**Intel DPDK가 NFV에서 꼭 필요한 이유는 무엇일까?**

**Intel DPDK**

* 왜 Intel의 DPDK가 최근 주목을 받을까?
* 그것은 NFV를 구현하는데 DPDK가 가장 중요한 역활을 하기 때문이다.
* 그럼 NFV부터 DPDK가 무슨 연관이 있는지 알아보았다.

**NFV?**

* NFV (Network Function Virtualization)는
  + 유럽 전기 통신 표준화기구 (ETSI : 유럽 전기 통신 전반에 관한 표준화기구)가 제창함
  + 통신망 서비스 회사에서 전용 네트워크 어플라이언스 장비를 대체하는 프로젝트 이다.
  + 즉, 전용 네트워크 어플라이언스 장비를 IA 서버의 가상 머신에서 동작하는 가상 어플라이언스로 대체하는 것을 말한다.
* "전용 어플라이언스 장비에서 제공하는 네트워크 장비"란
  + LTE 등의 무선 통신을 지원하는 통신 장치를 예를 들어 보자.
* LTE 백엔드의 구조를 보면 일반 IT 네트워크 장비와는 다르게 패킷을 처리하는 것을 알수 있다.
  + MME (Mobility Management Entity)
  + S-GW (Serving Gateway)
  + P-GW (Packet data network Gateway)
* LTE뿐만 아니라 기존의 3G 네트워크도 전문 네트워크 장비 제조업체가 하드웨어를 판매한다.
* 특정 하드웨어와 성능이 동일한 일반 서버에서 소프트웨어로 구현하자는 것이 NFV입니다.
* ETSI의 목표는 가상 머신 소프트웨어를 구현하는 것입니다.
* Telco망에서 사용할 가상 머신 소프트웨어를 NFV라고 합니다.
* NFV는 OpenStack처럼 클라우드 기반으로 구축되면,
* 운용자의 간편한 조작(Web UI와 같은) 버튼 하나로 새로운 통신 장비 군을 추가 할 수 있습니다.

**Intel DPDK 의한 네트워크 어플리케이션의 구현**

* NFV는 기존 장비를 동일한 성능에서라는 전제 조건이 붙어 있습니다.
* 일반 IA 서버에 Linux를 설치하고
  + 라우터 또는 방화벽과 같은 네트워크의 기본 기능을 이용하는 것은 간단하지만
* Telco 망에서 IA 서버가 전용 통신 장비 정도의 성능을 제공해야만 NFV가 가능합니다.
* 지금은 IA 서버의 네트워크를 이용하는 것은 하드웨어 성능이 한계가 있습니다.
  + 원래 Linux 네트워크 처리 전용으로 개발 된 OS라는 것은 없습니다.
  + Linux 커널이 제공하는 네트워크 기능은
  + 어디 까지나 상위의 응용 프로그램을 지원하는 인프라 기능입니다.
  + 네트워크 처리가 빨라도 그만큼 응용 프로그램의 실행 속도가 빠르다는 것을 의미하지 않습니다.
  + 즉 다양한 요구 사항 응용 프로그램을 실행하는 범용 OS가 다양한 네트워크 프로토콜을 지원해야합니다.
* 정리하면
  + 현재 Linux는 IA 서버의 네트워크 성능을 한계까지 사용하는 설계가 아니다.
  + Linux 커널은 네트워크 처리만을 전문하는 것은 아니다.
  + 범용 OS로 다양한 네트워크 프로토콜에 대응하는 필요가있다.
* 이런 점을 개선하기 위해서 Intel에서 만든 것이 DPDK입니다.
  + Intel DPDK (Data Plane Development Kit)
  + 현재 Linux에서 네트워크 처리에 특화된 애플리케이션을 작성하기위한 소프트웨어 라이브러리 집합입니다.
  + Intel DPDK를 이용하면 사용자는 Linux 커널 대신에 네트워크 패킷을 처리하는 응용 프로그램을 직접 만들 수 있습니다.

**DPDK를 사용하는 응용 프로그램의 구조**

* Intel DPDK를 사용하는 응용 프로그램은 Linux 커널을 거치지 않고 네트워크 카드에 도착한 패킷을 직접 받아 처리합니다.
  + 예를 들어,
  + 응용 프로그램을 특정 CPU 코어에 할당하여 가장 높은 우선 순위로 실행하면
  + Linux에 네트워크 처리 전용의 CPU 코어가 제공됩니다.
  + 따라서 CPU 코어의 성능을 한계까지 끌어 내
  + IA 서버의 네트워크 성능을 비약적으로 향상시킬 수 있습니다
  + 첫째는 커널의 IP Stack은 패스하고 어플리케이션이 패킷 자체를 통제하겠다는거고,
  + 둘째는 interrupt 없이 context switching 없이 패킷 처리를 하겠다는거죠



