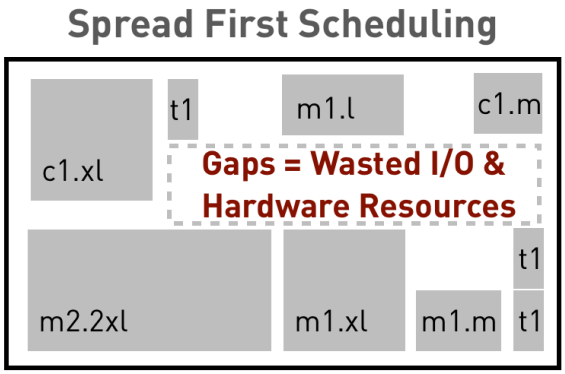
Cloud QoS and SLAs Mandate Efficient Scheduling

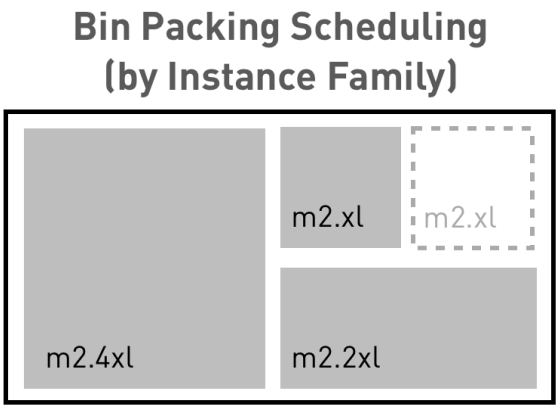
As data centers evolve from bare metal to virtualization to cloud provisioned application deployments, efficiently utilizing the compute, networking and storage resources is always a critical consideration. Storage technologies such as thin provisioning, data compression and de-duplication are examples of how storage is now being used more efficiently. Leveraging standardized image libraries, instance sizes and resource pools along with intelligent compute schedulers can similarly improve compute efficiency and deliver high return on investment.

Standard OpenStack Scheduling Doesn't Maximize Compute Node Utilization



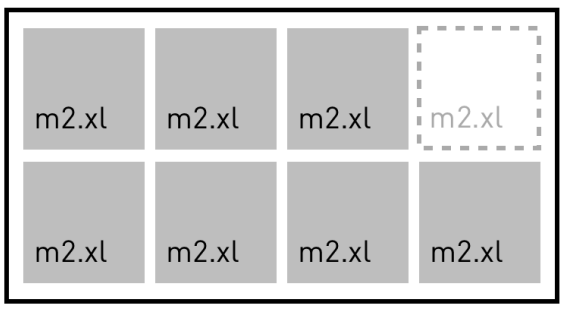
The OpenStack Nova Scheduler service takes requests for new virtual machines (VMs) and determines the best host server node to be used as the VM is provisioned. The scheduler's role is to apply algorithms that ensure efficient placement of VMs using a combination of filters and weightings that employ factors such as load, memory and CPU architecture. The standard OpenStack scheduling algorithm first filters out hosts with insufficient memory or compute resources and then the weighting selects the host with the most available memory. This is a "spread first" behavior that distributes VMs across all available nodes independent of the varying resource requirements of the VM instance types. This approach doesn't account for software to hardware cost/performance optimization.

AWS Scheduling Implements 'Bin Packing' to Map Instances into Resource Pools



AWS implements a bin packing scheduler to simplify capacity management, drive high utilization and ensure quality of service (QoS) guarantees. When running a production-grade cloud, even small inefficiencies will add up and significantly impact cloud utilization and VM performance.

When all the VM instances assigned to a physical node have similar ratios of memory to compute to storage, it's easy to manage the capacity of those instance types across the cloud and to ensure that nodes can be fully utilized without being over or under subscribed. When instances with varying resource ratios are assigned to a node, they will always have 'gaps' with wasted hardware and I/O resources. To ensure quality of service, collocating similar VMs on the same node ensures they have proportional access to network I/O which is not otherwise rationed by VM size. Having a node with one mission critical 2xl database instance and dozens of small instances can proportionally rob the database of critical compute cycles as well as networking and disk I/O.

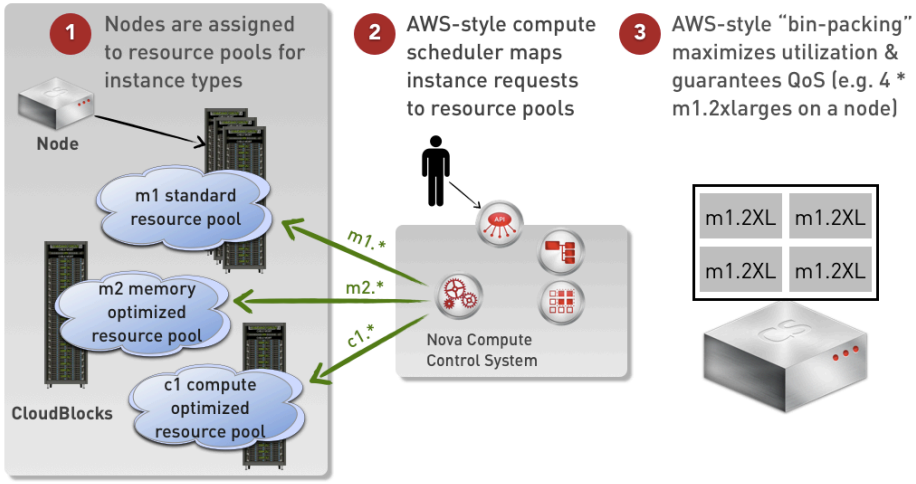


© 2013 The Cloudscaling Group, Inc.

As is common with most IaaS cloud deployments, AWS offers multiple instance sizes (e.g. small, medium, large, etc.) across multiple instance families (e.g. m1 standard, m2 memory optimized, c1 compute optimized, etc.). AWS bin packing schedules all the instances for a family into a common resource pool as the hardware is optimized to support the ratios of the instance type resource requirements.

Compute optimized instances are naturally deployed into resource pools of low memory server nodes and memory optimized instances are deployed into resource pools of high memory server nodes.

Cloudscaling OCS Delivers AWS-Style Scheduling to Maximize Utilization & QoS



As the Cloudscaling Open Cloud System (OCS) is modeled on AWS as a reference architecture, the OCS Compute Scheduler emulates the AWS approach to deliver behavioral fidelity (which translates to equivalent performance and efficiency). Cloudscaling implements AWS standard instance types as a default, and custom instances types can be defined that support varying compute, memory and disk allocations.

The Nova Scheduler implements a pluggable architecture that supports creation of custom algorithms to optimize placement of VM workloads. Cloudscaling developed the OCS Compute Scheduler as a specialized Nova plugin that **guarantees placement** of instances by mapping the requested instance type (all m2.xlarge) or instance family (all m2.*) to an optimized compute resource pool. This enables OCS to maximize utilization of compute nodes through bin packing and it **guarantees QoS** for the deployed VMs by enabling predictable management of subscription rates.

OCS Compute Scheduler Simplifies Capacity Management via Resource Pools

OCS facilitates capacity planning by grouping similar instance types on the same compute nodes, while also dispersing a single customer's instances across as many nodes in the resource pool as possible for fault tolerance. The OCS Compute Scheduler allows very explicit control of what instance types can be deployed to each node, and makes VM packing and oversubscription pretty easy to calculate and manage.

The OCS Scheduler leverages OpenStack host aggregates as a mechanism to further partition an availability zone. As new nodes added to the cloud contain hardware designed for a specific family of instances, the nodes are assigned to that instance type or family of instances, and only those instances are scheduled onto them.

cloudscaling
45 Belden Place
San Francisco, CA, 94104
Main: +1-877-636-8589
International: +1-415-508-3270
www.cloudscaling.com

© 2013 The Cloudscaling Group, Inc.

openstack CLOUD SOFTWARE

Cloudscaling is the trusted source for information on OpenStack and together with the community is making OpenStack more production-grade. For more information, please visit www.openstack.org.