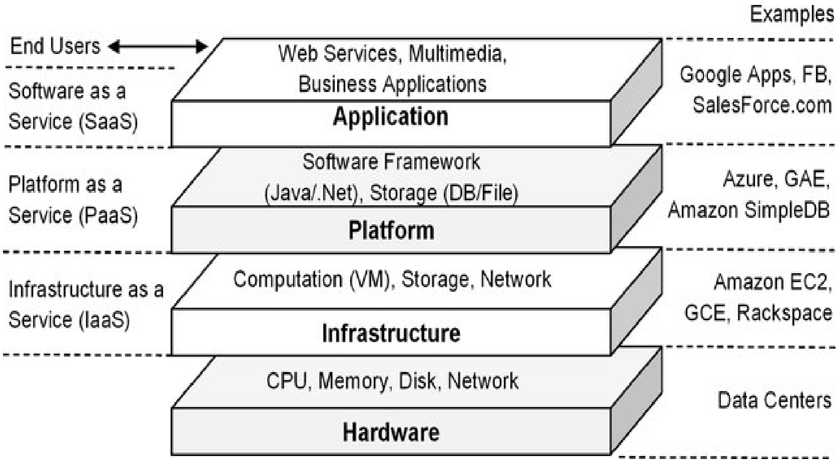
Approaches to improving the efficiency of data centers

Andrey Chupakhin

Lomonosov Moscow State University

Cloud Resource Scheduling and Optimization Workshop

Data centers



Efficiency of data centers

- Utilization of the physical resources (the utilization of computing resources and storage should be maximized and the workload of network resources should be minimized)
- Percentage of assigned requests for the creation of virtual networks relative to the number of arrived requests
- Performance of virtual machine
- Energy consumption

Efficiency of data centers

- Resources fragmentation – Migration, Live migration
- "Wide" virtual machines – NUMA

Virtual machines placement policies

For the virtual machines, the following assignment policies may be additionally specified:

- For each virtual machine, a set of physical servers on one of which it should be located; this is called the **VM-to-PM** affinity rules
- For each virtual machine, a set of physical servers on which it cannot be located; this is called the **VM-to-PM** anti-affinity rules
- The set of virtual machines that must be located on the same physical server; this is called the **VM-to-VM** affinity rules
- The set of virtual machines that must be located on different physical servers; this is called the **VM-to-VM** anti-affinity rules

Model of physical resources

Physical graph: $H = (P \cup M \cup K, L)$

- $(ph_1, ph_2, ..., ph_{n1}) = vh(p), p \in P$ – servers
- $(mh_1, mh_2, ..., mh_{n2}) = uh(m), m \in M$ – storages
- $(bh_1, bh_2, ..., bh_{n3}) = bh(k), k \in K$ – routers and switches
- $(lh_1, lh_2, ..., lh_{n4}) = rh(l), l \in L$ – physical links

Model of request

Virtual graph: $G = (W \cup S, E)$

- $(wg_1, wg_2, ..., wg_{n1}) = fwg(w), w \in W$ – virtual machines
- $(sg_1, sg_2, ..., sg_{n2}) = fsg(s), s \in S$ – virtual storages
- $(eg_1, eg_2, ..., eg_{n3}) = feg(e), e \in E$ – virtual links

Requests mapping

Placing request is a mapping:
$A : G \to H = \{W \to P, S \to M, E \to \{K, L\}\}$

- The physical resource cannot be overloaded: $\sum_{i \in R_j} x_i \leq y_j$, where R_j is the set of requests assigned to the physical resource j.
- The types of the physical and virtual resources must match each other: $x_i = y_j$
- The physical resource must have the required characteristics: $x_i \leq y_j$
- (servers) Not allowed to overload NUMA blocks on physical servers
- (servers) Each VM can only be assigned on one NUMA block of a specific physical server

The mapping is correct if the constraints 1-5 and the VM placement policies are satisfied for each request.

Residual resources graph

H_{res} with redefined functions are according constraint 1

- $vh_{res}(p) = vh(p) - \sum_{w \in W_p} fwg(w)$
- $uh_{res}(m) = uh(m) - \sum_{s \in S_m} fsg(s)$
- $bh_{res}(k) = bh(k) - \sum_{k \in R_k} feg(k)$
- $rh_{res}(l) = rh(l) - \sum_{e \in E_l} feg(e)$

Migration plan of virtual machines

- The sequence of single moving of virtual machines
- For each VM moving, the start and end of the migration are calculated
- All moving must be completed within the specified directive interval
- For each VM moving, the path for its migration is created in the physical resources graph

Problem of resources scheduling

- **Input**
  - New requests $Z = \{G_i\}$
  - Already assigned requests $B = \{G_j\}$ and its mapping $A_B : B \to H$
  - The residual graph of available resources: H_{res}
  - Migration time limit: T_{dir}
- **Find**
  - Find mappings:
    - $A_L : L \subseteq Z \to H_{res}$
    - $A_M : M \subseteq B \to H_{res}$
    - $|L| \to max$
  - Build a migration plan for a set M without violating the migration time limit T_{dir}

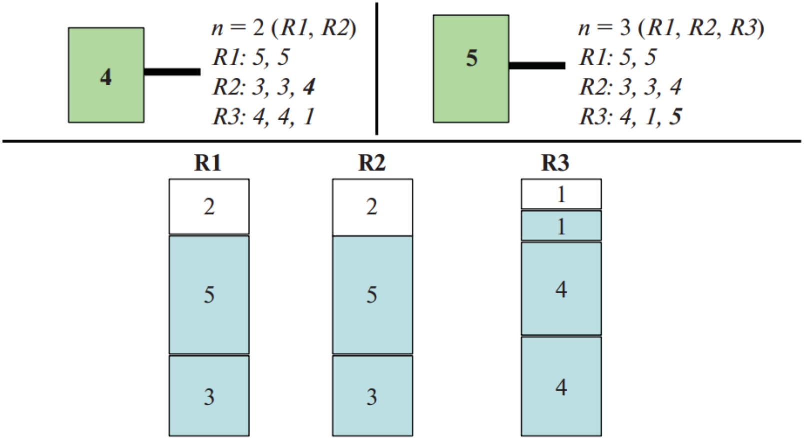
Algorithm

1. Sorting request ($C1$)
2. Choose request
3. Policies satisfaction
4. Choose element of request ($C2$)
5. Create set of physical servers according to 1-5 constraints (set A)
    - If $A! = \emptyset$
        - Choose physical element ($C3$)
        - Assigned element. Change H_{res}
    - If $A = \emptyset$
        - Limited research procedure. If unsuccessful **goto to Step 6**
        - Create migration plan. If unsuccessful **goto to Step 6**
6. Cancel assignment request elements

Greedy criterias

- Elements parameters: $(r_{e,1}, r_{e,2}, ..., r_{e,n})$
- Deficit for parameter: $d(i) = \frac{\sum_G \sum_{e \in E, E \in G} r_{e,i}}{\sum_{ph \in Ph} r_{ph,i}}$
- Cost of element: $r(e) = \sum_{i=1..n} d(i) * \frac{r_{e,j}}{\max_{G, v \in G} r_{v,i}}$
- $weight(e) = r(e) * \sum_l Throughtput(l)$
- C1 – choose request
- C2 – choose virtual element
- C3 – choose physical element
- C4 – choose NUMA block

Limited research procedure



Algorithm properties

- Ability to create migrate plan virtual machines
- Ability to define the required set of parameters for physical and virtual elements
- Ability of satisfaction of replacement policies for virtual machines
- Ability to consider NUMA architecture when assigne virtual machine
- Ability to provide the required balance between the algorithm's execution time and the quality of the resulting solution
- Ability to perform hard, soft, or hybrid policies
- Implemented algorithm can work in online mode, i.e. process requests sequentially

Conclusion from our research

- Algorithm with NUMA, Policies, Migration
- Tested on different datasets: real and synthetic
- The algorithm gives goods results on tree and fat-tree topologies
- Assign requests compactly – network utilization

Articles

- Comparing various approaches to resource allocating in data centers / P. M. Vdovin, I. A. Zotov, V. A. Kostenko et al. // Journal of Computer and Systems Sciences International. — 2014. — Vol. 53, no. 5. — P. 689–701.
- Zotov I. A., Kostenko V. A. Resource allocation algorithm in data centers with a unified scheduler for different types of resources // Journal of Computer and Systems Sciences International. — 2015. — Vol. 54, no. 1. — P. 59–68.
- Kostenko V. A., Plakunov A. V. Ant algorithms for scheduling computations in data processing centers // Moscow University Computational Mathematics and Cybernetics. — 2017. — Vol. 41, no. 1. — P. 44–50.
- Kostenko V. A., Chupakhin A. A. Approaches to improving the efficiency of data centers // Programming and Computer Software. — 2019. — Vol. 45, no. 5. — P. 251–256.
- Kostenko V. A., Chupakhin A. A. Live migration schemes in data centers // Programming and Computer Software. — 2020. — Vol. 46, no. 5. — P. 312–315.
- Vasil'eva Y. O., Kostenko V. A., Chupakhin A. A. Effect of virtual machine deployment policies on the efficiency data processing centers // Journal of Computer and Systems Sciences International. — 2020. — Vol. 59, no. 3.

Thank You!

- **Email**:
    - Valery Kostenko `kost@cs.msu.su`
    - Andrey Chupakhin `andrewchup@lvk.cs.msu.ru`
- **Code**: `https://github.com/andxeg/master_course`
    - branch **master**
    - branch **master_diploma_2018**
- **Code**: `https://github.com/andxeg/bachelor_diploma_2016`