Use cases for a new lease type in BLAZAR

The Computing Activities of a team in the Public Research and Public Administration are usually quite continuous, but of quite different intensity, over long periods of time (e.g. one year). The amount of computing resources effectively used by a team at a certain time may vary in a quite significant way. On these environments it is usual for the teams to stipulate with the Data Centers contracts for the provision of an average computing capacity to be guaranteed during a long period (e.g. one year) rather than of an agreed amount of computing resources that should be available at any given time.

In these Data Centers new hardware resources are acquired according to the user best estimates of the annual computing resources needed for their activities and partitioned among them. The partitioning policy is defined in terms of fractions of average usage of the total available capacity (i.e. the percentage of the Data Center's computing resources each team has the right to use averaging over a fixed time window). In order to respect the contracts, the administrators have to enforce that each stakeholder team, at the end of any sufficiently long period of time, and hence of the entire year, has got its agreed average number of resources. Moreover, since in general the request for resources is much greater than the amount of the available resources, it becomes necessary to seek to maximize their utilization by adopting a proper resource sharing model.

In the current OpenStack model, the resource allocation to the user teams (i.e. the projects) can be done only by granting fixed quotas to everyone. Such amount of resources cannot be exceeded by one group even if there are unused resources allocated to other groups. So, in such scenario, incoming requests are simply rejected, until somebody gives up his VM in which case the next person doing a request would get the free lease.

Past experience has shown that, when resources are statically partitioned (e.g. via quota) among user teams, it follows a very low global efficiency in the Data Center's resource usage and an increased cost for the user teams for their annual computing activities, compared with alternative more flexible and dynamic approaches allowing a continuous full utilization of all available resources. The main reason of these higher costs for a team comes from having to pay resources leaved idle for some periods rather than “borrowing” them to other teams more active in the same periods and getting in this way the possibility to get them back in greater amount when needed. In this way the Data Centers miss the opportunity to get a continuous saturation of usage of the total number of the available computing resources due to the discontinuities in the team's activities and are then obliged to charge these inefficiencies to users.

Typically and schematically each team requires virtual machines of three very different time durations in relations to different types of activities they need to carry out in the Data Centers:

1. Type1: unlimited duration time (UVM)
2. Type2: limited, but planned for a well defined date and for a medium-long (typically from 1 to 3 weeks) duration time (LVM)
3. Type3: limited but with short (typically from few hours to 1 day) duration time (SVM).

Activities of Type1 refer in general to services for hosting WEB sites or any data access services (e.g. FTP service, Metadata service or AAI services) which must be “forever” available.

Activities of Type2 refer to planned synchronous activities related to data analysis and visualization. People sit in front of a screen and interact with data using a cluster of VMs to produce quickly and interactively results that can be immediately analyzed to decide the next step to be done. In general these occur before important deadlines as those for the submission of a paper to a Conference in Science or the release of important results or reports in other domains, when scientists or managers need to go through sensible data to extract relevant information for their decisions or reports. This requirement is quite similar to the one addressed by the BLAZAR project (https://wiki.openstack.org/wiki/BLAZAR, previously known as climate) that was originally scoped for calendar based scheduling (such as in an HPC environment where the objective is to be sure to get 500 VMs from time X to time Y or reserving resources for a school class) and corresponds to the BLAZAR lease type “Delayed resource acquiring” or “scheduled reservation”.

Finally the activities of Type3 embrace an asynchronous model and refer to the serial systematic and repetitive sequential simulations or analysis of a large number of different data objects or data files. This is done usually by activating a large number of different analysis requests using the same application and virtual environment, but changing the input and output objects or files, while the start of new activities depend on the availability of new data sets or new improved applications. The teams do not have, in general, coincident periodic peaks of Type3 activities. There are periods in which only some teams are very productive while some others have small or no activities at all. Sometimes it happens that all teams are very active at the same time.

The Data Center administrators want to optimize their return of investment and the costs they charge the users for their average number of compute hosts of the data center they have acquired the right to use. They ask therefore the best algorithms for the dynamic allocation of all the available resources compliant with the following rules:

1. the utilization target for the computer center is 100%
2. at the end of the year the contract with each team must be respected
3. in addition also at the end of any configurable “medium-long” period (e.g. 2 days, a week, a month etc.) each team should have consumed exactly the assigned average fraction of usage of the total resources agreed in the contract

4. at any time any team having not yet saturated its own quota get VMs for its activities
5. there is no need for a team to have the fraction or average number of resources available at any instant, but point 3 has to be respected.

A new lease type approach respecting these rules is very different from the current Nova scheduling and from the lease types currently addressed by the BLAZAR project. Indeed the goal is to minimize the costs of usage of a number of compute hosts for users while still being able of complying a contractual average amount of resource provision and also to guarantee at any time the access to VMs to any team which has not yet saturated his contractual capacity in the defined time frame.

Definitely it is not a simple issue, not possible to implement with manual adjustments of the quotas, because the site administrator of large Data Centers has to take dynamic scheduling decisions to guarantee at the same time to its teams (of often several hundred) the satisfaction of conditions 1. to 5.

A possible technical choice for being able of optimizing the global usage and hence minimizing the costs for users is to enforce VMs of fixed duration for activities of type 3 to have a mechanism that generates systematically free resources when the time of a VM has expired. This enables regular possible changes of the teams that use the resources. Other mechanisms that could be used in case of competition of the Data Center resources is to introduce dynamically a limit in the time of life of a VMs assigned to a team if this has exceeded its average quota to give to the other teams the possibility to enter and increase their fraction until the contract is respected. Dynamically limiting VMs time would work for most user applications of the research sector. A good approach would be to first allow a user a chance to peacefully retire VMs themselves, then after waiting a while, terminate the VMs.

The voluntary retiring could be done with a slight addition to the API. e.g. something like 'nova target-show' which will tell the user how many VMs he should be using right now, so he can choose to reduce. This would be suitable for projects which use automated cloud provisioning systems, as many do.

When a resource becomes free, several teams will compete for them. The choice of the team to whom to give the first free resource must depend on the comparison of what has already been provided and thus is based on the number of incoming requests as well as the past usage. Storing incoming requests allows to prioritize incoming requests in a fair manner to ensure that each team get on average what they have been promised to get. Moreover, storing incoming requests instead of rejecting them immediately guarantees the possibility of keeping resources completely busy as users may not be polling the system manually at the right time.

To make it more clear, for simplicity let’s say that we have only two teams, "A" and "B" who are undertaking activities of Type3. The site administrator assigns at the beginning of a period of resource planning to "A" the 70% of the average total number of VMs that can be created on top of the available compute host (let’s call it “share” that is a concept different from the "quota" in the cloud's terminology) and therefore "B" has the remaining 30%. The site administrator wish to be able to make this partitioning dynamic, in such a way that "B" for “short” periods can use even all the resources belonging to "A" just if not used and vice versa in order of being able to always have saturated its total capacity while guaranteeing the average numbers in the “long” periods.

This corresponds to typical scenarios when the scientific teams or managers or public administrators require VMs for their asynchronous new data analysis processing characterized by serial activations of an application for a limited time window (i.e. the time necessary to complete the calculations inherent to a number of files or objects). Whenever the calculations have been completed the activity stop and the VMs are automatically released until a new activity starts, may be some weeks later.

This required optimization constitutes a well known IT problem which has already been tackled and solved in the past. The core component which handles the resource allocation optimization respecting rules 1 to 5 is the “fair share lease” which is made aware of the partitioning policy adopted and the historical resource usage. In particular it provides a dynamic priority algorithm and guarantees that the usage of the resources is finally distributed among users and teams according to their average number of compute hosts they have got in their contracts (i.e provide a fair-share lease) by considering the portion of the resources allocated to them in average (i.e. share) and the evaluation of the effective resource usage consumed in the recent past. The decision policy can take into account a very large variety of attributes required by the users by including, for instance, network and storage in addition to the typical ones as compute hosts and memory. Dynamic priority algorithms of this type have the merit to maximize the number of VMs a user can obtain for a certain cost at the price of some delay in reaching the agreed average values when for some time he was under-utilizing the system.

Activities of Type1 simply decrease the total amount of resources effectively available to a team for the fair share lease of activities of Type2 and Type3 during the whole time period taken into consideration by the site administrator for the enforcing of the share.

It would be interesting to discuss how much the features of the fair share lease could be still integrated in the new BLAZAR project and how we could collaborate at getting this done introducing in BLAZAR a new lease type. A lease with a scheduler for fair share appears as a good complement to the time based one and the work being done in BLAZAR on queues, submission to the cloud, termination on expiration and roles would be reusable almost ‘for free’.