

H/W 제조사를 위한 클라우드 네이티브

김병진

sirmd@outlook.com

목차

- 하드웨어를 위한 클라우드 네이티브 개요
 - 소프트웨어 기반 클라우드 네이티브 이슈
 - 하드웨어 기반 가속
- 클라우드 네이티브를 위한 하드웨어 구성
 - 제조사 솔루션
 - 오픈 솔루션
- 발전방향
 - Cloud Native 의 CNI
 - 하드웨어 기반 소프트웨어 개발

네이티브 클라우드

- Native Cloud(또는 Cloud Native)
 - 클라우드 컴퓨팅 모델을 사용하는 것을 기본으로 설계된 시스템과 서비스 또는 앱(애플리케이션)
 - 클라우드에서 시스템 및 서비스등을 운영하기 위해 설계된 시스템의 등장
 - 기존 시스템들 과의 구분을 위해 "클라우드 네이티브(네이티브 클라우드)" 라는 용어를 사용하기 시작
- Cloud Native
 - Cloud Native 개발 : 클라우드에서 사용을 전제로 사용되는 애플리케이션을 개발
 - 애플리케이션을 신속하게 구축(개발)하고 업데이트하면서 소프트웨어 품질을 개선하고 위험성을 낮추는 접근 방식

네이티브 클라우드

- Cloud Native 핵심
 - 앱 또는 애플리케이션을 만드는 방법과 배포
 - Public Cloud에 배포하는 것을 의미
 - 클라우드 전용 앱 또는 애플리케이션 개발(신규 개발) 구축
 - 기존 앱 또는 애플리케이션의 경우 최적화
 - 어떠한 환경에서도 연결하는 작업을 가속화(?) 할 수 있는 방법
- 목표
 - 비즈니스 요구사항의 변화 속도 대응
 - 사용자들이 요구사항을 빠르게 적용 및 배포(제공)

네이티브 클라우드

- **CNCF = Cloud Native Computing Foundation**

- Cloud Native 오픈소스 기술들을 추진하고 관리하는 단체
- 대표적으로 Kubernetes 와 Prometheus 프로젝트
- CNCF는 2015 년에 Linux Foundation 산하에 설립
- Kubernetes 를 Google에서 기증
- 컨테이너화되는 오픈소스 소프트웨어 스택을 사용하는 것을 의미
 - 플리케이션의 각 부분은 자체 컨테이너에 패키징
 - 동적 오케스트레이션을 통해 각 부분이 적극적으로 스케줄링 및 관리되어 리소스 사용률을 최적화
 - 마이크로서비스 지향성을 통해 애플리케이션의 전체적인 민첩성과 유지 관리 편의성 증가

- **일반적인 의미**

- 클라우드 컴퓨팅 을 활용 하여 "퍼블릭, 프라이빗 및 하이브리드 클라우드와 같은 현대적이고 동적 인 환경에서 확장 가능한 애플리케이션을 구축하고 실행" 하는 소프트웨어 개발 방식 (위키피디아 참조)

소프트웨어 기반 클라우드 네이티브 이슈

- 소프트웨어 기반 클라우드
 - 가상화 기술 위에서 사용되는 소프트웨어 개발
 - 응용 프로그램의 격리
 - 네트워크의 대역폭과 관리
 - 다른 가상 머신 또는 물리적 컴퓨터의 영향력
- 클라우드 네이티브 시스템의 속성(CNCF)
 - 애플리케이션 또는 프로세스는 소프트웨어 컨테이너에서 분리된 단위로 실행
 - 프로세스는 리소스 사용을 개선하고 유지보수 비용을 줄이기 위해 중앙 오케스트레이션 프로세스에 의해 관리
 - 애플리케이션 또는 서비스(마이크로서비스)는 명시적으로 설명된 종속 항목과 느슨하게 결합

클라우드 시스템의 원본-원류

- 클라우드 시스템의 핵심 – 분산 컴퓨팅
 - 분산 컴퓨팅 환경과 오류 : 분산 컴퓨팅의 오류(Fallacies of Distributed Computing)
 - 분산 시스템의 설계와 증명
 - 네트워크의 신뢰와 안전
 - 네트워크의 동질적
 - 대기 시간 = 0
 - 대역폭은 무한대
 - 토폴로지는 불변하고 관리자는 한 명
 - 전송 비용 = 0
 - 해결 : Kubernetes 및 Istio

클라우드의 데이터 플레인 - Review

데이터 플레인

- 클라우드에서 처리하는 데이터는 증가한다.
- 증가한 만큼 대역폭 확보
- 클라우드 센터에서 네트워크 선을 증설(?) 한다.
- 하드웨어 장비에서 받는 것도 한계 = 소프트웨어로 증설된 네트워크를 구성 및 관리 => 그래서 SDX 기술들이 필요

하드웨어 및 가상화에서 처리되는 기술

- 가상화에서 처리되는 기술이 발전 : PCI – Passthrough 방식
- SR-IOV : Single-Root IO Virtualization 방식
- SR-IOV와 PCI – Passthrough 방식 모두 가상 머신의 가상 포트가 호스트의 네트워크 인터페이스에 고정되는 단점
- 가상화 네트워크의 장점을 모두 소실되는 단점이 발생
- OVS-DPDK / NIC Offloading의 데이터 전송 가속 솔루션 (Data Plane Acceleration)

클라우드 네이티브를 위한
하드웨어 구성

CLOUD NATIVE 애플리케이션

- Cloud Native 개발
 - 데브옵스 (DevOps)
 - 애자일 방법론
 - 마이크로서비스 아키텍처
 - 클라우드 플랫폼
 - 쿠버네티스 및 도커와 같은 컨테이너
- 플랫폼 서비스(PaaS) 모델을 사용
 - 조직 내 인력과 이들의 협업 프로세스를 자동화하는 것에서 시작
 - DevOps를 도입하여 공통의 목적과 주기적인 피드백을 통해 개발팀과 운영팀의 협업을 지원
 - 결론 : 조직 문화의 변화

Cloud Native 기반 개발

- 데브옵스
 - 클라우드 네이티브 애플리케이션은 조직 내 인력과 이들의 협업 프로세스를 자동화
 - DevOps를 도입하여 공통의 목적과 주기적인 피드백을 통해 개발팀과 운영팀의 협업을 지원
- 컨테이너 (Container)
 - 애플리케이션 배포 유닛 및 독립적인 실행 환경을 제공
- 결론
 - DevOps 및 컨테이너 접근 방식에서는 하나의 대규모 릴리스 및 업데이트를 기다리는 것이 아니라 마이크로서비스처럼 여러 서비스가 탄력적으로 결합된 하나의 컬렉션으로 애플리케이션을 쉽게 출시

Cloud Native 기반 개발

- CI/CD (지속적인 통합과 제공)
 - 많은 조직이 실용적인 서비스 기반 아키텍처를 이용하여 계속해서 레거시 애플리케이션을 최적화
 - 최적화하는 지속적인 통합(Continuous Integration, CI)과 지속적인 제공(Continuous Delivery, CD) 및 완전히 자동화된 배포 운영 같은 DevOps 워크플로우를 통해 지원
- 마이크로서비스 아키텍처 (MSA)
 - 클라우드 네이티브 개발은 아키텍처의 모듈성, 탄력적인 결합, 그리고 독립적인 서비스에 중점
 - 비즈니스 역량을 구현하고 자체 프로세스를 실행하고 애플리케이션 프로그래밍 인터페이스(Application Programming Interfaces, API) 또는 메시징을 통해 커뮤니케이션
 - 커뮤니케이션은 서비스 메쉬 레이어를 통해 관리

클라우드 서비스에서의 데이터 증가

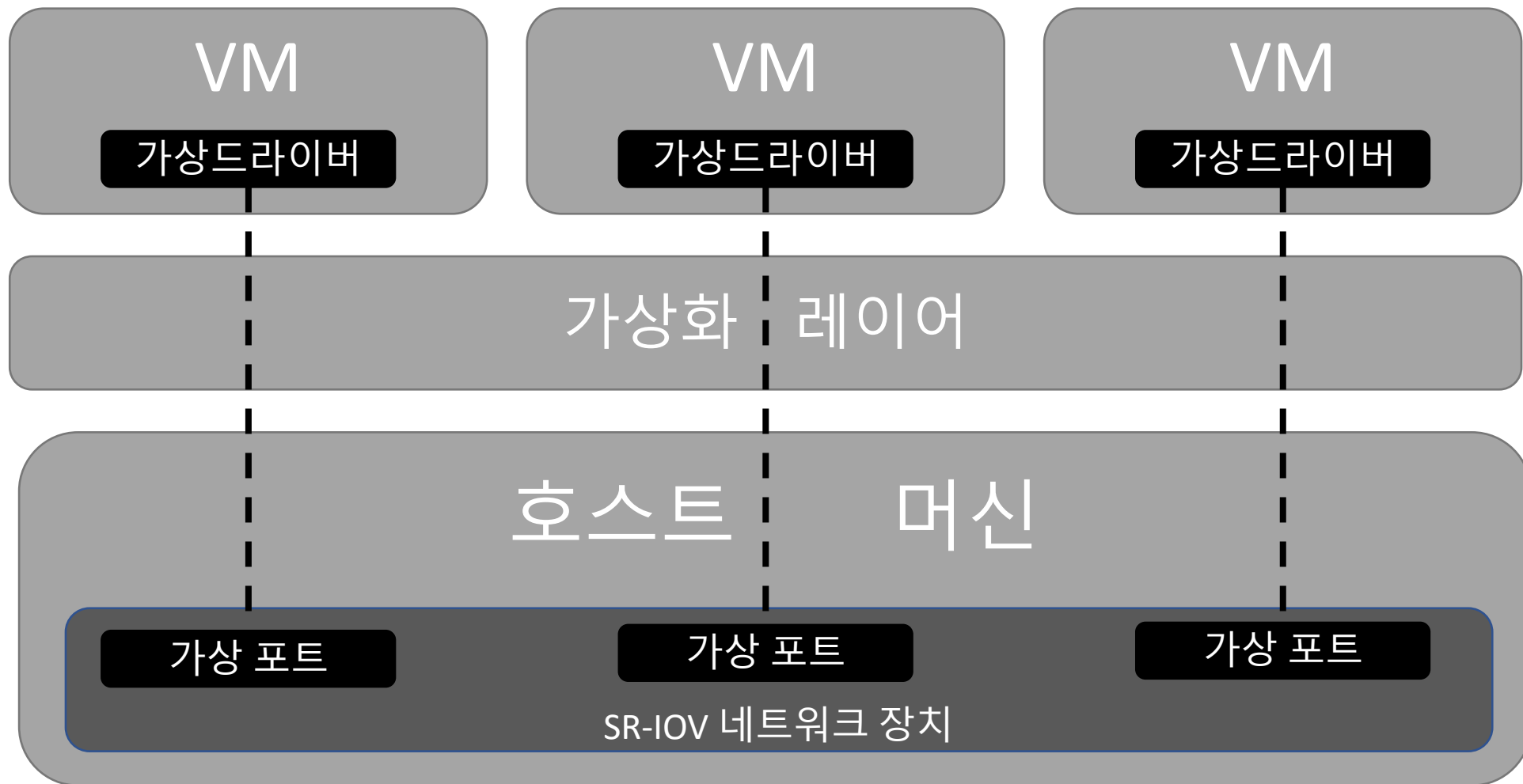
늘어나는 데이터 양의 처리 = 데이터 플레인 : 하드웨어에서 첫번째 가능한 부분

- 클라우드 서비스에서 사용자 데이터 양의 증가
- SDN의 구조를 다시 생각하면 제어 / 데이터 평면 구조에서 데이터 평면을 기억
- 데이터를 랜 카드에서 처리하는 양이 증가할 경우의 랜 카드 속도와 처리
- 예를 들어 그래픽 칩셋에서 처리하는 GPU
- 랜 카드에서 처리하는 스마트 NIC
- 데이터는 갈 수록 늘어나고 이 데이터가 이동할 때 패킷 단위로 이동
- 패킷을 분석 처리하고 빠르게 원하는 곳으로 이동하고 전달
- 내가 원하는 데이터가 빠르게 전달되는 것
- 전달되어 처리 => 데이터 센터 와 클라우드 센터
- 중간 단계에서 처리하여 최종은 정제된 데이터를 이용한 데이터 처리 후 사용자에게 다시 전달하는 과정 : 1초? 이내?

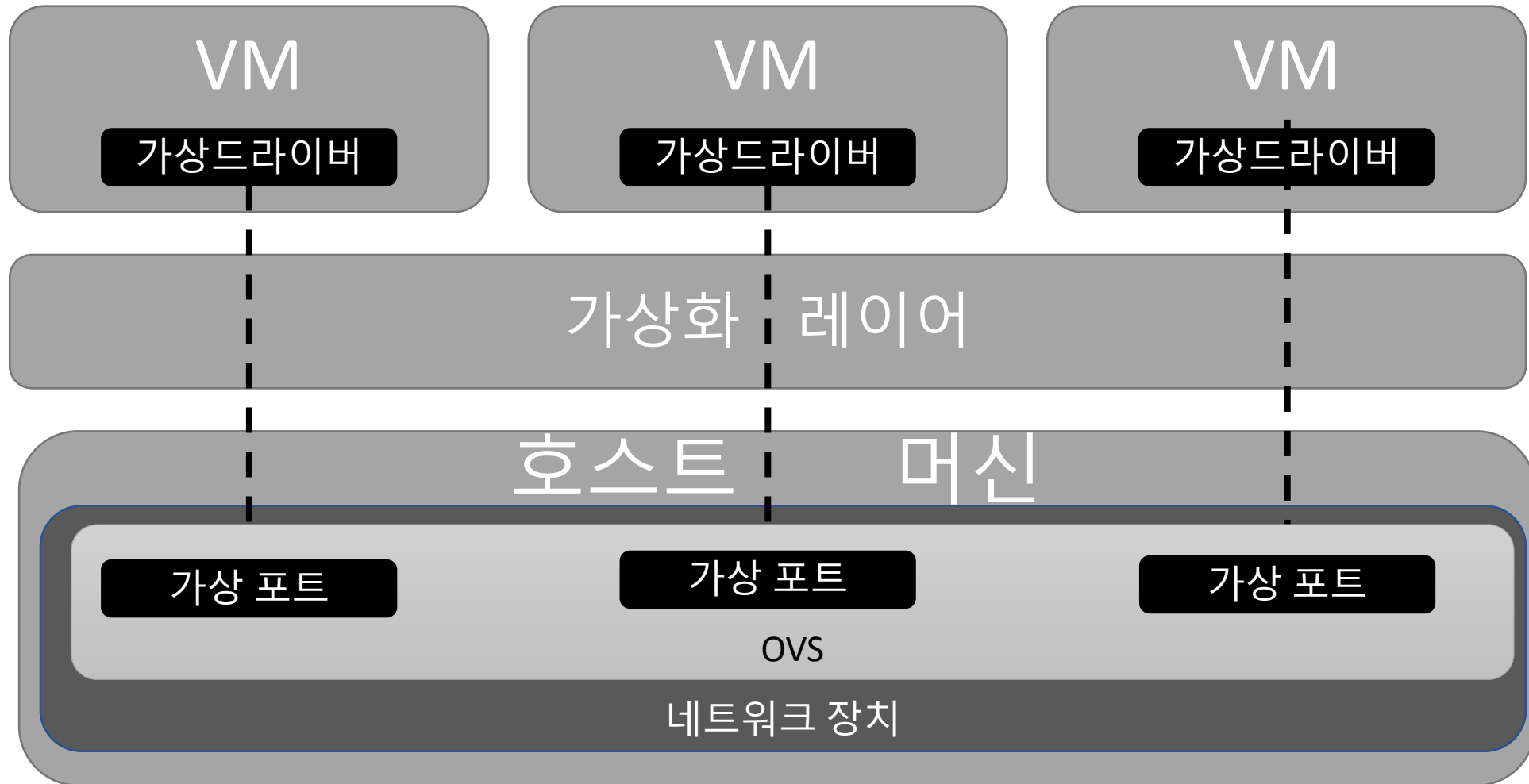
빠른 데이터 처리를 위한 플레인-네트워크 가상화 가속 기술사용



빠른 데이터 처리를 위한 플레인-네트워크 가상화 가속 기술사용



빠른 데이터 처리를 위한 플레인-네트워크 가상화 가속 기술사용

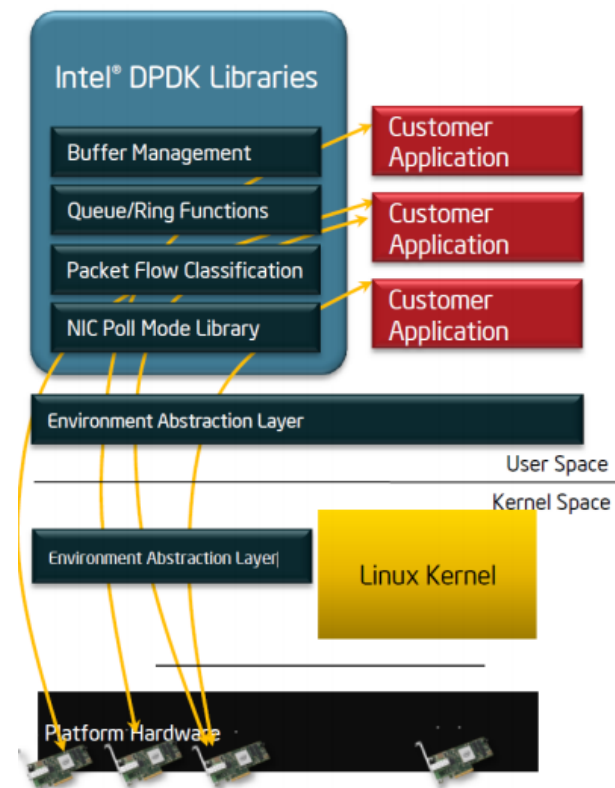


발전방향

Cloud Native의 기술

Intel DPDK

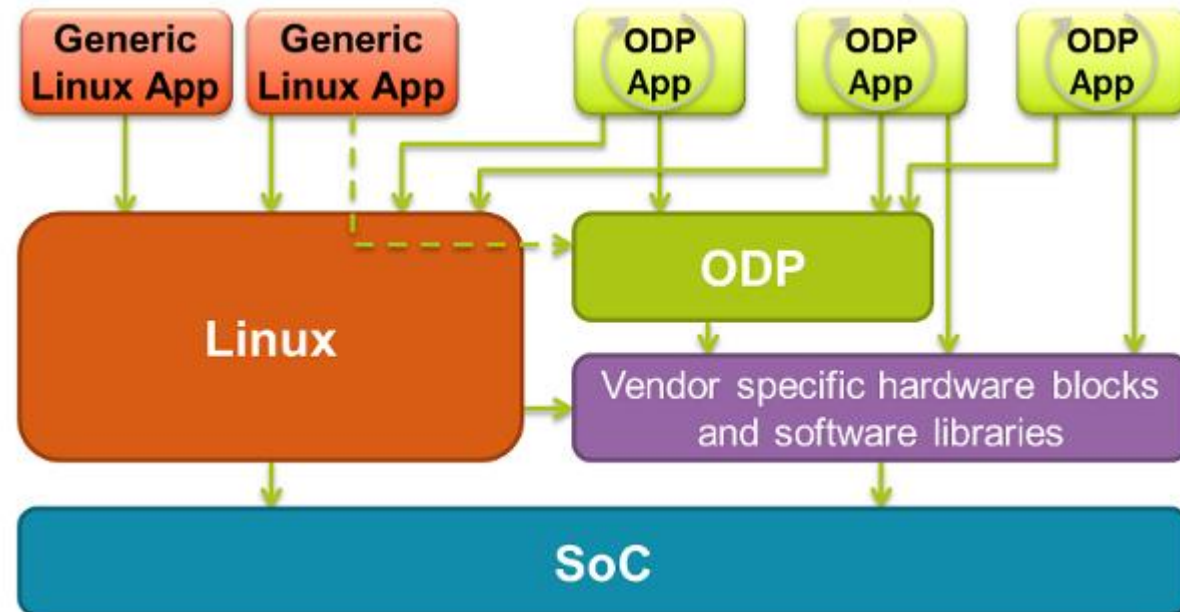
- Data Plane Development Kit
- 사용자 부분에서 커널을 통과하여 NIC에 직접 액세스 할 수 있도록 해주는 개발 도구
- 인텔 DPDK Lib API + EAL(Environment Abstraction Layer)
- 인텔 랜카드를 사용하는 x86 서버로 구현
- 인텔의 특정 랜카드만 구현 가능
- <https://www.intel.co.kr/content/www/kr/ko/communications/data-plane-development-kit.html>



Linaro ODP

ODP(OpenDataPlane)

- Linaro(리나로) = ARM 칩셋 및 모바일
- 모바일 환경등에서 높은 성능의 네트워크 데이터 플레인 응용프로그램을 위한 응용프로그램 프로그래밍 인터페이스를 정의하는 오픈소스 프로젝트



수고하셨습니다.

문의 사항 : sirmd@outlook.com