IT@인텔 백서

인텔 IT
데이터 센터 효율성
소프트웨어 정의 네트워크
9월 2014

기업에서 소프트웨어 정의 네트워크 채택

그러면 프로그래밍 가능한 인터페이스를 통해 네트워크를 가상화하여 인텔 응용 프로그램 개발자에게 보다 향상된 지원을 제공할 수 있습니다.

개요

인텔 IT는 네트워크 및 네트워크 서비스를 주문형으로 프로비저닝하기 위해 소프트웨어 정의 네트워크(SDN)를 도입하고 있습니다. 그러면 프로그래밍 가능한 인터페이스를 통해 네트워크를 가상화하여 내부 고객에 해당하고 빠른 속도의 민첩한 개발 환경에서 작업하는 인텔 응용 프로그램 개발자에게 보다 향상된 지원을 제공할 수 있습니다. 이 개발자들은 네트워크를 구성하고 서비스를 구축하면서 만들어진 병목 현상을 처리할 필요 없이 네트워크 리소스에 액세스할 수 있어야 합니다.

SDN을 구현하면 데이터 센터에서 가상 머신(VM)을 통해 비즈니스 가치를 극대화하고, 다음과 같은 추가 이점을 누릴 수 있습니다.

- 네트워크 프로비저닝 시간 단축
- 셀프 서비스 환경에서 네트워크 구축 프로세스 간소화
- 향상된 네트워크 관리 효율성을 통해 서비스 비용 절감

최근, SDN 산업은 SDN 구성 요소에서 괄목할 만한 성능 개선과 확장성 증대를 이루었습니다. 지난 2년 동안 우리는 SDN 배포에 따른 이점을 판단하는 기준으로 다음 요소를 평가해 왔습니다.

- SDN 구현을 보장하는 사용 사례 정의
- 노드 간 데이터 흐름을 알기 위해 다양한 SDN 아키텍처 테스트 및 분석
- 셀프 서비스 환경에서 네트워크를 구축하지만 네트워크 엔지니어링 배경 지식이 부족한 직원의 역할 및 책임 업데이트

우리는 지속적으로 SDN을 도입해 우리의 가상화 환경 전체에 배포하면서 SDN 구성요소와 아키텍처가 네트워크 서비스의 장애물을 어떻게 해결해 주는지 평가하고 이를 바탕으로 고객이 응용 프로그램을 보다 신속하게 배포할 수 있도록 지원할 계획입니다.

Sridhar Mahankali
인텔 IT 클라우드 네트워크 엔지니어

Sanjay Rungta
선임 책임 엔지니어, 인텔 IT

차례

개요 1

비즈니스 과제 2

- 데이터 센터에서 클라우드 사용 진화: 스토리지, 컴퓨팅 및 네트워크 2
- 프로그래밍 가능 가상화 네트워크를 통해 얻을 수 있을 것으로 예상되는 회사의 이득 3

솔루션 4

- SDN 아키텍처 선택 4
- 오버레이 네트워크 아키텍처 5
- 보안 아키텍처 변경 6
- 역할과 책임 변경 7
- 예비 결과 7

다음 단계 7

결론 7

관련 정보 7

약어 7

IT@인텔

IT@인텔 프로그램은 전 세계 IT 전문가를 우리 조직 내 동료 전문가들과 연결해 얻은 정보와 방법 및 전략을 공유합니다. 우리의 목표는 단순합니다. 비즈니스 가치를 창출하고 IT 부서가 경쟁력을 갖출 수 있는 인텔 IT 성공 사례를 공유하는 것입니다. 자세한 정보가 필요하면 지금 www.intel.com/IT 를 방문하거나 현지 인텔 담당자에게 문의하십시오.

비즈니스 과제

인텔 IT는 64개의 데이터 센터에서 약 55,000개 서버를 운영하면서 104,000명의 직원을 지원하고 있습니다.[1] 지난 10여 년간 인텔 IT는 사무실, 기업, 서비스 데이터 센터 환경의 서버들을 가상화해왔습니다.

2000년에는 가상 머신(VM)을 프로비저닝하는 데 90일 이상이 소요되었습니다. 오늘날에는 주문에 따라 VM을 프로비저닝하여 응용 프로그램을 개발하는 사업부(BU)의 출시 기간을 단축할 수 있습니다. 사무실, 엔터프라이즈 및 서비스 데이터 센터의 작업 부하 중 85% 이상이 VM에서 실행됩니다. 그러나 네트워크 리소스는 여전히 주문형 프로비저닝과 관련한 병목 현상을 겪고 있는데, 이러한 현상은 특히 사무실과 엔터프라이즈 데이터 센터에서 두드러집니다. 따라서 우리는 네트워크 서비스 프로비저닝 시간을 단축시켜 회사 업무를 보다 원활하게 지원할 수 있는 솔루션을 찾고 있습니다.

데이터 센터에서 클라우드 사용 진화: 스토리지, 컴퓨팅 및 네트워크

표 1의 연대기에 보여진 것처럼, 인텔의 가상화 사용 역사는 2000년으로 거슬러 올라갑니다. 2009년 경에는 엔터프라이즈 컴퓨팅 환경 가상화 비율이 12%에 불과했습니다. VM 프로비저닝에 90일 이상이 소요되던 이 시기에는 가상화 환경의 신뢰성도 편차가 컸습니다.

2009년에는 우리 사무실 및 엔터프라이즈 데이터 센터 서버 환경 대부분을 가상화하기 시작했습니다. 불과 3년만인 2012년에 엔터프라이즈 컴퓨팅 환경 중 75%가 가상화되었으며, 컴퓨팅 프로비저닝은 주문형이었고 가용성은 99.7-99.9%였습니다. 엔터프라이즈 컴퓨팅 중 85%가 가상화된 2014년 이후에도 꾸준히 컴퓨팅, 스토리지 및 네트워크 서비스를 가상화하고 이러한 서비스를 주문형으로 프로비저닝할 계획입니다. 이는 인텔의 응용 프로그램 개발 생태계를 크게 확대시켜 나가는 데 도움이 될 것입니다.

SDN 모델 이전에 사용하던 레거시 네트워크 프로비저닝 모델의 경우, 네트워크 프로비저닝에 필요한 총 작업 시간은 4-6시간이었습니다. 여기에 작업 시작과 중지, 네트워크 프로비저닝 요청의 백로그, 그 외 복잡한 상황들이 더해지면 네트워크 프로비저닝 과정에 13일이 걸릴 수도 있었습니다. 작업 지연은 보안 설정, 로컬 및 글로벌 부하 분산, 도메인 이름 시스템(DNS) 구성과 같은 다른 응용 프로그램 랜딩 작업에도 순차적으로 영향을 미칩니다.

요청량 증가에 대처하기 위해 우리는 주문형 네트워크를 동적으로 프로비저닝할 수 있도록 사무실 및 엔터프라이즈 데이터 센터 서버 환경에 프로그래밍 가능 네트워크

[1] "데이터 센터"를 정의하기 위해 인텔에서는 "서버 및 기타 인프라 구성 요소가 설치되는 100 제곱 피트 이상의 공간"이라는 IDC의 데이터 센터 크기 분류를 이용합니다.

표 1. 2000년에 소수의 가상 머신(VM)으로 출발하여 2010년에는 사설 클라우드로, 오늘날에는 하이브리드 클라우드로 발전했습니다.

	2000-2009년	2010년	2012	2014년 이후
	기존 호스팅	**메인스트림 가상화**	**인텔 클라우드 1.0**	**하이브리드 클라우드 2.0 컨버지드 클라우드**
가상화된 서버	12%	42%	75%	85%
프로비저닝	90일 이상	10일	주문형 컴퓨팅	주문형 컴퓨팅, 네트워크 및 스토리지
서비스 요청	설명서	설명서	일부 주문형	완벽한 셀프 서비스
신뢰성	가변 서버 신뢰성	99.7% VM 신뢰성	99.7%-99.9% 가용성	99.9% 신뢰성 가능

구성 요소를 배포하는 방식으로 전환해야 합니다. 가상 LAN(VLAN)과 가상 라우팅 및 포워딩(VRF)을 사용하면 실제 네트워크의 최상위에 위치하는 특정 네트워크 리소스를 가상화할 수 있지만, 이러한 기술 자체는 자동화와 셀프 서비스에 필요한 프로그래밍 기능은 가지고 있지 않습니다.

프로그래밍 가능 가상화 네트워크를 통해 얻을 수 있을 것으로 예상되는 회사의 이득

프로그래밍 가능 네트워크 서비스 구성 요소를 데이터 센터 네트워크에 통합하면 민첩성이 개선되고 프로비저닝 프로세스가 간소화되며 총 소유 비용(TCO)이 감소합니다.

- **민첩성.** 네트워크 구성과 부하 분산 장치 및 방화벽과 같은 서비스의 프로비저닝은 응용 프로그램 연결 프로세스에 병목 현상을 일으킬 수 있습니다. 프로그래밍 가능 가상화 네트워크를 사용하면 중앙 제어식 콘솔에서 네트워크 리소스를 동적으로 프로비저닝하거나 API를 사용하여 추가적인 사용자 지정 자동화를 구축할 수 있습니다. 우리는 네트워크 관리자의 작업 없이 데이터 센터 네트워크 서비스의 대부분을 주문형으로 프로비저닝할 수 있을 것으로 예상합니다.
- **단순성.** 주문형으로 프로비저닝되는 가상화 네트워크는 네트워크 관리자의 부담을 덜어주고 고객이 응용 프로그램 개발을 위한 자체 네트워크를 쉽게 생성 및 관리할 수 있도록 지원합니다.
- **TCO.** 네트워크 장비의 구성을 자동화할 수 있는 가상화 네트워크를 활용하면 개별 네트워크 구성 요소를 수동으로 구성해야 할 필요성이 감소합니다. 우리는 이 전사적 관점의 프로그래밍 가능 네트워크가 네트워크 운영 비용을 줄여줄 것이라 예상합니다.

솔루션

그림 1에 보여진 소프트웨어 정의 네트워크 연결(SDN) 기능을 활용하면 엔터프라이즈 데이터 센터 네트워크를 서버 가상화 방식과 유사하게 가상화할 수 있습니다. SDN은 네트워크의 SDN 지원 부분을 위한 중앙 집중식 프로그래밍 가능 인터페이스를 제공하기 위해 데이터 플레인에서 제어 플레인을 분리합니다. SDN은 네트워크 구성 요소를 프로그래밍 가능하도록 만들어, 더 새로운 다중 테넌트 모델과 분산 보안 액세스 제어를 제공하고 확장성을 개선하고 네트워크 응용 프로그램을 통한 신속한 서비스 혁신을 가능하게 합니다.

우리의 엔터프라이즈 데이터 센터 네트워크에서 설계 및 채택해 온 OpenStack* 기반 사설 클라우드 환경은 SDN 요소 통합을 위한 프레임워크를 제공합니다. OpenStack의 Neutron API는 오픈 소스 및 독점 프로그래밍 가능 네트워크 구성 요소를 사용하여 네트워크 서비스를 프로비저닝, 변경 및 삭제할 수 있습니다.

SDN 기능은 산업 분야에서 비교적 생소한 기술로, 발전을 거듭하고 있습니다. 인텔 IT는 SDN에 대한 평가를 통해 네트워크 프로비저닝 시간을 단축시키고 네트워크 서비스 프로비저닝 프로세스를 간소화하고 서비스 비용을 절감할 수 있을지 여부를 확인하고 있습니다. 평가는 SDN 컨트롤러의 성능 테스트와 분석을 중심으로 이루어집니다. SDN 컨트롤러는 각 테넌트의 네트워크에 대한 논리 뷰를 제시하고 네트워크 하드웨어와 테넌트 사이의 흐름 제어를 관리함으로써 SDN의 중추 역할을 합니다.

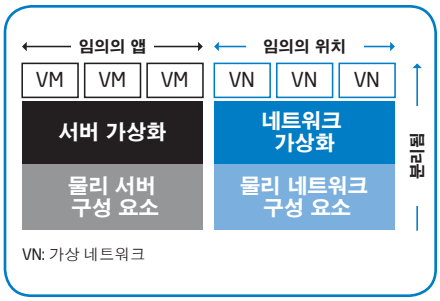

그림 1. 서버 가상화가 물리 서버에 연결되지 않은 여러 개의 가상 머신(VM)을 생성하는 것처럼, 소프트웨어 정의 네트워크(SDN)인 네트워크 가상화는 물리 네트워크 구성 요소에 연결되지 않은 여러 개의 가상 네트워크를 생성합니다.

네트워크의 진화: 하드웨어 고착화에서 개방형 소프트웨어로

컴퓨터 네트워크가 처음 보급될 당시 소수의 공급자만이 하드웨어와 소프트웨어(구동하는 실리콘 칩 포함)를 모두 제작했습니다. 따라서 여러 공급자들의 네트워크를 프로비저닝하려는 기업은 하드웨어, 소프트웨어 및 부대 서비스를 구매해야 했습니다.

산업이 성숙함에 따라 일부 공급자는 하드웨어 구성 요소를 독점 개발하는 노력을 기울이기 시작했습니다. 그 결과 소프트웨어 시장은 개방되고 제3자가 소프트웨어 기능을 개발하고 확장할 수 있게 되었습니다.

소프트웨어 정의 네트워크가 출현하면서 회사는 소프트웨어 개발을 조절하고 개방성을 더할 수 있게 되었습니다. SDN은 네트워크 프로비저닝 프로세스를 간소화 및 자동화하는 동시에 공급자 고착화 현상을 없애줄 것입니다.

SDN 아키텍처 선택

평가 결과 SDN은 두 가지 방식으로 구현할 수 있다는 것이 확인되었습니다. 첫 번째 방법은 전환-포워딩 테이블에 대한 직접 액세스 및 조작을 허용하는 공개 표준 통신 인터페이스인 OpenFlow*를 사용하는 것입니다. 두 번째 방법은 기존 네트워크의 최상위에 구축된 노드와 논리 링크로 구성된 오버레이 네트워크를 사용하는 것입니다. 이 방법은 기존 물리 네트워크에서 사용할 수 없는 네트워크 서비스를 생성하며 각 논리 링크는 기반 네트워크에 존재하는 수많은 물리 링크로 구성될 수 있습니다.

OpenFlow 포워딩 테이블을 지원하려면 OpenFlow에 새 하드웨어가 필요합니다. 오버레이 네트워크는 추가 하드웨어를 요구하지 않으므로, 새 장비에 대한 투자가 필요하지 않습니다. 또한 OpenFlow는 버전이 다양하며 모든 SDN 컨트롤러와 네트워크 하드웨어 공급자가 동일한 버전을 지원하는 것은 아닙니다. OpenFlow는 데이터 흐름이 일시적이고 네트워크가 흐름 변화에 기반한 재프로그래밍을 요구하는 경우에 적합합니다. 그러나 하드웨어 환경은 일정하지 않고 소스 및 대상 노드 사이의 데이터 흐름은 일반적으로 설정 후 예측 가능하므로, 오버레이 모델이 OpenFlow 모델에 비해 더 큰 가치를 제공합니다.

우리는 2012년부터 2013년 중반까지 오버레이 기반 아키텍처부터 OpenFlow 기반 아키텍처에 이르는 다중 아키텍처를 갖는 SDN 컨트롤러를 테스트했습니다. 네트워크 대역폭과 성능 기준점이 테스트 기준에 포함되었습니다. 그림 2는 인텔의 테스트 시스템입니다.

테스트는 3단계로 실시되었습니다.

- **1단계: SDN 컨트롤러 성능 문제.** 2012년 초반에 실시된 이 테스트 단계에서는 SDN 개념의 유효성을 입증하고 테스트한 모든 SDN 컨트롤러에서 공통적으로 발견되는 몇 가지 성능 문제를 파악했습니다. 이 기술을 도입하려면 먼저 업계에서 이러한 문제들을 해결해야 합니다. 예를 들어, 오버레이 네트워크가 없는 10기가비트 이더넷(GbE) 네트워크에 대한 한 테스트 시나리오에서 얻은 결과를 보면 2개의 하이퍼바이저에 존재하는 2개의 VM의 처리량이 9.39 Gbps인 것으로 나타났습니다. 동일한 설정에서 오버레이 네트워크를 사용했을 때의 처리량은 3.3 Gbps에 불과했습니다. 하이퍼바이저에 VM을 추가해도 성능은 개선되지 않았으며, 이는 오버레이 드라이버가 최적화되지 않았음을 나타냅니다.
- **2단계: 확장성 문제.** 2012년 후반부터 2013년 초반까지 실시된 이 테스트 단계에서는 처리량이 개선되었으나 확장성 문제에 부딪혔습니다. 수많은 VM과 흐름을 처리할 수 있도록 아키텍처를 확장할 수 없음을 발견했습니다. 각 통신에 대해 새 흐름이 시작되고 만료되지 않았으며, 터널 수도 많은 수의 VM에서 증가했습니다.
- **3단계: 흐름 최적화.** 2013년 중반에 실시된 최종 테스트 단계에서는 제품 공급자들이 흐름 확산을 해결했으며 소프트웨어가 성숙되고 안정화되었음이 확인되었습니다. 다양한 공급자들이 이 문제를 다르게 해결했지만, 각 VM은 다른 VM에 브로드캐스팅할 필요 없이 통신을 유지할 수 있는 것으로 나타났습니다. 흐름 연결은 소스 및 대상 주소를 토대로 최적화되었습니다.

오버레이 네트워크 아키텍처

SDN 오버레이 네트워크 구성은 데이터 센터 내부의 데이터 흐름을 최적화하는 데 가장 유용합니다. 2013년 하반기에는 생산 환경에 오버레이 네트워크를 배포했습니다. 그림 3은 SDN 아키텍처의 상위 수준 관점을 보여줍니다. 이 아키텍처의 세 가지 기본 구성 요소는 가상 스위치를 통해 VM에 연결되는 SDN 컨트롤러, 게이트웨이 노드, 하이퍼바이저 노드입니다.

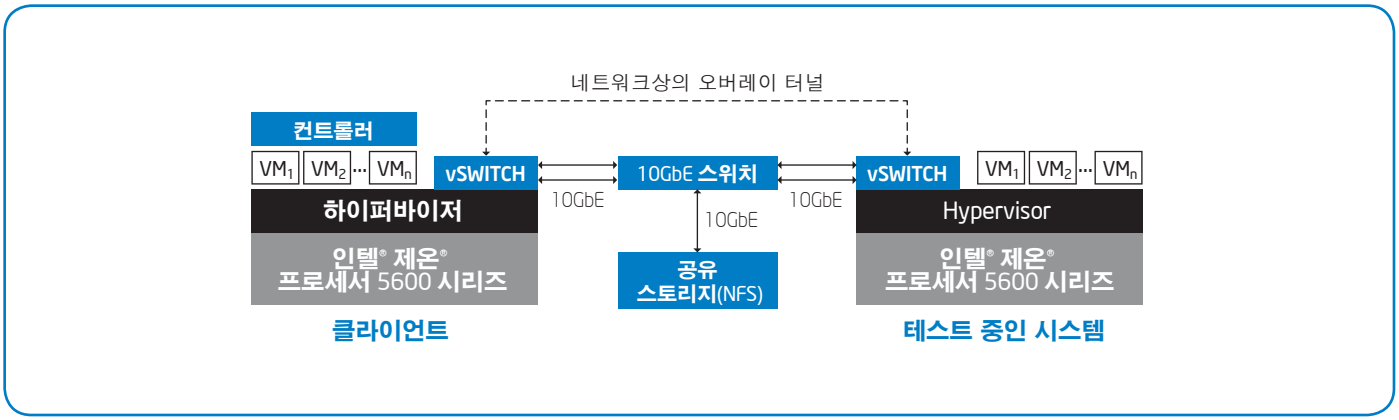

그림 2. 인텔의 테스트 구성. 소프트웨어 정의 네트워크 평가 시 오버레이 기반 아키텍처를 사용하여 SDN 컨트롤러 성능, 확장성 문제, 흐름 최적화를 테스트 및 모니터링했습니다.

SDN 컨트롤러

SDN 컨트롤러는 하드웨어로부터 네트워크를 추상화하여 SDN 네트워크를 가능케 합니다. 컨트롤러(실제로는 소프트웨어 응용 프로그램은 하드웨어의 구성 요소가 아님)는 응용 프로그램과 네트워크 장치 간 통신을 관리합니다. 대규모 구현의 경우, SDN 컨트롤러 클러스터를 형성하는 2개 이상의 SDN 컨트롤러가 존재할 수 있습니다.

하나의 콘솔에서 네트워크 전체를 보여주며 통신 흐름의 최적화와 구성 자동화를 특징으로 하는 지능형 네트워킹이 중앙 집중식 콘솔 접근법을 통해 만들어졌습니다.

- **데이터 흐름 최적화.** 기존 네트워크는 통신 소스에서 그 대상에 이르는 경로가 한 개입니다. SDN 컨트롤러는 하나의 흐름에 대해 여러 개의 경로를 지정하고 흐름의 트래픽을 여러 개의 노드로 분할할 수 있습니다. SDN 컨트롤러는 소스와 대상 노드를 기반으로 특정 데이터 흐름에 대한 네트워크 경로를 최적화합니다. 이러한 기능은 네트워크의 성능과 확장성을 높여줍니다.
- **구성 자동화.** 기존의 수동식 장치별 네트워크 구성과 달리 SDN 컨트롤러는 네트워크 장치 구성을 자동화함으로써 구성 시간을 절약하고 구성의 정확성과 일관성을 높여줍니다. 이 접근법은 네트워크 조건이 변경될 경우 구성을 쉽게 조정할 수 있다는 이점이 있습니다. 기본적으로 SDN 컨트롤러는 전체 네트워크 아키텍처가 하나의 기기인 것처럼 관리할 수 있도록 도와줍니다.
- **네트워크 전체 보기.** SDN 컨트롤러의 콘솔을 통해 네트워크 관리자는 전체 네트워크를 전반적으로 파악할 수 있으므로, 의사 결정과 관리 효율이 개선됩니다.

SDN 컨트롤러의 프로그래밍 가능 인터페이스는 기존 네트워크에 비해 네트워크 트래픽에 대한 네트워크 관리자의 제어 능력을 높여줍니다. 예를 들어, 특정 서버에 대한 인바운드 트래픽은 인텔의 보안 정책에 따라 방화벽을 통과해야 합니다. 그러나 아웃바운드 트래픽은 보안 위협을 노출하지 않으므로 방화벽을 반드시 통과할 필요는 없습니다. 기존 네트워크에서는 이 세분화된 제어 수준에 도달하기 어렵습니다. SDN을 사용하면 네트워크 엔지니어가 아닌 직원도 컨트롤러가 아웃바운드 트래픽을 방화벽으로 리디렉션하도록 쉽게 프로그래밍할 수 있습니다. 이러한 정책은 세션, 사용자, 장치 및 응용 프로그램 수준에서 SDN 컨트롤러를 통해 적용됩니다.

게이트웨이/서비스 노드

게이트웨이 노드는 SDN 컨트롤러에 의해 구성 및 관리되는 SDN 지원 클라우드 패브릭에 VLAN의 물리 네트워크 패브릭을 통합하는 방법을 제공합니다. 게이트웨이 기능은 두 환경을 통합하고, SDN 지원 하이퍼바이저가 비 SDN 지원 네트워크와 통신할 수 있도록 지원합니다. 서비스 노드는 VM들 간 브로드캐스트, 멀티케스트 및 흐름 설정을 처리합니다.

하이퍼바이저 노드

하이퍼바이저 노드는 VM을 호스팅합니다. 그림 3과 같이, 각 하이퍼바이저 노드는 가상전환 모듈을 실행하고 SDN 컨트롤러는 이곳에서 테넌트에 대한 가상 네트워크를 구축합니다. 하나의 하이퍼바이저가 여러 개의 가상 네트워크의 VM을 호스팅할 수

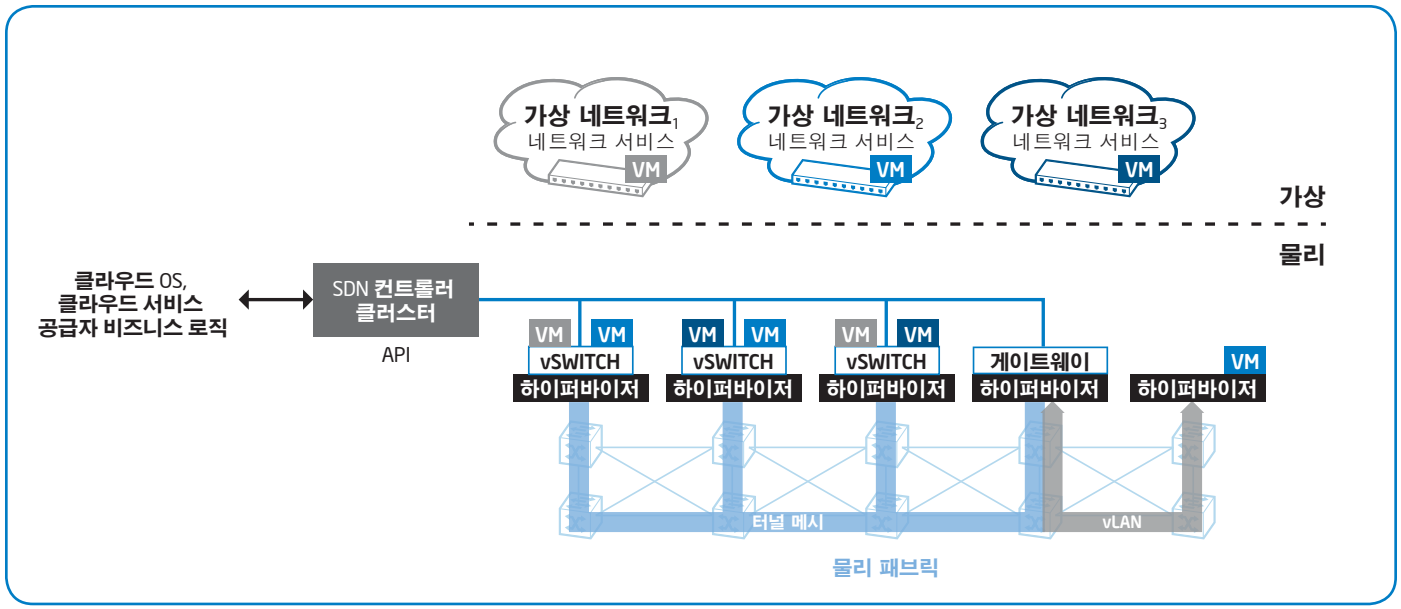

그림 3. 인텔의 오버레이 소프트웨어 정의 네트워크(SDN) 아키텍처는 SDN 컨트롤러 집합, 게이트웨이/서비스 노드, 하이퍼바이저 노드의 세 가지 기본 구성 요소로 구성됩니다. 기본적으로 SDN 컨트롤러는 전체 네트워크 아키텍처가 하나의 기기인 것처럼 관리할 수 있도록 지원합니다.

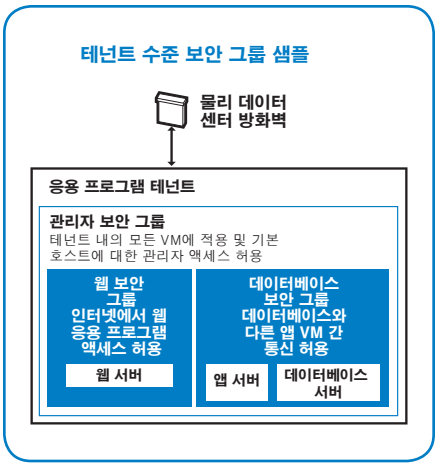

그림 4. 소프트웨어 정의 네트워크(SDN) 기능이 통합된 인텔의 OpenStack* 기반 사설 클라우드 환경의 보안 그룹 모델은 네트워크 액세스를 유연하게 제어할 수 있도록 지원합니다.

있습니다. 정책 제어는 하이퍼바이저와 VM 사이의 가상 스위치에 의해 관리됩니다.

보안 아키텍처 변경

소비자화 및 기타 업계 동향은 인터넷 연결 가상화 환경에 프로비저닝되고 있는 응용 프로그램의 증가에 기여해 왔습니다. 그 동안 우리는 네트워크 기반 분할과 네트워크 액세스 제어를 VM 보안을 위한 기본 장치로 사용해 왔습니다. 레거시 네트워크 프로비저닝 모델의 경우, 우리는 모놀리식 외부 방화벽 기기를 사용하여 VM 및 응용 프로그램에 세분화된 네트워크 액세스 제어 기능을 제공했습니다. 이러한 유형의 보안 모델은 확장성, 지원성, 위험성 문제를 내포했습니다. 예를 들어, 일부 환경에서는 외부 방화벽을 수천 개의 보안 규칙으로 강화해야 했기 때문에 보안 관리 효율성 문제가 발생하고 보안 위반 가능성이 증가했습니다.

SDN 기능이 통합된 OpenStack 기반 사설 클라우드 환경을 구축하면 더 분산적이고 테넌트 중심적인 보안 프레임워크로 전환할 수 있습니다. 각 응용 프로그램 테넌트는 인텔의 사설 클라우드 최상위에 위치하는 각각의 사설 가상 네트워크 환경으로 프로비저닝됩니다. 네트워크 액세스 제어는 보안 그룹을 통해 테넌트의 가상 네트워크 환경 내에서 관리됩니다.

보안 그룹은 기본적으로 각 테넌트 내부에서 전환할 수 있는 액세스 제어 규칙 모음으로 구성됩니다. 그림 4와 같이 여러 개의 보안 그룹을 VM에 적용하면 해당 VM에 대한 네트워크 보안을 정의할 수 있습니다.

호스트에서 강화된 응용 프로그램 테넌트 수준 보안 그룹을 이용하면 각 테넌트에 대해 관리 효율성이 더 우수한 액세스 제어 규칙 집합을 구현하고 이러한 규칙을 관리하는 셀프 서비스가 가능해집니다. 그러면 물리 방화벽 기기를 사용하여 데이터 센터 경계를 보호할 수 있고 관리 효율성이 더 우수하고 규모는 작은 보안 규칙 집합이 필요하게 됩니다.

역할과 책임 변경

이제 네트워크 인스턴스를 생성할 때 네트워크, 서버 및 컴퓨팅 팀의 책임을 힘들게 구분할 필요가 없습니다. SDN을 사용하면 이미 물리 네트워크 장치에 구현되어 있는 네트워크 기능을 가상 네트워크에 다시 구현할 수 있습니다. 따라서 SDN을 도입하려면 시스템 관리자와

실제 셀프 서비스 네트워크 사례

소프트웨어 정의 네트워크(SDN)는 네트워크 관리자가 아닌 직원들을 위한 셀프 서비스 활동을 프로비저닝하는 네트워크 서비스를 실현합니다. 예를 들어, 인텔의 IT 인큐베이터 프로그램 관리자인 Thomas Birch는 인텔 사업부(BU)가 웹 서비스를 론칭하는 데 사용할 수 있는 네트워크를 SDN을 사용해서 만듭니다.

이 스크린샷은 Birch가 만든 네트워크를 관리하는 데 사용되는 웹 인터페이스입니다. Birch는 네트워크 규칙을 정함으로써 인바운드 연결을 가상 머신(VM)에 설정합니다. Birch는 통신에 필요한 다른 VM은 무엇이고 그 이유는 무엇인지 검토하며, 모든 웹 인터페이스에서 네트워크 인스턴스를 배포 및 테스트하고 있습니다.

SDN을 도입하기 전 Birch는 모든 네트워크 서비스 프로비저닝을 한 명의 네트워크 관리자에게 요청해야 했습니다. 이제 Birch는 실시간으로 네트워크를 만들고 있습니다. Birch는 "SDN 이전에는 자동화에 한계가 있었고 설정 시에도 수동 조작이 많이 필요했습니다. 특정 포트를 개방해달라고 IT 엔지니어에게 요청해야 했고 저는 중간자에 불과했습니다. SDN은 제게 통제권을 주었습니다."라고 말합니다.

셀프 서비스 네트워크를 만들려는 직원은 웹 GUI를 사용하여 가상 네트워크 인스턴스를 프로비저닝합니다.

네트워크 관리자 간에 나뉘어 있던 기술을 결합해야 합니다.

이 격차를 없애기 위해 우리는 클라우드 시스템 관리 기능을 만들었습니다. 각 팀(네트워크, 서버, 컴퓨팅)의 대표들이 클라우드 환경의 개발과 운영 책임을 맡습니다. 이 대표들이 협력하고 기술을 공유합니다.

네트워크를 프로비저닝하는 직원의 학습 곡선은 존재하지만 네트워킹 백그라운드가 꼭 필요한 것은 아닙니다. 네트워크 엔지니어링 팀은 이 직원들이 기본 개념과 네트워크 보안 실무와 프로비저닝 GUI 가이드를 이해할 수 있도록 지원합니다. 우리는 방대한 네트워크 지식 없이도 고객이 네트워크 구성 요소를 주문형으로 쉽게 프로비저닝할 수 있게 지원하는 직관적인 응용 프로그램 환경 템플릿도 구축하고 있습니다.

예비 결과

서비스 프로비저닝 시간은 이미 개선되고 있습니다. SDN 컨트롤러의 네트워크 글로벌 보기와 SDN이 지원하는 셀프 서비스의 이점은 네트워크 서비스 요청에 대한 서비스 수준 계약(SLA)을 충족해야 하는 네트워크 관리자의 부담을 줄여주고 있습니다.

SDN은 네트워크 프로비저닝 프로세스를 간소화하고 있습니다. 네트워크를 생성하려면 약간의 교육이 필요하며, 특히 네트워킹 경험이 거의 없는 직원들의 경우 더 많은 교육이 필요합니다. 그러나 네트워크 엔지니어들의 지원을 받고 SDN 관리 응용 프로그램으로 작업하고 나면 직원들이 셀프 서비스 접근법의 이점을 활용할 수 있습니다. 민첩한 개발 작업에서 BU 지원을 방해하는 요인을 없애고 네트워크 배포에 대한 통제력과 실시간 배포 및 테스트 능력을 강화합니다.

초기 테스트에서는 SDN이 네트워크 관리 효율의 개선을 통해 서비스 비용을 절감하는 것으로 나타났습니다. 또한 현재의 인프라 투자를 유지하고 독점 하드웨어 및 전용 기기에 대한 의존도도 낮출 수 있습니다.

다음 단계

2014년에는 2013년의 초기 프로덕션 배포로부터 오버레이 네트워크 배포를 확대할 것입니다. 또한 오버레이 네트워크에 가치를 더해주는 OpenFlow 사용 방법을 모색할 예정입니다.

API를 사용하여 클라우드 환경 내에서 부하 분산 장치와 웹 응용 프로그램 방화벽과 같은 네트워크 서비스를 가상화할 계획입니다. CPU 오버헤드를 모니터링할 예정이며, 이러한 기능들은 하위 수준 하드웨어나 네트워크 인터페이스 컨트롤러(NIC)에 대한 프로세스 부하를 줄여주는 경우에 호스트에 추가될 것입니다.

하이퍼바이저 환경에서 SDN 환경으로의 마이그레이션은 많은 과제를 초래하지만, 이 마이그레이션을 간소화할 방법을 모색하고 있습니다.

모노리식 보안 모델에서 분산 보안 모델로 발전함에 따라 특히 호스트 제한의 관점에서 확장성을 모니터링해야 합니다. 또, 프로세스 부하를 최대한 줄일 계획입니다.

결론

SDN은 클라우드 환경을 위해 네트워크를 가상화하고 네트워크 엔지니어가 아닌 직원들도 오버레이 네트워크 구성 요소를 프로그래밍할 수 있도록 지원하므로 우리는 주문형 네트워크를 동적으로 프로비저닝할 수 있게 됩니다. 한 번 구성하는 데 며칠이 걸리던 네트워크 서비스를 이제 셀프 서비스 접근법을 통해 실시간으로 완료할 수 있습니다.

인텔은 이러한 SDN의 이점을 실현하고 있습니다.

- 더 빠른 서비스 프로비저닝
- 더 단순한 셀프 서비스 네트워크 생성
- 서비스 비용 감소

또한 현재 인프라에 대한 통제력이 강화되고 독점 하드웨어 및 전용 기기에 대한 의존도도 낮출 수 있습니다.

관련 정보

관련 항목에 대한 내용은 www.intel.com/IT를 참조하십시오.

- "비즈니스 혁신을 위한 인텔 IT의 데이터 센터 전략"
- "데이터 센터 네트워크 아키텍처를 10 기가비트 이더넷으로 업그레이드"

약어

BU	business unit(사업부)
DNS	domain name system (도메인 이름 체계)
GbE	gigabit Ethernet(기가비트 이더넷)
NIC	network interface controller (네트워크 인터페이스 컨트롤러)
SDN	software-defined networking (소프트웨어 정의 네트워크)
SLA	service-level agreement (서비스 수준 계약)
TCO	total cost of ownership (총 소유 비용)
VLAN	virtual LAN(가상 LAN)
VM	virtual machine(가상 머신)
VRF	virtual routing and forwarding (가상 라우팅 및 포워딩)

인텔 IT 모범 사례에 대한 자세한 내용은 www.intel.com/it를 참조하십시오.

성능 테스트에 사용된 소프트웨어와 워크로드는 인텔® 마이크로프로세서 성능 전용으로 최적화되었을 수 있습니다. SYSmark* 및 MobileMark*와 같은 성능 테스트는 특정 컴퓨터 시스템, 구성 요소, 소프트웨어, 운영 및 기능을 사용하여 측정됩니다. 해당 요소에 변경이 생기면 결과가 달라질 수 있습니다. 고려 중인 구매를 제대로 평가하려면 다른 제품과 결합하여 사용할 경우 해당 제품의 성능을 포함한 기타 정보 및 성능 테스트를 참고해야 합니다.

구성:

서버(테스트 대상 시스템)	모델: 인텔® S5520UR(Urbanna) ▪ 프로세서: 인텔® 제온® 프로세서 X5680, 3.33 GHz(2소켓, 6코어/소켓), 96 GB RAM, 하이퍼 스레딩 비활성화 ▪ BIOS: S5500.86B.01.00.0060 ▪ 네트워크 어댑터: 2 x 인텔® 이더넷 서버 어댑터 X520-2(10GbE 연결, 10GbE 스위치에 연결) ▪ 네트워크 드라이버: ixgbe 2.0.84.8.2-10vmw-NAPI ▪ 하이퍼바이저: VMware ESX* 5.1 RC(build 716794) ▪ 게스트 VM 구성: 1 vCPU, 4 GB RAM, RHEL 6.2(2.6.32-220.el6.x86_64), vmxnet3 및 ixgbevf, 24 VM ▪ 빅 스위치 컨트롤러(Corona) – 베타 릴리즈
클라이언트 구성	모델: 인텔® S5520UR(Urbanna) ▪ 프로세서: 인텔® 제온® 프로세서 X5670, 2.93 GHz(2소켓, 6코어/소켓), 96 GB RAM, 하이퍼 스레딩 비활성화 ▪ BIOS: S5500.86B.01.00.0060 ▪ 네트워크 어댑터: 2 x 인텔® 이더넷 서버 어댑터 X520-2(10GbE 연결, 10GbE 스위치에 연결) ▪ 네트워크 드라이버: ixgbe 2.0.84.8.2-10vmw-N ▪ 하이퍼바이저: VMware ESX* 5.1 RC(build 716794) ▪ 게스트 VM 구성: 1 vCPU, 4 GB RAM, RHEL 6.2(2.6.32-220.el6.x86_64), vmxnet3, 24 VM
네트워크 구성	모델: Extreme Networks Summit* X650 10GbE 스위치 ▪ 네트워크 연결: – 인텔® 이더넷 서버 어댑터 X520-2 10GbE 연결(서버에서) 2개 – 인텔® 이더넷 서버 어댑터 X520-2 10GbE 연결(클라이언트에서) 2개 – 10 기가비트 이더넷 컨트롤러 IX1-SFP+ 10GbE 연결(NetApp FAS6240) 1개
스토리지 구성	NetApp FAS6240 ▪ 버전 8.0.1 P1 ▪ 인텔® 제온® 프로세서 E5540, 2.53 GHz (8 proc), 48 GB RAM ▪ 512 GB의 PAM-II ▪ NFS 스토리지, 10GbE
애플리케이션	Netperf* 2.5
사용된 데이터 수집 도구	▪ Netperf에서 캡처된 처리량 ▪ VMware ESX* 호스트 상에서 "esxtop" 유틸리티 캡처 시스템 활용

이 용지에 제공되는 정보는 성격상 일반적인 정보로, 특정 지침을 위해 고안된 것이 아닙니다. 권장 사항(잠재적인 비용 절감 포함)은 인텔 경험을 기반으로 한 것이며 추정치에 불과합니다. 인텔은 다른 경우 더 나은 결과를 얻을 수 있다는 보증을 하지 않습니다.

이 문서의 정보는 인텔 제품과 함께 제공됩니다. 이 문서는 어떠한 지적 재산권에 대한 라이센스도 명시적 또는 묵시적이나 금반언에 의해 제공하지 않습니다. 인텔의 판매 약관에 명시된 경우를 제외하고 인텔은 어떠한 책임도 지지 않으며 특정 목적에의 적합성, 상품성 또는 특허권, 저작권 및 다른 지적 재산권 침해에 관한 책임이나 보증을 포함하여 인텔 제품의 판매 및 사용과 관련된 어떠한 명시적 또는 묵시적 보증도 부인합니다.

인텔, 인텔 로고, 인텔 제온, Look Inside. 및 Look Inside. 로고는 미국 및/또는 기타 국가에서 인텔사의 상표입니다.

*기타 명칭 및 브랜드는 해당 소유업체의 자산입니다.

Copyright © 2014 Intel Corporation. All rights reserved. 미국에서 인쇄 ♻재활용하십시오

0914/WWES/KC/PDF 330118-001US

intel®

Look Inside.™