$ : The destination port number in TCP header
4:
5: if  $p_s \in \text{PortNumberArray}$  or  $p_d \in \text{PortNumberArray}$  then
6:    $\text{skb} \rightarrow \text{priority} \leftarrow 1$  ▷ Update metadata to indicate that the
   packet belongs to the high priority application
7: else
8:    $\text{skb} \rightarrow \text{priority} \leftarrow 0$  ▷ Tag low priority in metadata for the other
   packets
9: end if

```

도면4

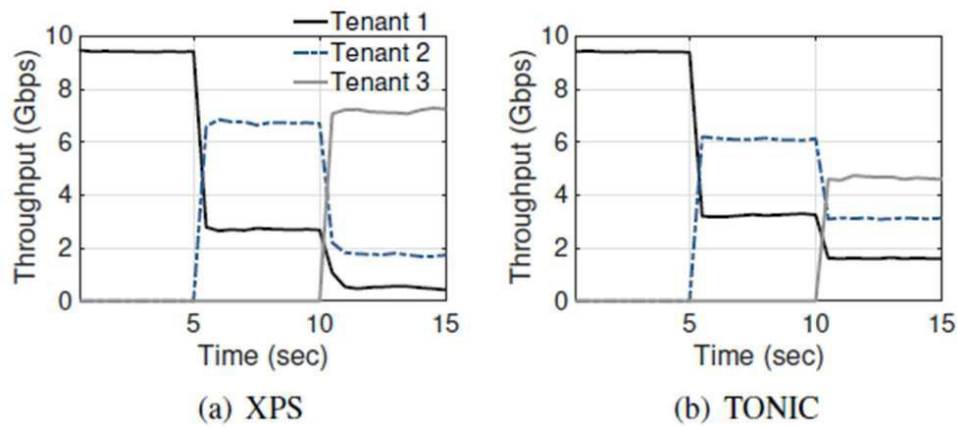
Algorithm 3 Enqueueing Decision in TONIC

```

1: Inputs:
2:  $i$ : An integer indicating Tx queue index
3:  $j$ : An integer denoting packet priority
4:
5:  $i \leftarrow \text{skb} \rightarrow \text{queue\_mapping}$  ▷ Assigned in Alg. 1
6:  $j \leftarrow \text{skb} \rightarrow \text{priority}$  ▷ Assigned in Alg. 2
7: if  $j = 1$  then ▷ High priority?
8:   Enqueue(head,  $i$ ) ▷ Insert the packet into the head of queue  $i$  for
   traffic prioritization
9: else
10:  Enqueue(Tail,  $i$ ) ▷ Insert the packet into the tail of queue  $i$  as usual
11: end if

```

도면5



도면6

