;login:
OCTOBER 2013 VOL. 38, NO. 5
ELECTRONIC SUPPLEMENT

Conference Reports

ICAC '13: 10th International Conference on Autonomic Computing

San Jose, CA
June 26-28, 2013

Keynote Address I

Summarized by Daniela Loreti (daniela.loreti@unibo.it)

Perspectives on Virtualized Resource Management

Carl Waldspurger

Carl Waldspurger began by explaining that hypervisors introduced a powerful extra level of indirection between hardware and software, which makes virtualization a successful strategy, but also introduces problems of complexity, performance isolation between workloads sharing the same hardware, and of the semantic gap between software running on a virtual machine and the host layer representing the hardware managed by the hypervisor. The hypervisor and the guest operating system (GOS) indeed seem like black boxes to each other.

Waldspurger continued by illustrating some practical examples of past work to solve measurement problems. In a virtualized infrastructure, measurements are indeed fundamental if you want to get a control optimization of the system, but there are several reasons why they can go wrong. The CPU interrupts sent to a descheduled virtual machine, for example, accumulate until it is resumed, and then typically they are sent to the guest operating system with a higher frequency breaking its CPU assumptions. Some modeling problems were addressed as regards the prediction of allocation and reallocations of resources in a virtual infrastructure and how these decisions can influence higher levels. An example of modeling was given regarding cache management, illustrating the improvements brought by Qureshi and Patt's work to make the expensive Mattson algorithm practical. Waldspurger also presented a new work on spatial sampling, CloudPhysics I/O MRCs, which enables online construction of miss rate curves showing two orders of magnitude improvement in both time and space costs.

Typically the systems to be modeled are complex, because we must take into account CPU, memory, network and I/O to conduct a performance analysis, and different approaches can be followed: (1) analytical modeling, which simplifies the assumptions and can lead to loss of particular characteristics of the system, but is in general fast and convenient to use; (2) simulation is another possibility; (3) active experimentation on real systems, changing some parameters and sampling the effects on the state of the system—for example, cloning and migrating virtual machines to test their performance with fewer cores; this latter approach unfortunately has an issue related to the external dependencies of the running VM, which must be replayed for the cloned copy and involves protocol-specific proxies; and (4) the final method is passive observation, collecting in a database all the past situations encountered, on the basis of which we can predict future system behavior. This approach involves special techniques to deal with critical mass of data and effectively enable model-by-query for the lookup of similar scenarios.

Waldspurger then focused on the higher-level control mechanisms, such as time- or space- sharing techniques, migration between machines or cores, etc. He emphasized that obtaining flexibility requires mechanisms that will support several different policies and that policy and mechanism should be kept separate. As an example, he presented the co-scheduling technique, adopted by VMware, to give a virtual machine the illusion of a dedicated multiprocessor, and the ballooning strategy to solve the problem of dynamic memory adjustment between the VM memory and hypervisor cache.

Finally, Waldspurger focused on policies usually expressed at the resource level, specifying the available physical resources allocated for the user. He noted that this is probably not the best way to express policies from the customer's point of view. A better way is probably related to the application level, in terms of response time or transaction rates, for example. In the real world, formal definitions of QoS and SLAs are surprisingly rare, and utility functions are even harder to express. Instead of using resource-level policies, a more direct alternative can be providing the customer with an "unhappy button" to give a single-bit feedback about his or her satisfaction. The Empathic Systems Project at Northwestern University is an example of research on methods to incorporate end-user satisfaction in computer architecture implementation.

In future research, Waldspurger highlighted the necessity of more accurate measurement to solve distortions resulting from descheduling, give the GOS access to hardware counters, and provide distributed measurements. From the modeling point of view, a lot of work remains to be done on big-data techniques and multi-resource modeling. More effective mechanisms are needed, for example, to enable end-to-end QoS controls. Regarding policies, future research should be in the direction of finding more intuitive and user-friendly ways to express them, addressing multiple resources from the application level point of view, realizing empathic systems, and also finding an accurate market-based model for pricing. Waldspurger looks to Resource Management as a Service (RMaaS) as a future direction, with possible hybrid automation transparently involving human experts with crowdsourcing possibilities.

Michal Rabinovic asked Waldspurger to clarify what he meant by "policies expressed at the application level"; it seemed to him that

modern clouds already provide this kind of information. Waldspurger answered that he was not talking about global performance metrics, but about solving higher-level problems in terms of application metrics (e.g., "I need to achieve a certain SLA accessing my database" and so on). He clarified that a lot of work in this area has already been done, but surely a lot remains to do. Andrew Tanenbaum commented that many years ago, Butler Lampson had suggested something like the "unhappy button," but with a twist: by pushing the button the customer would receive more resources with some probability, and otherwise would be logged out of the system, thus reducing the load for others. Zichen Xu asked if some sort of control on host power consumption had already been done in terms of physically switching off the machine when it was not being used. Waldspurger answered that when he was working in collaboration with VMware, one of the things they focused on was a system called Distributed Power Management (DPM), letting customers power off hosts when they were underutilized and power them on when resources were needed. VMware received a lot of feedback from customers, but the final result was influenced by the fact that users were pretty scared to turn off hosts they might not be able to turn back on. People responsible for keeping systems up do not care much about power consumption, so the mechanism was poorly applied.

Jia Rao asked about parametric measurement of the VM related to resource management: how can you isolate the power consumption of a VM? Waldspurger was aware of some work that VMware had done on this topic, trying to understand which amount of power consumption could be attributed to a virtual CPU activity. The same work can be done for the memory consumption tracking the accesses to memory by every single VM.

Cloud Management

Summarized by Daniela Loreti (daniela.loreti@unibo.it)

Application Placement and Demand Distribution in a Global Elastic Cloud: A Unified Approach

Hangwei Qian, VMware, Inc.; Michael Rabinovich, Case Western Reserve University

Hangwei Qian started by noticing that in a globally distributed multi-tenant cloud platform, identifying the correct number of databases on which an application should be running and how the client requests for this application should be distributed among the replicas is crucial. He formulated the problem in terms of a multi-objective minimization problem, in minimizing the overall cost, the number of application replicas, and the number of placement changes is necessary. The idea is to first address the problem of deploying the application on all the datacenters (full deployment) and then remove the underutilized instances.

For the first problem he used a minimum-cost network model, in which the flow represents the requests for a specific application going from a cluster client (Yam nodes) to each possible datacenter (DCn). The cost of each edge of the graph is the aggregate distance between a cluster client accessing a specific application and a specific datacenter. The authors propose a method to reduce the number of nodes in the network and make the problem easier to solve: the idea is for each pair-node Yam to rank the datacenters from the nearest to the furthest, producing a permutation of datacenter preferences. Now the Yam nodes can be grouped if they share a prefix of the permutation. The length of the prefix is a parameter.

After the resolution of this, a merged min-flow model is necessary to remove the underutilized application instances. To do so Qian suggested stating a deletion threshold (DT). If the amount of flow (i.e., requests of an application on a specific datacenter) is over this threshold, it is considered normal, and otherwise a reorganization of the application allocation becomes necessary: all these underutilized instances are removed and the algorithm tries to distribute their flows on the residual capacities. Because this second operation can result in a large number of placement changes, the authors propose a hysteresis ratio (HR) placement that basically reduces the DT threshold of some factor HR.

The performance evaluation of this approach was driven through a simulation on CSIM. The authors considered 20 datacenters, each serving 10,000 req/sec. By reducing the length of the prefix, they observed an increase in response time, but also a dramatic reduction in the algorithm execution time. Consequently, the prefix length was set to 3 for subsequent experiments. They show that the execution time is linear with the number of clusters, but superlinear with the number of datacenters. Finally, varying the demand change ratio, they compared their approach with other heuristics showing better performances of the former in both response time and number of dropped requests.

Samuel Kounev asked whether they also dealt with dynamic changes in the workload requested. Qian answered that the approach needs to be recalculated every day or every week to find the best application allocation. Jia Rao asked why the evaluation was conducted stating that every datacenter can serve 10,000 req/sec. The author answered that this parameter was caused by simulator limitations, but it was enough to test the system performances. Jia Rao then asked whether they dealt with dependencies between different applications. Qian answered that it would possibly be a future extension of this work.

To Reserve or Not to Reserve: Optimal Online Multi-Instance Acquisition in IaaS Clouds

Wei Wang, Baochun Li, and Ben Liang, University of Toronto

Ben Liang presented an online algorithm to let cloud customers decide which cloud offer they should adopt. A user can usually decide to run instances of an application with a pay-per-use model or prepay to reserve some resources and obtain a significant usage discount. Because the workload of the applications is mostly not known in advance, deciding which option to select is usually a nontrivial problem for customers.

The work focuses on two optimal online algorithms to support the customer decision without any knowledge of future workload, only considering the normalized discount α after reservation. They formulate the issue in terms of a minimization problem: under the constraint of having more available resource instances than the demand, we must minimize the total cost, composed of on-demand utilization and reservation fee. The exact solution of this problem is computationally prohibitive, but the authors observe that it is a generalization of the Bahncard Problem, based on which the deterministic and randomized online algorithms can be used to find a suboptimal solution in a computationally feasible way. The authors also took the definition of break-even point (β) from the Bahncard Problem to address the cost at which it is indifferent to the user either to work with on-demand or reserved instances. At every time the online deterministic algorithm looks back at the past reservation period. If the on-demand cost is greater than β, a new reservation is necessary and the history must be corrected such that the mistake will not afflict the following rounds. That this approach always finds a solution with cost less than (2-α) times the cost of the optimal solution can be proved, and is optimal for any online deterministic algorithm. The cost can be reduced even more using a randomized online algorithm, which strikes a good balance between reserving conservatively and aggressively. The idea of this algorithm is to compute the deterministic algorithm with a value z in [0, β] instead of β itself. Indeed, if we chose z=0 the algorithm always reserves instances; on the other hand, if z=β many fewer reservations are approved, The value of z is picked randomly using a defined density function f(z).

The Google cluster usage traces were used for the evaluation of the algorithms as if they were computing demand traces from a cloud infrastructure. They divided Google users into three groups by the demand fluctuation level and observed that for users with highly fluctuating demands the cost of the two proposed algorithms is similar to the cost of having all on-demand instances. As expected for users with low fluctuating demand, the best solution is reserving all the instances, while in the case of medium fluctuation the deterministic and the randomized approaches are the best choices.

Timothy Wood noticed that the purchasing option is usually a yearly contract, so there should not be a time constraint in the resolution of the problem, and even an optimal algorithm can be used even if it is computationally hard. Liang answered that the optimal algorithm is supposed to know the behavior of the future demand, which is not possible in real cases. When the number of instances is very large, obtaining an optimal offline solution can be computationally prohibitive. Jeff Kephard asked if they also dealt with the application of this approach from the cloud provider point of view. Liang answered that the initial study focuses indeed on the provider side, trying to understand which is the most profitable configuration of on-demand and reserved instances.

Elasticity in Cloud Computing: What It Is, and What It Is Not

Nikolas Roman Herbst, Samuel Kounev, and Ralf Reussner, Karlsruhe Institute of Technology

Nikolas Roman Herbst noted that, even though the term "elasticity" is pretty commonly used in literature, especially related to the cloud computing environment, there is no formal definition of this concept, and indeed it is sometimes mixed with the concept of system speed in reaction, scalability, and precision. Herbst defined the elasticity property of the system as "the degree to which a system is able to adapt to workload changes by provisioning and deprovisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible." The crucial meaning of elasticity is indeed the ability of the system to follow changes in the demand of resources, avoiding overprovisioning and underprovisioning to fulfill a power-aware behavior without violating the SLAs.

If we want to compare two systems with different efficiency rates, we must consider that using the same workload would result in an unfair contest because the more efficient system would seem to also have a better elastic behavior. But if we stress this efficient system with a higher workload, as to generate the same resource utilization of the other system, it can appear that it is no longer the most elastic. The authors summarized these concepts into a metric for elasticity to scale up, which is defined as inversely proportional both to the time to switch from an underprovisioned state to an optimal one and to the average amount of resources underprovisioned during the underprovisioned period. A similar definition, but related to overprovisioning, was given for elasticity to scale down.

K. Shankari asked whether the author considered the possibility that sometimes the violation of SLAs is more costly than simple overprovisioning. Herbst answered that they did not deal with this problem, but it should be only a matter of finding the best weights for elasticity to scale up and elasticity to scale down.

K-Scope: Online Performance Tracking for Dynamic Cloud Applications

Li Zhang, Xiaoqiao Meng, Shicong Meng, and Jian Tan, IBM T.J. Watson Research Center

Li Zhang focused on a possible application of queueing network modeling to dynamically adapt a distributed environment to changes in workload and amount of requested resources. The idea is to consider different classes of jobs, each one characterized by a specific arrival rate, an additional delay and an average response time. The system is supposed to be divided into a certain number of tiers, each one with an average utilization and a background utilization. The average service time of jobs from a specific class at a certain tier is another important variable of the model. Even if in the queueing network model the utilization of a tier and the response time of a job class are unknown variables and can be calculated from the other parameters, the authors pointed

out that in a real system they can be easily observed, while the other variables are pretty much unknown and can vary over time. Therefore the idea of a Kalman filter can be used to address this problem and estimate the values of the unobservable quantities. Since some equations turn out to be nonlinear, the Kalman model has been changed to address this situation.

The authors performed two kinds of experiments to evaluate the approach. They applied the Kalman filter to a simulated environment, considering the data collected from different real applications. They identified three types of requests with different arrival rates and, based on these traces, they calculated utilization, response time, and their relative error respect to the observed parameters. After that the authors tested the approach on a real system using SOABench to measure the performance and try to do some predictions. For example, they varied the number of cores allocated to a service and computed the observable parameter values with Kalman equations. The results showed that the prediction of utilization and response time changes follows the real behavior of the test environment.

At the end of the presentation Xiaoyun Zhu pointed out that in a real system we do not have all these parameters, so it seems we can only evaluate the approach from the bottom and then try to infer the utilization of the system from that. Zhang answered that a few years ago they applied this idea to infer the utilization with an offline algorithm using a set of workloads to find the right metrics, and then another set of workloads to evaluate the performance (recalling a machine learning technique). The same approach can be used for online computation, assuming that parameters change at a low rate, and so a small window of time can be used for computation.

Virtual Machine Management

Summarized by Hiep Nguyen (hiep_nguyen@ncsu.edu)

AGILE: Elastic Distributed Resource Scaling for Infrastructure-as-a-Service

Hiep Nguyen, Zhiming Shen, and Xiaohui Gu, North Carolina State University; Sethuraman Subbiah, NetApp Inc.; John Wilkes, Google Inc.

Nguyen stated that distributed resource scaling for infrastructure as a service (IaaS) clouds calls for a practical and efficient solution that must be lightweight, application-agnostic, and able to predict the overload in a near future of at least 1–2 minutes, which he called medium-term prediction, in order to provide enough time for creating a new VM. Current schemes such as reactive, trace-driven, and model-driven resource scaling schemes either cannot predict the overload early enough or need to make certain assumptions.

Nguyen introduced AGILE, an elastic resource scaling system for IaaS. He started by presenting pre-copy live cloning augmented by dynamic copy-rate configuration to achieve an immediate performance scale-up with little interference. To achieve medium-term prediction, AGILE uses a wavelet-based medium-term prediction algorithm. The basic idea is to decompose the original signal into multiple-scale signals, perform prediction at decomposed signals, and then synthesize the prediction of the original signal from the predictions of the decomposed signals. Finally, AGILE uses online resource pressure modeling to infer a proper level of the required resource in order to meet the SLO violation rate target. The evaluation showed that AGILE was able to achieve 3.42x better true positive and 0.34x the false positive than alternative schemes in terms of predicting the overload. AGILE was able to handle the overload effectively, reducing both SLO violation and resource cost.

Someone from VMware asked about the clarification of true positive rate and false positive rate. Nguyen said AGILE predicts the resource usage and uses the resource pressure model to evaluate the overload state, so these accuracy numbers reflect how well AGILE can predict the overload state. Timothy Wood (George Washington University) asked about scaling multi-tier applications: how does AGILE know which tier it should scale? Nguyen replied that the online resource pressure model is built for each tier, and AGILE will trigger scaling for the tier that will have a predicted overload. Ming Zhao (Florida International University) asked if there are multiple VMs in one tier, which VM will be cloned. Nguyen replied that AGILE will pick the least-loaded VM.

PACMan: Performance Aware Virtual Machine Consolidation

Alan Roytman, University of California, Los Angeles; Aman Kansal, Microsoft Research; Sriram Govindan, Microsoft Corporation; Jie Liu and Suman Nath, Microsoft Research

Alan Roytman started by describing the need for consolidating VMs in cloud infrastructures to improve efficiency; however, consolidating VMs might result in performance degradation due to resource contention. Current practice can only use half the number of cores and must leave the other half unused simply to avoid a performance issue. Alan described the VM placement as an NP-Complete problem and their tool (PACMan) as performance-aware, so that can minimize the resource cost while performance degradation is guaranteed to be within a specified bound.

Alan formally defined the problem of consolidating VMs on k-core servers. PACMan uses a VM profiling technique that can model the performance degradation when co-locating a given set of VMs. Alan then described the complexity of the VM placement problem, which is NP-Complete when k >= 3. In this case, PACMan proposes an approximate algorithm to solve the VM consolidation problem in polynomial time. Alan said that the proposed algorithm can achieve a resource cost within ln(k) of the resource cost of the optimal solution. PACMan has another mode named Eco-Mode where VMs are tightly packed and PACMan's goal is to minimize the performance degradation. A heuristic using VM swaps is used to handle this case. Alan finally presented the experiments. The results showed that PACMan was able to achieve

nearly optimal VM placement (within 10%), yielding 80% utilization with low performance degradation. In Eco-Mode, PACMan reduced the performance degradation by 52%.

Someone commented that the upper bound ln(k) might be good for the four-core servers; however, if k is large (e.g., 20), this upper bound does not show much improvement. Then he asked whether the author observed the upper bound when running the experiment and whether PACMan was tested with k larger than 4. Alan replied that they observed the value that is close to the upper bound and they haven't tested with k>4. John Wilkes (Google) commented that Google never leaves the cores unused; typically, there are hundreds of tasks running on a single physical host. Someone from VMware asked, what if we could co-locate more than k VMs on a single k-core server? Alan replied that it wouldn't affect the algorithm.

Working Set-Based Physical Memory Ballooning

Jui-Hao Chiang, Stony Brook University; Han-Lin Li and Tzi-cker Chiueh, Industrial Technology Research Institute (short paper)

Tzi-cker Chiueh described the motivation of minimizing the total physical memory consumption from a set of co-located VMs. He said that the VM needs to allocate an amount of memory just equal to the true working set (TWS). Thus, if we could estimate the working set of the VM accurately, the consolidation ratio (i.e., maximum number of co-located VMs) could be improved.

Chiueh then introduced a working set estimation technique. This is based on the observation that the number of "swapin" and "refault" counts will be close to zero if the physical memory allocation of VM is larger or equal to the working set. So in order to estimate the working set size, the physical memory allocation is gradually decreased and VM's swapin and refault counts are actively probed. At the instant that the VM's swapin and refault counts start becoming non-zero, the working set size is equal to the physical memory allocation. Based on this working set estimation technique, a memory ballooning technique is proposed which sets the balloon target equal to the memory working set. The evaluation shows that this technique could reduce up to 15.07% in balloon target while preserving a better performance.

Someone from VMware commented that the sampling-based approach is the general technique that can be applied to any type of guests, and he liked the idea of using swapin.

Coriolis: Scalable VM Clustering in Clouds

Daniel Campello, Carlos Crespo, Florida International University; Akshat Verma, IBM Research—India; Raju Rangaswami, Florida International University; and Praveen Jayachandran, IBM Research—India

Daniel Campello began by describing the benefit of analyzing the similarity between virtual machine images; however, it is not easy to make the similarity analysis scalable as the image sizes are big. Their system (Coriolis) is scalable in terms of classifying a VM's content similarity and semantic similarity.

Campello described a tree-based clustering algorithm that has running time of 0(n log n) where *n* is the number of images. The key idea is to reduce the total number of similarity operations by using hierarchical multi-level similarity to restrict the comparison to only similar clusters. Furthermore, Coriolis reuses the similarity computations with effective caching. Coriolis is compared against the k-medoids clustering algorithm, and the experiments show that it only took Coriolis 10 hours to cluster 99 images while it took a k-medoids scheme three days.

Someone asked about the benefits of computing image similarity. Campello replied that it depends on the type of application. For migration, it can choose to migrate all similar images at one to reduce cost. In case of troubleshooting, we can compare the failed server with a successful server of similar type.

MapReduce Workloads and Key-Value Stores

Summarized by Nikolas Herbst (herbst@kit.edu)

iShuffle: Improving Hadoop Performance with Shuffle-on-Write

Yanfei Guo, Jia Rao, and Xiaobo Zhou, University of Colorado, Colorado Springs
Awarded best paper!

Yanfei Guo introduced the popular MapReduce framework invented by Google in 2004 as well as the corresponding programming and execution model. A MapReduce execution consists of map, partition, combine, shuffle/sort and, finally, the reduce tasks. To motivate the presented work iShuffle and its contributions, Guo stated that due to a non-uniform key distribution the partition sizes vary significantly, leading to a disparity in reduce completion time. Reduce tasks are not started evenly due to inflexible scheduling and tight coupling of shuffle and reduce tasks leaving task-internal and -external parallelism unexploited.

iShuffle addresses these shortcomings by decoupling the shuffle phase from the reduce tasks by deterministically pushing map outputs proactively to slave-nodes. In addition, iShuffle balances the partition sizes using an online estimation mechanism and introduces a higher degree of flexibility for reduce task scheduling. For the partition size estimation, a linear model is calibrated to link input data size and partition size. The heuristic placement algorithm "largest partition first" is then applied, and the fair-share scheduling is disabled for reduce tasks to prevent waiting for unfinished maps.

Their approach was evaluated using the Purdue MapReduce Benchmark Suite (PUMA) on a 32-node Hadoop cluster. The results showed the iShuffle minimizes the shuffle delay significantly, leading to a job completion time reduced by up to 30%. The balanced partition placement results in up to 12% shorter execution times.

Vinay Deolalikar asked for an explanation of the term "data skew." Guo responded that "data skew" refers to key collisions in a hash table. K. Juan asked about the possibilities of custom partition

functions for iShuffle. Yanfei Guo replied that supporting this for more fine-grained estimations will be part of future work.

AUTOPLACER: Scalable Self-Tuning Data Placement in Distributed Key-Value Stores

João Paiva, Pedro Ruivo, Paolo Romano, and Luís Rodrigues, INESC-ID Lisboa, Instituto Superior Técnico, and Universidade Técnica de Lisboa

Best Paper Award Finalist

João Paiva began his presentation by stating that co-located processing with storage can improve performance. The commonly used random placement approach wastes resources for inter-node communication whereas an optimized data placement improves locality and reduces remote requests. A fine-grained placement would only be possible using costly offline optimization. So the main challenges are collecting usage statistics efficiently and deriving optimized placement information to exploit data locality while preserving a fast lookup for data items.

AUTOPLACER addresses these challenges by gathering only statistics on hotspot items and offering a fine-grained optimization in locality for hotspots in an online, round-robin manner. The hotspots are identified by applying the lightweight and memory-efficient Top-K stream analysis algorithm. The placement problem is relaxed from an Integer Linear Programming (ILP) to a Linear Programming (LP) problem. The result is then encoded and broadcast using a novel structure named Probabilistic Associative Arrays, and the data is moved accordingly. The results show a significant reduction of remote operations and an increased throughput.

Christopher Stewart asked about addressing more complex constraints in the ILP. The question was taken offline. Zichen Zu asked why the TCP-C benchmark had been selected for evaluation. Paiva answered that in fact this benchmark is less favorable to the approach due to being based on transactions, and other comparable benchmarks are synthetic and harder to support locality modification.

Adaptive Information Passing for Early State Pruning in MapReduce Data Processing Workflows

Seokyong Hong, Padmashree Ravindra, and Kemafor Anyanwu, North Carolina State University

Kemafor Anyanwu explained that information passing (IP) techniques are commonly used to optimize join queries in classical database systems. The passing of state information induces overheads whose costs may outweigh the benefits. To keep this intermediate state footprint small, the classical database techniques rely on statistics and indexing. MapReduce workflows contain a lot of state handling, but IP has not been investigated for parallel executions so far. MapReduce implies a restricted communication model in a shared-nothing environment. Introducing IP for MapReduce has an impact on design and implementation, but offers possible savings in execution time.

Kemafor Anyanwu proposed an Adaptive IP approach for Hadoop workflows that uses a benefit estimation model (BEM) which collects statistics by query-piggybacking. The approach consists of an execution plan analyzer, a cost estimator, and an information passing planner. The statistics efficiently collected using the HyperLogLog logarithmic counting method and are compactly represented by applying Bloom filters.

The experimental evaluation on an 21-node (1 master) Hadoop cluster running the TPC-H benchmark with a 40 GB data set shows that the proposed IP approach outperforms in terms of execution times for small reference ratios. The heap memory consumption is especially reduced compared to a Hive-based IP. In addition, it was shown that disabling the benefit estimation results in longer execution times due to the induced overheads as of the default Hive implementation.

Christopher Stewart asked about the applicability for high data volumes. Kemafor Anyanwu answered that the approach has been tested with a feasible data volume of 50 GB. Stewart then asked about the flexibility of the approach to change the amount of nodes during runtime. The answer was that it is unusual to modify the number of nodes for a loaded MapReduce cluster.

Management of Big Data Systems

Summarized by João Paiva (jgpaiva@gsd.inesc-id.pt)

To Auto Scale or Not to Auto Scale

Nathan D. Mickulicz, Priya Narasimhan, and Rajeev Gandhi, YinzCam, Inc., and Carnegie Mellon University

Nathan D. Mickulicz presented the architecture of YinzCam, a real-time mobile-based application sports service hosted on Amazon's cloud, which displays a multi-modal behavior related to different stages of the game and year. The work was motivated by the fact that the system must be able to decide when to scale up and down, with special focus on scaling down since it is easy to make this decision prematurely and it may hinder the user experience at particularly important times.

In order to tackle this problem, the authors applied the Amazon Auto Scaling mechanism to their architecture, and found that by using a low limit on CPU for scaling up and a slow scale down they were able to improve the user experience. However, this also resulted in a large waste of resources. After inspecting the base architecture again, the authors concluded that using Auto Scaling was hiding inefficiencies in the system, and after optimizing the architecture they were able to reduce the required resources by up to eight times.

Nathan was asked about the spin-up time of the application. He replied that even though one instance can in fact be created instantly, the average is around four minutes. Another member of the audience asked if the authors tried experimenting with different image sizes. Nathan replied that the implementation of their service does not allow it. Finally, Nathan was asked about the methodology to do

research on his large production system. He replied that they capture traces of days and then recreate the load offline.

Big Data Exploration via Automated Orchestration of Analytic Workflows

Alina Beygelzimer, Anton Riabov, Daby Sow, Deepak S. Turaga, and Octavian Udrea, IBM T. J. Watson Research Center

Deepak S. Turaga presented a system that automates the orchestration of analytical flows, which was deployed in a Big Data scenario. The system allows the end user to create, deploy, and manage analytic workflows with minimal effort, by defining only a high-level specification of her objectives.

The proposed system uses planning to identify adequate analytic workflows given high-level descriptions of composition patterns and individual analytical building blocks. Machine learning is used to explore the space of possible workflows and automatically combine the flows in response to dynamically changing data characteristics. The system was tested with a complex ECG workload.

Turaga was asked whether the authors tested problems with high dimensionality, and he responded that the monitoring example has a few hundred features. Someone asked about the relation between planning and loading, and whether the authors thought about how they could use a partial plan to guide the system. Turaga found the idea useful and will try it in the future. Someone queried about which flows the system is required to test. Turaga replied that they test all flows, since a flow always has to run somewhere. Another audience member asked about how complicated it would be to hand tune the ECG test. Turaga replied that a co-worker worked on that part of the system and hence he couldn't respond. Finally, someone asked what was specific to big data in this work. Turaga replied that the work provides the ability to run multiple analysis on the same data.

ThroughputScheduler: Learning to Schedule on Heterogeneous Hadoop Clusters

Shekhar Gupta, Christian Fritz, Bob Price, Roger Hoover, and Johan DeKleer, Palo Alto Research Center; Cees Witteveen, Delft University of Technology

Shekhar Gupta presented a system that reduces the time for completion of Hadoop jobs in clusters composed of heterogeneous nodes. The main idea behind the system is that it matches Hadoop tasks with the machines which better match the job's requirements. To achieve this, the system uses an online learner to profile the tasks of a job and decide where to run them.

The system was evaluated on a five-node cluster, and the authors showed that not only does it lead to considerable improvements in performance but has no negative impact on a homogeneous cluster.

Someone questioned Shekhar about how his work affects data locality in Hadoop. Shekhar replied that the system has a scheduling step where the jobs are first matched with nodes with good data locality, and only then optimized for heterogeneity. Shekhar was then queried about starvation, whether optimizing for best latency could or could not lead to a job not running due to lack of a suitable machine available. Shekhar replied that the system will always choose hosts available at the time, even if not perfectly suited for the task.

Real-Time User-Centric Management of Time-Intensive Analytics Using Convergence of Local Functions

Vinay Deolalikar, HP-Autonomy Research

Vinay Deolalikar presented a mechanism to obtain useful information in real time from time-intensive analytics. The key idea is that while an algorithm may take a long time to return a result, we can obtain meaningful information in the early stages of the algorithm, by analyzing its internal state.

To do this, Vinay proposed decomposing the objective function of a time-intensive analytic into multiple local, user-centric functions which can converge faster than the global function does. This approach was applied to the k-means algorithm, by visualizing it as an algorithm where documents flow from one bucket to another, and concluding that a bucket has stabilized when no documents flow out of it.

Vinay was asked how this approach takes advantage of the fact that the flows are from one bucket to another. Vinay replied that since the flows are local, we can allow a cluster to converge while the remainder of the algorithm hasn't yet converged. Another member of the audience asked how this approach is better than just reducing the convergence criteria. Vinay replied that he tried simply leaving k-means sooner; however, in fact the results are not quite as good as this approach. For some of the instances, sometime there are strong flows until right at the end of the algorithm, and if the algorithm finished earlier, a good percentage of the optimization would be lost.

AutoTune: Optimizing Execution Concurrency and Resource Usage in MapReduce Workflows

Zhuoyao Zhang, University of Pennsylvania; Ludmila Cherkasova, Hewlett-Packard Labs; Boon Thau Loo, University of Pennsylvania

Zhuoyao Zhang presented an automation framework for optimizing the configuration of the reduce task in Hadoop. This work is motivated by the fact that jobs behave differently according to the number of reduce tasks associated with them, and configuring such parameters requires user experience and expertise.

To tackle this problem, the authors proposed using an ensemble of performance models to predict the completion time of jobs, combined with optimization strategies which derive the best system configuration according to the performance models. The system was tested using TPC-H on a 66-node cluster. The results showed that the model was able to accurately predict the performance of the system, and that the system can indeed lead to a reduced resource usage.

There were no questions.

Self-Aware Internet of Things

Summarized by Sebastian Niemann (niemann@sra.uni-hannover.de)

Self-Healing and Optimizing of the HIP-Based M2M Overlay Network

Amine Dhraief, HANA Research Group, University of Manouba; Khalil Drira, LAAS-CNRS, University of Toulouse; Abdelfettah Belghith, HANA Research Group, University of Manouba; Tarek Bouali and Mohamed Amine Ghorbali, HANA Research Group, University of Manouba, and LAAS-CNRS, University of Toulouse

Mahdi Ben Alaya presented this paper, filling in for the original authors. The Machine-to-Machine (M2M) paradigm is based on the communication of several monitored devices over some sort of network with processing devices, which in turn may also be monitored devices. Despite the fact that this approach breaks with the well-known "keep it simple in the middle, smart at the edge" network paradigm and adds more complexity within the middle, a rapid increase of M2M systems, which may outnumber the number of humans in several years, can be observed. This motivated Amine Dhraief et al. to develop smarter M2M networks by integrating self-healing and self-optimizing properties.

The presentation focused on introducing self-healing and self-optimization within their previously developed HIP-based M2M overlay network called HBMON, which decouples end-host identification from its localization. The self-healing property of HBMON is mainly based on the failure recovery procedure of the reachability protocol REAP and its coupling with the host identify protocol (HIP), while the self-optimization property is based on autonomic path selection capabilities, which shall lead to selecting the best available overlay path in term of RTT.

To evaluate the approach, the authors used the OMNeT++ simulator coupled with the HIPSim++ framework in order to estimate the failure detection and recovery time, as well as the path exploration, by exposing the system to a transient failure after 20 ms. The given results demonstrated that the presented approach is able to detect these kinds of failures and recover from them.

Between Neighbors: Neighbor Discovery Analysis in EH-IoTs

Shruti Devasenapathy, Vijay S. Rao, R. Venkatesha Prasad, and Ignas Niemegeers, Delft University of Technology; Abdur Rahim, CreateNet

Abdur Rahim started by discussing an analytical model for the study of the neighbor detection performance in an energy-harvesting Internet of Things domain. After some clarifications about the computation, the results based on the analytical model were presented. The numeric results for the node discovery time over the node density displayed a logarithmic-like trend of neighbor discovery (ND) with increasing node density and an exponential-like trend over the bandwidth. Overall, the results implied a superiority of an omnidirectional approach over a directional one.

Based on the results of the analytical model, Abdur Rahim et al. simulated two setups for the omnidirectional and directional approach, which in contrast to the results of the analytical model suggested there's a tradeoff between energy harvesting and neighbor detection when choosing an omnidirectional model over a directional one. Furthermore, it could be observed that the neighborhood detection time did not continue to improve after a certain value of the supercapacitor capacitance was reached.

Several in attendance asked if the type of the antenna, whether omnidirectional or directional, could be adjusted at runtime in regards to the needed performance. Abdur Rahim replied that this was possible in general, but it might reduce the performance even more since additional energy consumers need to be added.

Towards a Generic Architecture and Methodology for Multi-Goal, Highly-Distributed and Dynamic Autonomic Systems

Sylvain Frey, EDF R&D and Télécom ParisTech, CNRS LTCI; Ada Diaconescu, Télécom ParisTech, CNRS LTCI; David Menga, EDF R&D; Isabelle Demeure, Télécom ParisTech, CNRS LTCI

Sylvain Frey presented work on the development of a generic architecture for self-configurative multi-agent systems with conflicting goals for overlapping system parts. The authors' general research questions where how to develop adaptable feedback loops, integrate feedback loops for different goals, and deal with conflicting goals and openness.

He motivated his talk by a concrete example of smart homes which are connected to a smart micro-grid. A smart home was described as a system that integrates different devices like context-aware heating, entertainment, lighting, and security through its own control logic, while a smart micro-grid is simply a local electrical network. With smart micro-grid managers imposing load management goals for the grid, which should limit the maximal power consumption of each smart home, and consumers imposing comfort goals for their households, which might violate the power consumption rule, a policy is needed to resolve these conflicting goals.

Frey then briefly discussed several parts of the presented generic architecture, which consists of application-specific abstract layers which may be enhanced by several control layers, and focused on the steps of their development methodology: specification of goals, identification of (relevant and context) resources, definition of control elements, and identification of conflicting zones. Their future work will focus on an even more rigorous analysis of self-configuring systems in order to identify and develop more reusable paradigms and models for devising future organic computing systems, as well as autonomic computing systems.

Learning Deployment Tradeoffs for Self-Optimization of Internet of Things Applications

Arun Kishore Ramakrishnan, Nayyab Zia Naqvi, Zubair Wadood Bhatti, Davy Preuveneers, and Yolande Berbers, KU Leuven

Arun Kishore Ramakrishnan presented his work on a methodology to learn deployment trade-offs for self-optimizing systems like motion activity monitoring or fall detection for elderly citizens within the context of the Internet of Things.

The presented approach was based on managing variability zones within a Pareto-front in order to overcome the drawbacks of classical approaches within the context of the Internet of Things. In the classical approach, each configuration corresponds to exactly one point within the Pareto-front, but this leads to high resource consumption and may even increase the inconsistency of the system due to the openness of the Internet of Things. In contrast, the presented approach represents each device through certain statistical properties, and groups components with similar properties in order to reduce the overall number of Pareto-front representations.

Someone asked what would happen if the relevant profiling information, which is collected during design time, changes during runtime so that the estimated Pareto-front needs to be adjusted. Arun Kishore Ramakrishnan argued that the given approach might be used iteratively during runtime in order to adjust to these changes.

Keynote Address III

Summarized by Zichen Xu (xuz@ece.osu.edu)

Opportunities for Autonomic Behavior in Mobile Cloud Computing

Dilma Da Silva, Qualcomm Research

Beginning with her early work in graph theory in object-oriented programming, Dilma Da Silva reviewed her past experience in the automatic computing community, detailing the pros and cons. She then discussed her work at IBM in a resource adaption study inside an operating system kernel, with a focus on the optimization of managing file objects, soft rebooting, and self-healing. Comparing this with the work in early '90s and current hot topics in MapReduce and cloud, Dilma suggested that hints from her early work could be brought back to address the challenges that we are facing today.

Dilma then addressed current hot topics in mobile cloud computing, especially when facing the challenge of the exponentially increased mobile traffic in global Internet traffic. She stated that smartphone adaptation is unprecedented, and is one of the fastest-growing domains of computing research. She compared the progress of research work in ICAC and Mobisys in terms of two topics: crowdsourcing support and resource management. Many challenging questions are raised by this, such as pre-fetching data management and synchronization optimization. Dilma discussed the difference between mobile workloads and server workloads with a usage breakdown to highlight those research questions, and she encouraged researchers to take advantages of such "low hanging fruit" to broaden research in this area.

Last, Dr. Silva discussed her opinions on how automatic computing will play in the field of cloud computing by invoking four steps: observe, learn, act, and refine. She shared many engineering works, such as the elastic load balancer for 150 high-CPU instances running Python Django, the success of Facebook and Pinterest, etc. When it comes to mobile cloud computing, the adaptations suggested by Dilma are offloading, tuning cloud capability, and adjustable communication. Offloading would mean augmenting mobile computing with the "mighty cloud capability." One easy fruit to pick is optimizing the cost model of those operations in mobile cloud computing. The keyword behind it is the mobile backend as a service (MBaaS). She closed her talk with a big framework of all important components in mobile cloud computing waiting for researchers to take the challenge.

Dr. Xiaohui (Helen) Gu asked about offloading in two extremes, using the cloud as mobile storage and using the mobile as one computation node. Dilma answered with state-of-the-practice ways to handle offloading with examples of streaming data. Also, she pointed out that there are many performance metrics that can be considered in the offloading phase, which could produce a great number of interesting results. Dr. Xiaoyun Zhu asked about optimization of the service in Pinterest with the performance bound by the Amazon EC2 service threshold. Zichen from OSU shared concerns of energy management in this field, and Shankari from VMware discussed the privacy and security concerns that may turn into the biggest conversation in the mobile cloud computing field.

Scheduling

Summarized by Zichen Xu (xuz@ece.osu.edu)

Zoolander: Efficiently Meeting Very Strict, Low-Latency SLOs

Christopher Stewart and Aniket Chakrabarti, Ohio State University; Rean Griffith, VMware

Chris Stewart began his talk with the landscape of the problem of meeting the strict SLO for Internet service which usually has a slow-response-time penalty. In this problem, revenue and hardware costs scale out with the growth of the arrival request rate. Specifically, according to the authors, one slow outlier of 100 concurrent requests may cause a 1% revenue drop. This created the interesting and challenging question that the authors tried to address: how to constrain the SLO within 99.9% of accesses completed within 15 ms. The key message behind this report is that although more resource assignments do not equal more throughput, more resource assignment to one access would increase the possibility of this access to meet a stronger SLO.

Chris presented Zoolander, a novel system design for strong SLO guarantees. In the implementation of Zoolander, he discussed the model extension in the case of timeout and resend scenarios. The major metric in evaluation and predicting the SLO is, at scale D (D is the number of copies), the SLO = 1 - (1-λ)D, where λ is the probability that one request missed the 15 ms SLO. Chris said that one big challenge has been addressed in their report—duplication in a shared DC network. With the same queueing delay, as in a general M/G/1 queueing model, Zoolander finds the best approach depending on the arrival rate.

Chris discussed their evaluation of Zoolander in a physical testbed. Compared to the setup in EC2, Zoolander reduced the SLO violation by 32% while ensuring 20 requests completed within 150 ms. Chris said that EC2 may fail SLO because of the nature of its energy saving features.

Dr. Wood asked questions about the real-time key-value stores with strong time constraints and deadline-aware scheduling. Chris replied that SLO can also be very important in those scenarios, with an example of scale-out modeling. Wood continued to voice concerns about batching workloads, which Chris agreed was important although not addressed in the presentation.

Preemptive Reduce Task Scheduling for Fair and Fast Job Completion

Yandong Wang, Auburn University; Jian Tan, IBM T.J. Watson Research; Weikuan Yu, Auburn University; Li Zhang and Xiaoqiao Meng, IBM T.J. Watson Research; Xiaobing Li, Auburn University

Yandong Wang started by discussing MapReduce and Hadoop, highlighting that the original Hadoop fair scheduling actually caused unfairness in a mixed workload of long batch jobs and small interactive jobs. He noted the monopolizing behavior in the Hadoop scheduling system, with short jobs having to wait until previously arrived large jobs finished the reduce phase, causing a delay of up to 19x according to the authors' experiment data.

Yandong presented their work to differentiate and preemptively schedule different workloads for fast and fair job completion. To achieve high efficiency and fairness for any mixed workloads, Yandong introduced a new mechanism called Preemptive ReduceTask. By coordinating all two-phase jobs and allowing lightweight work-conserving preemption, the authors claim that their methods can achieve high efficiency and fairness through eliminating the monopolizing behavior. Wang demonstrated their result with a 13% savings from their scheduling comparing with the original Hadoop scheduling. At the same time, they could achieve the same degree of fairness.

Chris (OSU) asked about the overhead and the size of HDFS to store the immediate data for such scheduling. Wang replied that they assumed that storage space is unlimited in their environment. Ioan shared the concerns of scheduling a heavy-duty workload in both cheap scheduling cost and fine-grained granularity. Eugen Feller asked about the performance of the authors' work in the Yahoo MapReduce framework. Wang explained that their method is a general approach, which could be easily extended to other platforms such as Yahoo! MapReduce.

QoS-Aware Admission Control in Heterogeneous Datacenters

Christina Delimitrou, Nick Bambos, and Christos Kozyrakis, Stanford University

Christina Delimitrou started her presentation about cloud DC admission control and scheduling by discussing how workload dynamics, such as characteristics and arrival time, could raise problems in the control.

Christina described two keywords in handling such problems: heterogeneity and interference. As a solution, she presented ARQ, the admission control protocol for cloud DCs, which has many merits as being application- and QoS-aware, scalable, and lightweight. She presented two baselines as classification scheduling and greedy scheduling. The authors built a small-scale platform with many different workloads. She showed the simulation data of their protocol to highlight their methods in scalability. Overall, based on experimental results, she claimed that ARQ provides the best performance in over-subscription scenarios.

Eugen asked about the definition of jobs and the dependency model among different workloads. Due to the limited time of this short presentation, she briefly replied that by her definition, the interference profiling could provide such information, which is discussed in detail in the paper.

Performance Inconsistency in Large Scale Data Processing Clusters

Mingyuan Xia and Nan Zhu, McGill University; Yuxiong He and Sameh Elnikety, Microsoft Research Redmond; Xue Liu, McGill University

Mingyuan Xia began by discussing state-of-the-practice large computing clusters with a focus on sharing. A physical cluster can be partitioned in many virtual clusters for different jobs. However, the fixed capacity of each cluster may cause problems in performance inconsistency. Mingyuan focused on Cosmos, a MapReduce environment from Microsoft with a fixed partition of different-sized VMs, which has the aforementioned problem.

Considering all cluster-related parameters at a given time point, Mingyuan formulated the problem as a MaxMin optimization problem and found its solution to provide the best fit for all work in a small set of VMs, and went on to discuss how it impacts the resource management and VM partition in long-term scheduling.

The authors evaluated their solution in under/overloaded scenarios with fast and slow virtual clusters. They found that a slow VM has a much higher chance of congestion and performance degradation when facing workloads with high variation. As future work, Mingyuan proposed a credit-based solution to grant different weights to different VM clusters based on historical usage information. There were no questions.