

ALTO
Internet-Draft
Intended status: Standards Track
Expires: July 8, 2014

Y. Lee
Huawei Technologies
G. Bernstein
Grotto Networking
D. Dhody
Huawei Technologies
T. Choi
ETRI
January 4, 2014

**ALTO Extensions for Collecting Data Center Resource Information
draft-lee-alto-ext-dc-resource-03**

Abstract

In a networked application environment where resources (e.g., computing, storage, etc.) are geographically distributed in Data Centers, a key decision is to allocate the application request to an "optimal" Data Center location in which to host the application request. Key constraints in this decision include resource availability, network cost, infrastructure constraints (e.g., power) and others. This draft proposes an ALTO extension to support data center resource information model and its related protocol extensions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 8, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
1.1.	Requirements Language	4
2.	Data Center Compute Resource Models	4
2.1.	Current IaaS User Resource Models	4
2.1.1.	Cost or Pricing Models	5
2.2.	Internal Data Center Resource Models	5
3.	ALTO-Interface Data Center Resource Information Model	6
4.	ALTO Protocol Extension	6
4.1.	Pull Based Query/Response	7
4.2.	Push Based Query/Response	8
5.	Security Considerations	8
6.	IANA Considerations	8
7.	References	9
7.1.	Normative References	9
7.2.	Informative References	9

[1.](#) Introduction

In a networked application environment where resources (e.g., computing, storage, etc.) are geographically distributed in Data Centers, a key decision is to allocate the application request to an "optimal" Data Center location in which to host the application request. Key constraints in this decision include resource availability (e.g., memory, storage, CPU, etc.), DC network cost, DC network resource constraints (e.g., bandwidth), structure constraints (e.g., Data Center power consumption) and others.

This draft describes data center resource information model in the context of i2aex (infrastructure to application exchange) and proposes its related ALTO protocol extensions.

The following figure depicts the key components in a networked application environment where Data Centers provide resources for an application.

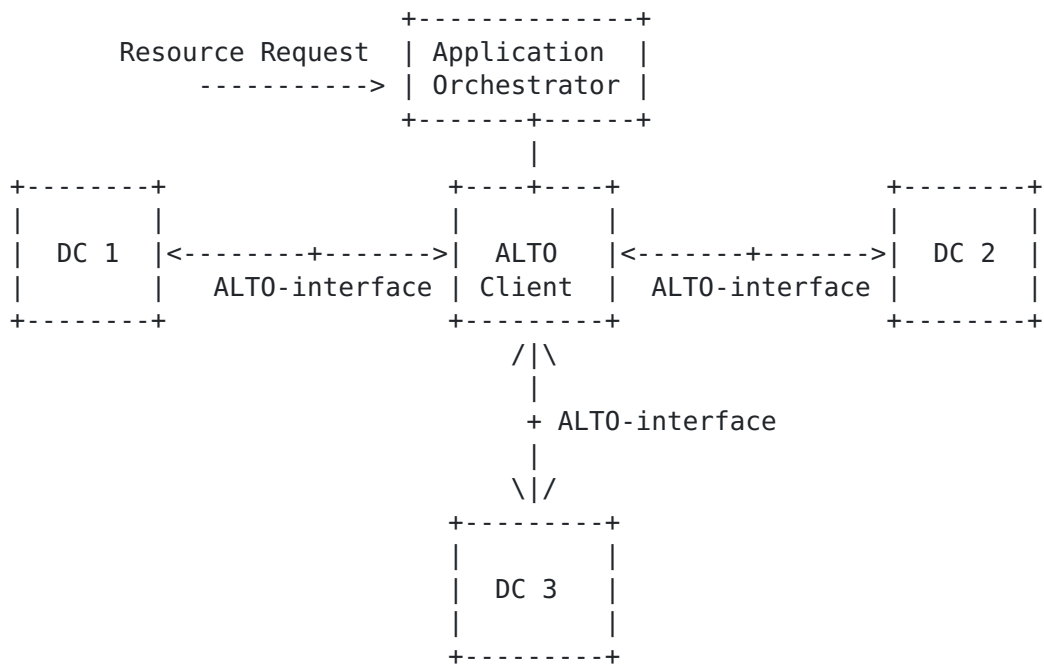


Figure 1: ALTO Architecture in Distributed Data Center Networks

Figure 1 shows that ALTO Client can establish an ALTO-interface to each data center to collect the abstracted data center resource information and then select an "optimal" data center location based on the collected resource information for the resource request.

Resource request arrives to the "application orchestrator" which is a separate functional entity from the ALTO Client. The collected Data Center resource information by the ALTO Client needs to be sent to the "application orchestrator" where the optimal data center selection is made. How the application orchestrator interfaces with the ALTO Client is out of the scope of this document. Also note that the "application orchestrator" and the ALTO client maybe co-located.

The potential information to be shared concerning capacity, performance, structure, and/or network costs associated with a data center may be considered sensitive to the data center owner/operator. It is assumed that ALTO client interfaces with data centers in a trusted relationship.

Combined compute and network resource optimization is of value to both application owners and data center operators. For example a data center operator with multiple buildings in a metropolitan area may also want to balance compute and network costs. When looking to

model compute resources we consider both application owner and data center owner perspectives.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

2. Data Center Compute Resource Models

2.1. Current IaaS User Resource Models

In a typical infrastructure as a service (IaaS) data center model, application software runs within one or more virtual machines (VMs). These VMs are allocated to physical hardware by IaaS scheduling software where they are run under the supervision of a virtualization hypervisor.

To achieve a given level of performance a particular VM within an application needs a specified amount of compute resources. The raw compute resources are (fast dynamic) memory, virtual CPUs (VCPUs), and dedicated local disk storage. One can think of a virtual CPU as an execution thread on a processor core, however, the notion maybe hypervisor specific. [Editor's note: we have currently only looked at details on the Xen hypervisor.]

Currently IaaS data center operators offer a fixed number of combinations of resources (memory, VCPUs, local disk) on which an application may run a VM. These are referred to as instance types.

[TO DO: give examples of some instance types from Amazon or OpenStack.]

A scalable application is typically implemented so that it can run on a number of VMs with the number varying depending on load or other conditions. Different VMs may assume different roles in the application, e.g., a controller/dispatcher, request processor, data base engine, etc... These different roles may dictate the use of different instance types for the different VMs.

We are led to a model where an application will utilized a variable number of VMs over time. These VMs may play different roles within the application and hence require different combinations of processing resources.

The data center can furnish a limited number of instance types (combination of resources) for an application to use. In addition

there is a finite amount of resources that the data center may have to offer a particular application.

Currently two different approaches appear to be used by IaaS providers: (a) Provide no information about available compute capacity and respond positively or negatively to requests for resources, i.e., a specified number of VM instances of a given type. (b) Provide overall quotas (limits) on memory, number of VCPUs, and local storage [[OpenStackQuotas](#)].

In this document, we consider the latter model. In such case, ALTO client will collect the overall data center resource information: memory, numbers of VCPUs, local storage, etc. This point will be elaborated in details in [Section 2.2](#).

[2.1.1](#). Cost or Pricing Models

IaaS service providers have introduced a number of pricing models. One provider [[EC2Price](#)] currently offers three different models based on the notions of (a) reserved instances, (b) on demand instances, and (c) spot instances.

For the more complicated models http based interfaces are given to facilitate queries and purchases. [TBD: Are there generic or parameterized pricing models that could represent a significant fraction of important cases?]

The pricing models discussed in this section are envisioned to be implemented by application orchestrator shown in Figure 1. As the user interacts with the application orchestrator and sends its request, the application orchestrator can make decisions whether it should honor the request or not based on the collected information via the ALTO client interface. The information collection to the ALTO client from various data centers is the critical piece that facilitates this process of selecting the optimal data center.

[2.2](#). Internal Data Center Resource Models

When previously discussing data center resources from an application perspective, the IaaS provider abstracts away the specifics of the hardware to a large degree. However when an IaaS provider is seeking to minimize its costs to provide services then the particulars of hardware resources are important: in particular, the cost of power at one site versus another site, the efficiency of physical hosts in delivering a given number of VCPUs and/or memory. Such information along with actual hardware capacity and usage can be used to weigh data center resource costs relative to bandwidth costs.

[To do: compare Amazon's ECU (elastic compute unit) with VCPU notion or other hypervisor computation unit notions.]

3. ALTO-Interface Data Center Resource Information Model

ALTO Client collects a set of the abstracted resource information from each participating Data Centers. The following information is the list of Data Center resource abstract information that will give the ALTO Client a good level of abstracted view of the status of each participating Data Center.

This collected information will be used for the ALTO Client to find the data center where the requested resource can be provided (via an interface with the application orchestrator).

- o Data Center Identifier (DCI)
- o Data Center Location Identifier (e.g., IP address of the gateway node)
- o Time Stamp
- o Abstracted Memory Usage
- o Abstracted CPU Level
- o Abstracted Power Consumption Level
- o DC Network cost
- o DC Network resource constraints

The list above is not an exhaustive list and can be expanded. Note also that how to represent physical resources information to abstract information is out of the scope of this document and is subject to further research.

4. ALTO Protocol Extension

```
Information Model:
object
{
  DCInfo dc-info;
} InfoDCResource : ResponseEntityBase;

object
{
  VersionTag          vtag; [OPTIONAL]
  JSONNumber          dcid;
  TypedEndpointAddr   addr;
  JSONString          timestamp;
  JSONNumber          ramusage;
  JSONNumber          srvload;
  JSONNumber          powerusage;
  JSONNumber          cost;
  ...
} DCInfo;
```

4.1. Pull Based Query/Response

In this case, the ALTO client should make HTTP request for DC Info to the ALTO server. The ALTO server will respond only when the request is made by the ALTO client.

```
GET /getdcinfo HTTP/1.1
Host: alto.example.com
Accept: application/alto-dcinfo+json,application/alto-error+json

HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-dcinfo+json
{
  "meta" : {},
  "dc-info" : {
    "vtag" : ["1266506139"],
    "dcid" : 1,
    "addr" : ["ipv4: 10.18.51.151"],
    "timestamp" : "02 January 2014 0000H UTC",
    "ramusage" : 60,
    "srvload" : 25,
    "cost" : 10
  }
}
```

4.2. Push Based Query/Response

In this case, the ALTO client should make an initial HTTP request for DC Info to the ALTO server. Further the ALTO server can continue to refresh the information to ALTO client.

Initial Request and Response is similar to [Section 4.1](#).

Further the ALTO server MAY refresh by sending Response on its own, based on future ALTO server notification/refresh mechanisms.

[Editor's Note: should be aligned to the ALTO server notification/refresh mechanisms]

5. Security Considerations

TBD.

6. IANA Considerations

TBD

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

7.2. Informative References

[OpenStackQuotas]
"OpenStack Quotas", <<http://docs.openstack.org/trunk/openstack-ops/content/quotas.html>>.

[EC2Price]
"Amazon EC2 Pricing",
<<http://aws.amazon.com/ec2/pricing/>>.

Authors' Addresses

Young Lee
Huawei Technologies
1700 Alma Drive, Suite 500
Plano, TX 75075
USA

EMail: leeyoung@huawei.com

Greg M. Bernstein
Grotto Networking
Fremont, California
USA

EMail: gregb@grotto-networking.com

Dhruv Dhody
Huawei Technologies
Leela Palace
Bangalore, Karnataka 560008
INDIA

EMail: dhruv.ietf@gmail.com

Tae-Sang Choi
ETRI
161 Gajong-Dong
Yusong-Gu Daejeon
Republic of Korea

EMail: choits@etri.re.kr