**Metadata Considered Harmful…to Deduplication:**

**Yedeklenen veri içerisinde aynı olanların birleştirilmesi (tekilleştirme) depolama alanı etkinliğini artırmak için kullanılmaktadır. Ancak çok dikkatli bir şekilde yapılsa bile bu işlemin etkinliği depolanan veriye bağlıdır. Bu sunum da birçok dosya formatının tasarım problemleri nedeniyle tekilleştirmeye uygun olmadığı gösterilmiştir. Bu araştırmada gömülü üst veride (metadata) tekilleştirmenin iyileştirilmesi için 3 farklı model incelenmiştir. Bu sunumda üst verinin data ayrılması ile tekilleştirme işleminin 5,6 kata varan oranda iyileştiği öğrenilmiştir.**

**Accordion: Multi-Scale Recipes for Adaptive Detection of Duplication**

**Tarif (recipe) dosyaların içeriklerini bir dizi blok tanımlama değerleri (hash) ile gösteren bir üst veridir (metadata). Tarif verileri kullanarak iki dosya okunmadan içerikleri karşılaştırılabilir. Ancak küçük blok boyutu kopyalamayı maksimize ederken, büuük blok bıyutu küçük tarifler ile kolaylıkla karşılaştırılabilmektedir. Bu sunumda, Accordion ismi verilen bir program ile farklı boyutlarda blok kullanımı gösterilmiştir. Accordion programının 2 farklı tipte algoritma ile etkin bir şekilde çalıştığı ifade edilmiştir.**

**To ARC or Not to ARC**

**Ön bellek (cache) veri yolu (datapath) üzerindeki ön belleğin yönetimine odaklanmıştır. Veri yolu ön belleklerinde her bir ön bellek kaybı ön belleğin güncelleştirilmesine neden olmaktadır. Ancak ön bellek güncellemesi verini çıkarılıp tekrar ön belleğe alınması nedeniyle bilgisayar kaynağının kullanımı açısından maliyetli bir işlemdir. Veri yolundan olmayan ön bellekler ön bellek kayıplarında her defasında güncelleştirmeye ihtiyaç duymamaktadır.Bu araştırmada ARC (adaptivte replacement cache) algoritmasının geliştirilmiş hali olan mARC (multimodel adaptivte replacement cache) önbellek yer değiştirme algoritması anlatılmıştır. Bu algoritmanın ARC algoritmasına göre önbellek performansını arttırdığı, ön bellek güncelleme sayısını azalttığı öğrenilmiştir.**

**Terra Incognita: On the Practicality of User-Space File Systems**

**Web tabanlı uygulamalar veri çıktı miktarını artırmak ve kullanıcı gecikmesini engellemek maksadıyla “Memcached” gibi ön bellek sistemlerine bağımlıdırlar. Bu sistemlerde ki küçük iyileştirmeler büyük kazançlar sağlamaktadır. Mesela, verinin ön bellekte bulunma oranı %1 oranında arttığında, uygulamanın gecikme oranın %25 azalır. Ancak bir çok sistem jenerik ilk giren ilk çıkar (first in first serve) ve sabit boyutlu veri ataması dizayn yapısını sahiptir. Bu sunumda bu tür dizayn yapısına sahip Memcachier ön bellek servisinin performans kayıpları anlatılmıştır. Yapılan deney sonuçlarına göre bazı durumlarda jenerik ön bellek sistemlerinin 3 kat daha düşük performans ile çalıştıkları tespit edilmiştir. Bu çalışmada ön bellek sistemlerinde performans iyileştirmesi maksadıyla Dynacache olarak adlandırılan bir önbellek kontroller mekanizması anlatılmıştır. Yapılan deneyler neticesinde Dynacahe’in verinin ön bellekte bulunmama oranını %65 azalttığı öğrenilmiştir.**

**The Cloud is Not Enough: Saving IoT from the Cloud**

**The Internet of Things (IoT) bulut altyapısından faydalanan yeni sınıf uygulamalardır. Ancak, cihazları direkt olarak buluta bağlamanın bazı dezavantajları vardır. Bu çalışmada bahse konu dezavantajlar ile IoT için çözüm yöntemleri anlatılmıştır. Bu kapsamda, veri merkezli dizaynın IoT için daha iyi bir mimari olduğu ifade edilmiştir. Bu sunumda veri merkezli dizayn kullanılarak oluşturulan dağıtık mimariye sahip Global Data Plane (GDP) platformu anlatılmıştır ve GDP’nin bulut merkezli mimaride karşılaşılan problemleri çözüm yöntemleri belirtilmiştir.**

**Technical Sessions**

**Spartan: A Distributed Array Framework with Smart Tiling**

**Makine öğrenmesii, bilimsel araştırmalar ve biyoloji alanında çalışanlar MatLab, R ve NumPy gibi güçlü ve verimli dizin programlama dillerini kullanılar.Dağıtık dizin çatısı progrmları makinelere dağıtmayı amaçlamaktadır. Ancak dapıtılmış dizin yapılarına makinelerin lokal erişmesi problemi halen çözülememiştir. Bu sunumda Spartan ismi verilen dağıtık dizin çatısı anlatılmıştır. Spartan n boyutlu bir dizini maksimim yerel erişmeyi sağlayacak şekilde makinelere tahsis etmektedir. 12 farklı alandan seçilen uygulamalarda Spartan kullanılmış ve Spartan’ın otomatik atama mekanizmasının elle atamay gerek kalmadan iyi vee geliştirilebilir bir sonuç verdiği öğrenilmiştir.**

**Callisto-RTS: Fine-Grain Parallel Loops**

**Bu sunumda Oracle laboratuvarlarında geliştirilen ve Callisto-RTS olarak isimlendirilen ve çoklu soket paylaşımlı bellek mimarisine sahip makineler için geliştirilmiş paralel bir sistem anlatılmıştır. Callisto-RTS’in paralel döngülerin detaylı zaman planlamasını desteklediği ifade edilmiştir. Detaylı zaman planlamasının iş yükününün makinelere dağıtılmasını azalttığı ve yük dengesizliğini azalttığı öğrenilmiştir. Callisto-RTS öncelikle çekirdek başına düşen döngü müktarına göre ilk atamayı gerçekleştirmekte ve müteakiben asenkron bir şekilde iş taleb eden çekirdeklere iş paylaşımını yapmaktadır. 2 soket Intel 64 makine ve 8 soket SPARC makine üzerinde yapılan deneylerde Calisto-RTS’in iş yükü paylaşımı ayarlanması ihtiyacını azalttığı ve SPARC mimariye sahip makinelerde %39 oranında performans artışı sağladığı öğrenilmiştir.**

**Shielding Applications from an Untrusted Cloud with Haven**

**Bu çalışma İşletim Sistemleri Dizan ve Uygulaması 14 konferansında en iyi makale ödülünü almıştır.**

**Günümüzde bulut bilişim altyapısı güvene ihtiyaç duymaktadır. Bulut kullanıcıları hem servis sağlayıcının hem de global olarak dağıtılmış yazılım/donanımın özel verilerine ulaşmadığına güvenmektedir. Bu sunumda bulut ortamında çalışan programın ve verinin üzerinde çalıştığı platformdan (bulut operatörünün işletim sistemi, sanal makineler rvb.) koruyan zırhlı uygulama kavramı anlatılmıştır. Bu kapsamda SQL Serrver ve Apache gibi uygulamaların ilk prototip zırhlı uygulaması olan Heaven sistemi geliştirilmiştir. Heaven’ın software koruması ve fiziksel atakları önlemek için Intel SGX teknolojisini kullanmakla beraber binary dosyaların da kullanılan makineden zararlı yazılımları karşı korunmasını sağladığı öğrenilmiştir.**

**Surviving Peripheral Failures in Embedded Systems**

**Çevre birimleri zaman zaman çökmektedir. Ancak gömülü sistemler çevresel bilgisayar sistemlerinden kaynaklanan problemlerin tolere edilmesini kullanıcıya bırakmaktadır. Bu sunumda kaynak kısıtı bulunan gömülü sistemler için yarı otomatik kurtarma sistemi olan Phoenix anlatılmıştır. Phoenix çevresel donanım birimlerinin dahili ve harici durumlarını hafif kontrol noktaları ile control eden bir mekanizmadır. Bu systemin 6KB bellek kullanarak çevresel birimlerde çökme olmadı durumunda sistemi stabil duruma geri getirdiği ve kurtarma zamanı ve memory kullanımını azalttığı öğrenilmiştir.**

**In a World of Ephemeral Containers, How Do We Keep Track of Things?**

**Endustri oturumları kapsamında icra edilen bu sunumda Google firması tarafından birgisayar konteynerlerinin nasıl yönetildiği anlatılmıştır. Günümüzde ortaya çıkan pattern konteynerlardan oluşan bilgisayar kümeleridir. Bu sunumda konfigürasyon yönetimi, veri depolama ve diğer firmaların/bilişim insanlarının Google Bulut Platformudna bunları nasıl yönettiği tartışılmıştır. Bunlara ilave olarak bir açık kaynak küme yöneticisi ve zamanlayıcısı olan Kubernete’ler öğrenilmiştir. Kubernete’ler büyük miktarlardaki konteynerlerde uygulamaların yerleştirilmesi ve yönetilmeisini kolaylaştırmaktadır. Bu yöntemlerin PaaS (Platform as a Service) ve VMs (Sanal Makine) teknolojilerinde geniş bir uygulama alanı olduğu öğrenilmiştir.**

**Mambo: Running Analytics on Enterprise Storage**

**Büyük very (big data) specific bir yapısı bulunmayan ve geleneksel database sistemlerinde işlenemeyen büyül very setleri olarak tanımlanmaktadır. İşletmeler bu data setlerini depolamak ve analiz etmek için Apache Hadoop gibi analitik araçlara yönlenmektedir. Apache Hadoop yazılımı büyük miktarda ve çeşitlilikte very setinin porgramlama modelleri kullanarak bilgisayar kümelerinde dağıtık olarak işlenmesini sağlamaktadır. Hadoop iş yükünü, very depolamasını ve very yönetimini entegre etmektedir. Hadoop genellikle dışarıdan gelen harici very için kullanılmaktadır ancak ağda mevcut bulunan depolama alanlarında da mevcut veri için kullanımına ihtiyaç duyulmaktadır. Bu maksatla, sunumda Hadoop için NFS konnektor analtılmıştır. NFS konnektor Hadoopun veriyi taşımadan yada ayrı bir depolama alanı yaratmasına gerek kalmadan işlediği ve bu kapsamda kullanılan mimari yapı öğrenilmiştir.**

**Realtime High-Speed Network Traffic Monitoring Using ntopng**

**Bu çalışma LISA 14 konferansında en iyi makale ödülü almıştır.**

**Ağ trafiğinin izlenmesi sunucu sayısı, protocol ve topoloji çeşitliliği nedeniyle gün geçtikçe daha zor hale gelmektedir. Sanallaştırılmış ortamlar ve bulut servisleri atanmış izleme donanımlarını yerine sanallaştırma temelli trafik izleme uygulamalarına yönelmiştir. Bu sunumda, açık kaynaklı yüksek hızlı ağlar için dizayn edilmiş ve “ntopng” olarak adlandırılan trafik izleme uygulamasının dizayn ve uygulaması anlatlmıştır. Ntopng’nin önemli özellikllerinin gerçek zamanlı analiz, uygulama protokollerini ve kullanıcı davranışlarını tanımlama olduğu belirtilmiştir.**

**GridGraph: Large-Scale Graph Processing on a Single Machine Using 2-Level Hierarchical Partitioning**

**Bu sunumda tek bir makinede büyük boyutlu garifiklerin işlendiği bir sistem olan GridGraph anlatılmıştır. GridGraph ilk aşamada ön işleme ile grafiği 1 boyutlu (1D) köşe parçalarına ve 2 boyutlu (2D) kenar bloklarına bölmektedir. 2. Seviye parçalama işleme devam edereken yapılmaktadır. Bu yöntem ile GridGrapph’ın hesaplama için gerekli very giriş çıkışını azalttığı öğrenilmiştir. Çalışma kapsamında yapılan deneylerde GridGraph’ın bellek kapasitesi ve disk band genişliği ile uyumlu çalıştığı ve GraphChi ve X-Stream gibi son teknoloji ürünü sistemlerden daha iyi bir performan ile çalıştığı ifade edilmiştir. Ayrıca GridGraph’ın performansının dağıtık sistemler ile yarışabildiği ve bulut ortamında maliyet etkin olduğu belirtilmiştir.**

**Mahimahi: Accurate Record-and-Replay for HTTP**

**Bu sunumda HTTP tabanlı uygulamaların ağ trafik kayıtlarının tutulması ve tekrar simüle edilmiş ağ ortamındna oynatılmasını sağlayan Mahimahi çatısı anlatılmaktadır. Öncelikle Mahimahi’nin web uygulamalarının çoklu-server yapısını dikkatli bir şekilde simüle ettiği (ABD’de bulunan en çok ziyaret edilen top 500 sitenin %98’ini) ve kendi ağ trafiğini izole etiiği, üçüncü olarak da uygulamasının ve genişletilmesin kolay olduğu belirtillmiştir. Çalışma kapsamında Mahimahinin (1) HTTP,SPDY ve QUIC performansı analiz edilmiş, (2) bu protokellerin neden optimum olmadığı araştırılmış, (3) bu problemleri ortadann kaldırmak için bulut tabanlı tarayıcı olan Cumulus geliştirilmiş ve diğer sistmelerin Mahimahi’yi nasıl kullandığı incelenmiştir.**

**Bolt: Faster Reconfiguration in Operating Systems**

**Dinamik kaynak boyutlandırması yoğun yük durumlarında ekstra kullanılabilecek kaynakların tespit edilmesini ve düşük yük zamanlarında bazı kaynakları kullanmayarak enerji tasarrufu sağlamaktadır. İşlemci sayısını değiştirmek düşük yük zamanlarında güç tasarrufu için oldukça önemlidir. Mevcut işlemci boyutlandırma sistemleri problemli işlemci çekirdeklerinin kaldırılması ve değiştirilmesi şeklinde yapılmaktadır. Bu sunumda, mevcut altyapıyı kullanarak bıyutlandırma gecikmesini azaltan Bolt mekanizması anlatılmıştır. Bolt ayrıcı bir defada birden çok çekirdeğin eklenmesini ve çıkartılmasını sağlamaktadır. Bolt sistemi x86 ve ARM mimarilerinde test edilmiş ve işlemcilerin offline duruma geçişte 20 kat daha hızlı geçiş sağladığı, işlemcilerin devreye alınmasınnda x86 mimarili sistemlerde 10 kat, ARM mimarili sistemlerde 20 kat daha hızlı olduğu öğrenilmiştir.**