ACCLOUD: FPGA ile Hızlandırılmış Yeni bir Bulut Mimarisi
ACCLOUD (ACcelerated CLOUD): A Novel FPGA-Accelerated Cloud Archictecture

Alper Yazar*, Ahmet Erol†
Savunma Sistem Teknolojileri Sektör Başkanlığı
Aselsan A.Ş., Ankara
Email: *ayazar@aselsan.com.tr, †ahmeterol@aselsan.com.tr

Ece Güran Schmidt
Elektrik Elektronik Mühendisliği
Orta Doğu Teknik Üniversitesi, Ankara
Email: eguran@metu.edu.tr

***Özetçe* —Bu bildiride bulut tabanlı veri merkezleri için ACCLOUD (Hızlandırılmış Bulut) isimli yenilikçi bir mimari önerilmektedir. ACCLOUD veri düzleminde, bulut sunucularına takılı veya tek başına çalışan FPGA kartları donanım hızlandırıcı olarak çalışmaktadır. Buna uygun olarak FPGA yeniden yapılandırılabilir bölgeleri sanallaştırılmış olarak kullanıcıya işlemci, hafıza, disk gibi mevcut bulut kaynaklarına ek olarak sunulmaktadır. Bulut kontrolü OpenStack işçerçevesinin donanım kaynaklarını dahil edecek şekilde uyarlanmasıyla yapılmaktadır. ACCLOUD kaynak yönetimi, kullanıcı isteklerini işlemci ve donanım hızlandırıcılar arasındaki ödünleşimi gözeterek karşılayan yenilikçi bir yaklaşımla sağlanmaktadır.**

***Anahtar Kelimeler*—Bulut bilişim, Veri merkezi, Donanım hızlandırıcı, FPGA, Kaynak yönetimi.**

***Abstract*—In this paper, we propose ACCLOUD (ACcelerated CLOUD) as a a novel architecture for cloud data centers. ACCLOUD features FPGA cards which work with the cloud servers or stand alone for hardware acceleration in the data plane. To this end, FPGA reconfigurable regions are virtualized and offered to the user together with other cloud resources including CPU, memory or disk. The cloud control is carried out within OpenStack framework incorporating the hardware resources. We propose a novel resource management approach for ACCOUD which fulfills the user requests by considering the tradeoffs between assignment of soft computing resources and hardware accelerators.**

***Keywords*—Cloud computing, Data center, Hardware accelerator, FPGA, Resource management.**

I. Giriş

Bulut bilişim sistemleri büyük ölçekli dağıtılmış hesaplama sistemleridir. Yapılan araştırmaya göre şirketlerin %80'i bir biçimde bulut bilişimden faydalanırken %33'ü yoğun olarak bulut bilişime dayalı çalışmaktadır [1]. Bulut bilişim sistemleri ölçek ekonomileri (economies of scale) fikrine uygun olarak soyutlanmış (abstracted), sanallaştırılmış (virtualized), dinamik olarak ölçeklenebilen, yönetilen bir kaynak ve servis havuzunun kullanıcılara isteklerine göre internet üzerinden dağıtılmasını sağlarlar. Burada dağıtılan kaynaklar, hesaplama gücü (işlemci-CPU), hafıza (memory), depolama (disk) ya da ağ haberleşme kapasitesi (network bandwidth) gibi fiziksel makine kaynaklarıdır.

Bu bildiride bulut tabanlı veri merkezlerinde kullanıcılara sunulan işlemci ve hafıza gibi mevcut kaynaklara yeniden yapılandırılabilir donanım kaynaklarını ekleyen ve bu kaynaklarla birlikte IaaS, PaaS ve SaaS hizmetlerini birlikte verimli şekilde yöneten yüksek başarımlı yeni bir bulut tabanlı mimari olan ACCLOUD mimarisinin tasarımı ve planlanan ilk gerçekleştirim çalışmaları sunulmaktadır.

II. Bulut Bilişim ve Önceki Çalışmalar

Bulut bilişim sistemlerinin sağladığı başlıca servisler; altyapı (Infrastructure as a Service-IaaS), Platform (Platform as a Service- PaaS) ve Yazılım (Software as a Service-SaaS) dır [2]. Başlıca Bulut Bilişim Servis sağlayıcıları Microsoft Azure [3], GoogleCloud [4], Amazon Web Services (AWS) [5] olarak sayılabilir. IaaS; CPU, hafıza, disk ve ağ gibi kaynakların isteklere göre tahsis edilmesidir. Bu kaynakları kullanıcılar istedikleri işletim sistemini ve yazılım paketlerini kurarak kullanabilir. PaaS kullanıcılara kendi yazdıkları kodları koşturmaları için sanal makineler (Virtual Machine-VM) ve işletim sistemleri sunmaktadır. SaaS kullanıcılara kurulumu ve bütün verisi bulut üzerinde olan uygulamaları kullanma imkânı sağlar. Bulut tabanlı veri merkezleri fiziksel makine kaynaklarını sanallaştırma yolu ile dinamik olarak kullanıcılara sunmaktadırlar.

Bulut kaynaklarının soyutlanıp, sanallaştırılarak kullanıcılara dağıtılması için bir işletim sistemi gibi davranan *bulut bilişim kontrol platformları* arasında en çok kullanılan açık kaynaklı platform Openstack'tir [6], [7]. OpenStack kontrol modüllerinden ve her bulut bileşeninde çalışan sürücülerden (agent) oluşur. Başlıca OpenStack modülleri; kaynak dağıtımını yapan Nova, VM ağ konfigürasyonunu yapan Neutron, VM imajlarını tutan Glance ve Swift ile VM kullanım istatistiklerini toplayan Ceilometer'dir. Mevcut OpenStack gerçekleştirimi donanım kaynaklarını desteklememektedir.

Bulut bilişim sistemlerinde kaynak atama yöntemleri; kaynak verimi ile kullanıcı isteklerinin gelme biçimi ve hizmet hızı arasındaki ilişkileri ve enerji verimliliğini gözetmektedirler. Kaynak Yönetimi NP-hard bir problemdir [8], [9].

978-1-5386-1501-0/18/$31.00

CPU ölçeklenmesinin yavaşlaması ve Moore Kanunu'nun geçerliliğinin azalması nedeni ile bulut bilişim tabanlı veri merkezi sunucularına donanım hızlandırıcılarının (hardware accelerators-HA) eklenmesi sadece akademik değil uygulama alanında da çok yeni ve kabul gören bir konudur. Bu yeni donanım kaynaklarının bulut bilişim paradigmasına uygun biçimde mevcut CPU, hafıza, disk, bant genişliği (Bandwidth-BW) kaynakları ile birlikte kullanıcılara atanması gerekmektedir. [9], Microsoft veri merkezlerinde kullanmak üzere yeniden yapılandırılabilir FPGA ile hızlandırılan bir bulut tabanlı veri merkezi mimarisi önermektedir. Çalışmada FPGA kısmi yeniden yapılandırma (partial reconfiguration), kaynak yönetimi, kullanıcılara FPGA kaynaklarının sunulması ve kullanıcı ihtiyaçlarına göre ayrılması sunulmamaktadır. Ağ kaynaklarının sanal makineler ve FPGA arasında paylaştırılması incelenmemiştir. Kaynak yönetimi yapıldığı fakat çalışmanın kapsamında olmadığı belirtilmiştir. [10] 'da FPGA üzerinde birden fazla kısmi yeniden yapılandırılabilir bölge (Reconfigurable region-YYB) bulunmaktadır. FPGA YYBler OpenStack kullanarak kullanıcıya IaaS/HaaS (Hardware as a Service) olarak atanmaktadır. Bu çalışmada hesaplama düğümünde çalışan uygulama için gerektiğinde başka bir düğüm üzerindeki FPGA kaynaklarının hızlandırıcı olarak kullanılması ya da doğrudan birden fazla FPGA gerektiren bir uygulamanın çalıştırılması incelenmemiştir. Her iki çalışmada da FAC (FPGA Accelerator Card-FAC) düğümünün sunucudan bağımsız çalışması incelenmemiştir.

III. ACCLOUD Mimarisi

Önerdiğimiz ACCLOUD (ACcelerated CLOUD) mimarisi Veri Düzlemi, Kontrol Düzlemi ve Yönetim Düzlemi olarak yapılandırılmıştır. Mimari blokları Yönetim düzlemi detaylandırılmış olarak Şekil 2'de gösterilmektedir.

Veri Düzlemi: FPGA hızlandırıcı (FAC) düğümlerinden ve standart bulut bilişim sunucusuna FAC eklenerek elde edilen ACCSERVER hesaplama düğümlerinden oluşur (Şekil 1). ACCSERVER'ların ağ arayüzü FAC üzerinde gerçeklenerek filtreleme gibi ağ paketlerinin işlenmesi sunucu yükünü artırmadan gerçekleştirilir. FAC yapısı Şekil 3'te görülmektedir. FPGA platformu olarak Xilinx Zynq ürün ağacından bir SoC seçilmiştir. Bu SoC'de FPGA alanına ek olarak (PL = Programmable Logic), iki ARM Cortex-A9 çekirdekli bir işlemci (PS = Processing System) bulunmaktadır. Bu yeterince güçlü ARM işlemci üzerinde gömülü Linux işletim sistemi ile beraber OpenStack nova compute yazılımı çalıştırılarak FAC bulut ağı üzerinde bir bilgisayara takılı olmadan da bağımsız olarak çalışabilmektedir. Bu şekilde önceki çalışmalarda kullanılan soft işlemcilere göre daha yüksek başarım elde edilmesi amaçlanmaktadır. FAC donanım kaynakları; hizmet olarak altyapı (IaaS) kapsamında kullanıcılara kendi tasarladıkları donanımları yükleyip çalıştırabilecekleri FPGA alanları, PaaS kapsamında derlenmiş ve çalıştırılmaya hazır donanım hızlandırıcılar (Hardware Accelerator-HA) olarak sunulabilir. YYB'lere atanan HA'ların BW kontrolü FPGA üzerinde gerçeklenen giriş anahtarı ve dahili anahtar üzerinde token bucket benzeri bir çizelgeleme ile yapılır. 40 Gbps TOR isimli arayüz Ethernet anahtarı ile olan arayüzü, 40 Gbps HOST isimli arayüz FAC eğer bir sunucuya PCI-e üzerinden takılıyorsa o sunucunun Ethernet portuna bağlanacak arayüzü göstermektedir. FPGA içerisinde bloklar kendi aralarında ve ARM işlemci ile hızlı veri akışını ve ilgili blokların konfigürasyonunu sağlayan arayüzlerden haberleşmektedir.

Kontrol Düzlemi: OpenStack modülleri FPGA kaynaklarını yönetecek şekilde ACCLOUD mimarisine uyarlanmaktadır. OpenStack kontrol blokları Nova, Neutron, Cinder, Glance ve Swift Kontrolcu işlevleri aynı düğümde (Bulut Kontrolcü düğümü-BK) çalışmakta, ayrıca hem hesaplama düğümlerinde hem de FAC düğümlerinde bulunan işlemcide Nova Compute ve network (neutron) çalışmaktadır. FAC üzerinde çalışan modül diğer Hesaplama Düğümlerinde çalışan Openstack Nova-Compute modülüne benzer şekilde ve BK düğümü ile başarılı bir şekilde haberleşecek yetkinliğe sahiptir. BK düğümüne FPGA YYB kaynaklarının son durumlarını iletir ve BK düğümünden gelecek olan YYB başlatma isteklerini başarılı bir şekilde yerine getirir. Nova Compute hesaplama düğümlerindeki hypervisor ile libvirt arakatmanı üzerinden konuşmaktadır. IaaS-M Yönetici, PaaS-M ya da SaaS-M'den gelen IaaS isteği (VM (CPU, Hafıza, Disk, BW), FPGA YYB(YYB sayısı, tipi), FPGA Hızlandırıcı) BK'da çalışan Nova API'ya, takiben Nova Scheduler (Çizelgeleyici) modülüne gönderilir. Nova Scheduler Modülü Nova Resource Tracker ile güncel Kaynak durumunu tutan Nova Database (Veri tabanı) den kullanabileceği kaynakları öğrenir. Nova Scheduler ACCLOUD-MAN yöntemine uygun olarak kullanılacak kaynakları, FPGA YYB'lerini seçer ve veri tabanını günceller. Ardından ilgili düğüm (ler)deki Nova Compute daemon (artalan süreci-servis)'lara karar verilen kaynakların ayırılması için mesaj gönderir. Bu mesaj içeriği FPGA kaynaklarını da tanımlamaktadır. Nova Compute, Neutron Kontrolcu işlevi'nden başlatılacak sanal makine için IP ve MAC adreslerini, Glance imaj deposundan VM için istenen parametrelere göre imaj ister. FPGA YYB'lere yazılacak hızlandırıcılar için de Glance imaj veritabanı yapısı FPGA YYB'lere yazılacak hazır donanım hızlandırıcıları saklayabilecek şekilde güncellenmektedir. Sanal makine imajı Nova Compute tarafından alınıp libvirt soyutlama katmanı üzerinden bir hypervisor'a verilir. Ayrılan kaynakları Nova compute, BK'da çalışan Nova DB'ye iletir. IaaS-M'ye kaynakların ayrıldığını belirtir. FPGA YYBlerine yazılacak imajlar bulut servis sağlayıcısının SaaS kapsamında sunduğu kendi donanım hızlandırıcılarına ait olabildiği gibi kullanıcıya IaaS kapsamında YYB atandığında programlanmak istenen donanım olabilir. Bu ikinci durumda kullanıcı bitstream'i Glance'ye yükler ve oradan Nova yolu ile YYB'ye yazılır. Her iki durumda da bitstream imajlarının YYB sınır donanımına uyumlu olması gerekmektedir. FPGA YYB'lerine yazmak için hypervisor benzeri FPGA-visor adını verdiğimiz bir yazılım kullanılır. Bu yazılım, bulut hesaplama düğümlerinde hypervisora benzer şekilde libvirt üzerinden ya da TCP bağlantısı ile nova compute ile haberleşir. ACCLOUD mimarisinde FAC üzerindeki SoC ilk güç aldığında bağlı olduğu bir hafıza biriminden bir ilk imaj yükler. SoC üzerindeki ARM işlemci çalışmaya başlar (boot). Bu imaja göre YYBler etrafındaki statik donanım programlanır. SoC üzerindeki ARM işlemcide koşan FPGA-visor yazılımı ile bir GNU/Linux işletim sistemi koşacak ve Bulut Sunucu'da bulunan nova compute yazılımına benzer şekilde çalışır. Yukarıdaki iş akışına uyumlu olarak Nova compute glance'dan bitstream imajını alır. ARM üzerinde çalışan nova compute neutron'dan YYB için MAC adresi, glance'den de imaj aldıktan sonra PCAP üzerinden bu imajı YYB üzerine yazar.

Yapılacak olan gerçekleşimde FPGA YYB kaynakları diğer

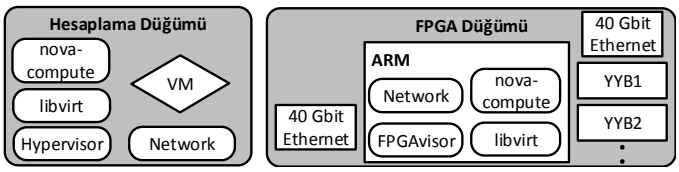
kaynakların (CPU, Hafıza, Disk gibi, BW) gösterimine uyumlu olarak OpenStack Compute modülüne eklenmesi ve veri tabanında da gerekli değişiklikler yapılması planlanmaktadır. Yapılan değişiklikleri doğrulamak için bir fiziksel makine üzerinde birden fazla sanal makine ve aralarında bir sanal ağ oluşturularak bir test ortamı kurulacaktır. Bu sanal makineler üzerinde BK ve hesaplama modülleri oluşturularak uyarlanan OpenStack yazılımının istenilen şekilde çalıştığı doğrulanacaktır.

Yönetim Düzlemi: ACCLOUD-MAN yaklaşımında IaaS-M, PaaS-M ve SaaS-M yazılımları yeni gelen isteklere dinamik kaynak ataması, kaynak kullanımın periyodik değerlendirilerek en verimli hale getirilmesi ve olay tetikli olarak (koşan bir işin kullanımının çok artması gibi) kaynak atamasının tekrar hesaplanması işlerini yapar. Şekil 2'de IaaS-M Bulut Kontrolcusu ile tümleşik gösterilmektedir. Yer kısıtı nedeni ile optimizasyon formülasyonu sadece SaaS için verilmektedir. IaaS-M, PaaS-M ve SaaS-M bileşenlerinin MATLAB/Tomlab yazılımı üzerinde gerçeklenmesi planlanmaktadır.

IaaS Yöneticisi (IaaS-M), VM isteklerini dış kullanıcıdan (kurumuna donanım alan sistem yöneticisi gibi), PaaS-M ya da SaaS-M'den alır. Aldığı isteklere ve takip ettiği VM kaynak kullanımına bağlı olarak optimal kaynak atamasını yapar ve VM'leri başlatır. IaaS-M tarafından kaynak ataması (Resource Allocation, RA) üç sebeple yapılır: 1) Dışarıdan (kullanıcı, PaaS-M, SaaS-M) gelen VM istekleri 2) Verim artırılabilmesi için periyodik RA yeniden hesaplanması sonucu (VM göçü, işlevi biten VM'lerin kapatılması) 3) Bir olay tetiklemesi ile RA'nın yeniden hesaplaması. Örnek olay olarak bazı VM'lerin fazla yüklenmesi sonucu koşan işlerin SLA (Service Level Agreement) / SLO (Service Level Objective) kısıtlarının sağlanmaması verilebilir. Buna göre kaynak ataması (RA) için CPLEX kullanılarak genel bir MIP (Mixed Integer Linear Programming) problemi oluşturulur. Amaç fonksiyonu (objective function): En az sayıda PM (fiziksel makine) kullanarak, işlerin ihtiyacını (SLA/SLO gereklerini) tam karşılayacak şekilde kaynak atamak, en az miktrada VM göçü yapmaktır. Başarım (Performance) tahmin edilen kaynak ihtiyacı ile atanan kaynak miktarının farkı olarak tanımlanmaktadır. Göç eden VMler için ceza (penalty) uygulanır.

PaaS Yöneticisi (PaaS-M), IaaS-M ile kaynak atamasını koordineli yapar. PaaS-M tarafından kaynak ataması IaaS-M ile aynı üç sebeple yapılır. Buna göre IaaS-M'den halihazırda kullanılan kaynakları ve kullanılabileceği kaynakları öğrenir. Buna ek olarak IaaS-M'den kaynak kullanım tahmini (prediction) alır. Bu bilgilere göre PaaS-M, IaaS-M'ye kaynak isteklerini yapar ve işleri mevcut VM'lere ya da yeni VMlerin başlatılmasını isteyerek bu yeni VMlere bir optimizasyon formülasyonuna göre atar. Optimizasyon problemi sadece yeni işleri, yeni işleri ve mevcut koşan işlerin bir alt kümesini ya da yeni işleri ve mevcut koşan işlerin tamamını VM'lere yerleştirecek şekilde ayarlanabilir. Eğer mevcut işler optimizasyona dahilse PaaS Monitörü halihazırdaki ve tahmin edilen değerleri PaaS-M'ye sağlar. Yeni işler için kullanıcı açık olarak PaaS'den kaynakları talep eder ya da PaaS çalıştırılacak olan işin özelliklerine (kod büyüklüğü gibi) bağlı olarak tahmini kaynak ihtiyacını belirler.

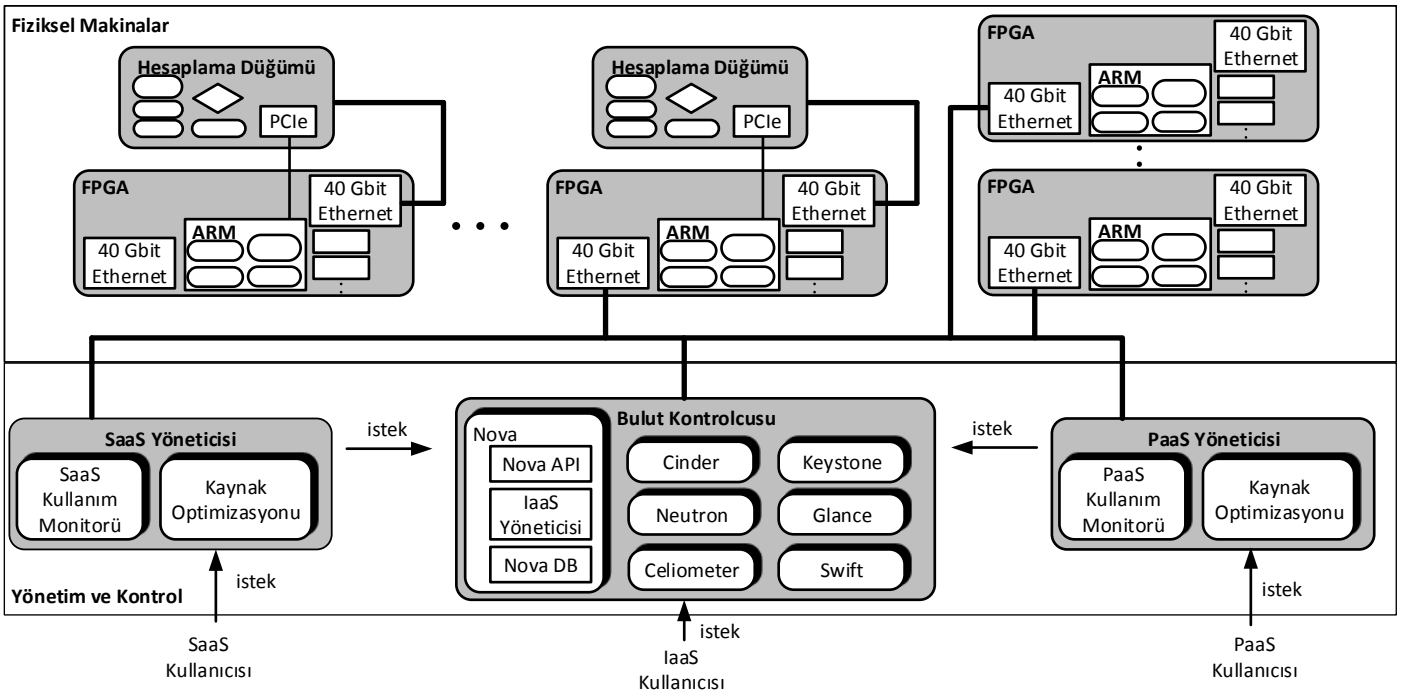
SaaS Yöneticisi (SaaS-M), Şekil 2'ye uygun olarak bulutta koşan yazılım servisine veri gönderip sonuç alan kullanıcıdan istekleri alır. Örnek olarak: GoogleCloud kullanıcısı SaaS olarak sunulan resim tanıma uygulamasına bir resim göndererek tanım alma işi verilebilir. İstekler için kaynak gereksinimi veri büyüklüğü ile tanımlanır. SaaS-M işlerin kaynak kullanma tahminlerini hesaplama düğümlerinde çalışan ve işlerin kaynak kullanımını takip eden SaaS Monitör'den alır. Kaynak cinsi $k \in \{m, f, c, b, d\}$ ile, kaynaklar, hafıza m, FPGA f, CPU c, BW b ve disk d ile gösterilir. Buna göre SaaS-M makine m üzerindeki kullanılabilir (available) kaynakları m_k^A, IaaS-M'den alır. VM ve Uygulama Kümeleri $\mathcal{M}$ ve $\mathcal{A}$ ile, uygulama $a \in \mathcal{A}$'yı koşturan VM kümesi $\mathcal{M}^a$ ile gösterilsin. Yeni VM'lerin kümesi $\hat{\mathcal{M}}$, her $\hat{m} \in \hat{\mathcal{M}}^a$ için $\hat{m}_k^A$ önerilen kaynaklar olsun. $m \in \mathcal{M}$ için tahmin edilen (predicted) kullanım m_k^P olsun. Her uygulama a için iş ve yeni işler kümeleri $\mathcal{J}^a$, $\hat{\mathcal{J}}^a \subseteq \mathcal{J}^a$ olsun. SaaS-M işi koşturmak için FPGA hızlandırıcıları da içeren seçenekleri değerlendirir. İşler için gerekli kaynakları hesaplar, her uygulama için en uygun seçeneği belirler. Bu kapsamda hangi işlerin mevcut VMler/FPGA donanım hızlandırıcıları üzerinde koşacağı hangi VM/FPGA donanım hızlandırıcıların başlatılmak üzere IaaS-M'den isteneceğini hesaplar. Büyüklüğü s olan bir iş j için seçenekler o_j, her seçenek o için gerekler $k_o = f_{o,k}(s)$ ile gösterilir. Başarım fonksiyonu $p_o = f_{o,p}(s)$, o için koşma hızı ya da harcanan güç miktarıdır. Nihai amaç en az VM kullanımı ve göç sayısı ile en fazla başarım elde etmektir. Minimize edilecek amaç fonksiyonu $\min_z Y + \gamma_Y \hat{Y} + \gamma_P P + \gamma_G G$ olarak ifade edilirse $\gamma_{Y,P,G}$ ifadeleri seçilecek ağırlıkları, z tüm karar değişkenleri vektörünü gösterir. Bu durumda kullanılan mevcut VM sayısı $Y = \sum_{m \in M} y_m$, yeni VM sayısı $\hat{Y} = \sum_{\hat{m} \in \hat{M}} y_{\hat{m}}$, iş başarımı $P = \sum_{m \in M \cup \hat{M}} \sum_{j \in J} \sum_{o \in O_j} x_{j,m,o}$, iş göçü $G = \sum_{j \in J} \sum_{o \in O_j} \sum_{m \in M} t_{j,m,o}$ olur. Burada y_m mevcut VM $m \in M$ kullanılıyor, $y_{\hat{m}}$ yeni VM $\hat{m} \in \hat{M}$ kullanılıyor, $x_{j,m,o}$ iş j VM $m \in M \cup \hat{M}$ seçenek o ile atanır ise 1 diğer türlü 0 olmaktadır. Amaç fonksiyonunun çözümünde bu bildiride tamamına yer verilemeyen çeşitli kısıtlar bulunur. Örneğin SaaS tarafından kullanılan kaynakların, $k \in \{m, f, c, b, d\}$, IaaS tarafından sağlanan kaynakları aşmaması gerekir.



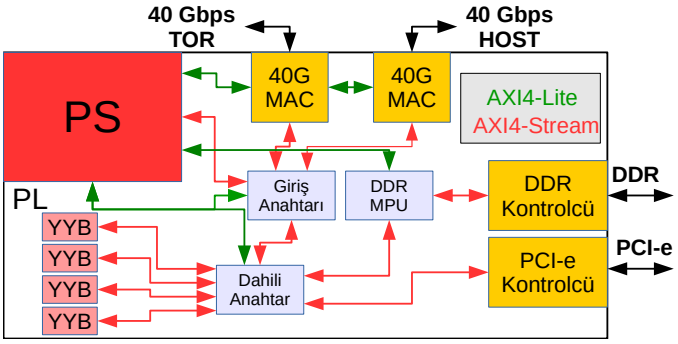
Şekil 1. ACCLOUD Veri Düzlemi bileşenleri

IV. SONUÇ

ACCLOUD Bulut tabanlı veri merkezleri için yeni bir donanım hızlandırıcısı mimarisidir. Bu mimaride, sunuculara takılı ya da üzerindeki ARM işlemci sayesinde tek başına çalışabilen FPGA kartlar (FAC) donanım hızlandırıcı olarak çalışmaktadır. Bu şekilde her düğümü sunucu olan bir bulut yapısına göre daha az güç tüketilmesi amaçlanmaktadır. FPGA üzerindeki kısmi yeniden yapılandırılabilir bölgeler (YYB) bilinen bulut kaynakları ile benzer şekilde kullanıcıya sunulmaktadır. Kaynakların ataması donanım kaynaklarını entegre etmek üzere uyarlanan açık kaynaklı OpenStack yazılımı ile yapılmaktadır. Donanım hızlandırıcıların kullanılmasının özellikle gerekli olduğu uygulamalar günümüzdeki 40 Gbps ve üstü hızlarda paket incelemesi ya da akan ağ



Şekil 2. ACCLOUD Yönetim Mimarisi



Şekil 3. FPGA Hızlandırıcı Kart

trafiğinin şifrelenmesi gibi ağ uygulamalarıdır. Bu nedenle ACCLOUD mimarisinde ağ trafiği doğrudan FPGA üzerinden geçirilmektedir. ACCLOUD SaaS Yönetici yazılımı kullanıcı verisinin büyüklük gibi özelliklerini değerlendirerek verinin sanal makinede ya da FPGA üzerindeki YYB'lere uyacak şekilde önceden tasarlanmış donanım hızlandırıcıdaki gerçekleştiriminde işlenmesine sistematik olarak karar verebilir. ACCLOUD kaynak yönetimi IaaS, PaaS ve SaaS yöneticileri arasında kaynakların soyutlanarak optimal biçimde sunulması ve kullanılmasını sağlayan yeni bir yönetim düzlemi olan ACCLOUD-MAN ile yapılmaktadır. ACCLOUD-MAN yaklaşımında IaaS, PaaS ve SaaS Yönetici yazılımları yeni gelen isteklere dinamik kaynak ataması, kaynak kullanımın periyodik değerlendirilerek en verimli hale getirilmesi ve olay tetikli olarak (koşan bir işin kullanımının çok artması gibi) kaynak atamasının tekrar hesaplanması işlerini yaparlar. ACCLOUD gerçekleştirimi sürmektedir. Laboratuvar ortamında sunucular, yüksek hızlı ağ anahtarı ve FPGA kartları ile gerçekleştirim tamamlandığında başarım testleri yapılacaktır.

TEŞEKKÜR

Yazarlar desteklerinden dolayı Savunma Sistem Teknolojileri Sektör Başkanlığı - ASELSAN A.Ş.'ye teşekkür ederler.

KAYNAKLAR

[1] "State of the Cloud Report," http://www.rightscale.com/lp/state-of-the-cloud.

[2] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *2009 Fifth International Joint Conference on INC, IMS and IDC*, Aug 2009, pp. 44–51.

[3] "Microsoft Azure What is Azure," https://azure.microsoft.com/en-us/overview/.

[4] "Run your application using the same technology and tools used at Google," https://cloud.google.com/products/.

[5] "Amazon Web Services," https://aws.amazon.com/.

[6] M. Mahjoub, A. Mdhaffar, R. B. Halima, and M. Jmaiel, "A comparative study of the current cloud computing technologies and offers," in *2011 First International Symposium on Network Cloud Computing and Applications*, Nov 2011, pp. 131–134.

[7] "Open Source Software For Creating Private and Public Clouds," https://www.openstack.org/.

[8] A. Yousafzai, A. Gani, R. M. Noor, M. Sookhak, H. Talebian, M. Shiraz, and M. K. Khan, "Cloud resource allocation schemes: review, taxonomy, and opportunities," *Knowledge and Information Systems*, vol. 50, no. 2, pp. 347–381, Feb 2017.

[9] A. M. Caulfield, E. S. Chung, A. Putnam, H. Angepat, J. Fowers, M. Haselman, S. Heil, M. Humphrey, P. Kaur, J. Y. Kim, D. Lo, T. Massengill, K. Ovtcharov, M. Papamichael, L. Woods, S. Lanka, D. Chiou, and D. Burger, "A cloud-scale acceleration architecture," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2016.

[10] S. Byma, J. G. Steffan, H. Bannazadeh, A. L. Garcia, and P. Chow, "Fpgas in the cloud: Booting virtualized hardware accelerators with openstack," in *2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines*, 2014, pp. 109–116.