

Adaptive Computing Enterprises

Технический Комментарий

Технический комментарий для компании Модуль / редакция 1.1

Юрий Рассохин, региональный менеджер в странах
Центральной и Восточной Европы и Ближнего Востока
8/8/2014

Corporate Headquarters 1712 S. East Bay Blvd., Ste. 300, Provo, UT 84606 USA | P (801) 717-3700 | F (801) 717-3738

EMEA Office 3000 Cathedral Hill, Guildford GU2 7YB UK | P +44 1483-243578 | F +44 1483-243666

Asia Pacific Office 9 Battery Road, 12th Floor, Straits Trading Building, Singapore 049910 | P +65 6597-7053 | F +65 6597-7099

Содержание

1. Портфолио Adaptive Computing	3
2. Архитектура Moab	3
3. «Всё есть ресурс»	4
4. Пример планирования ресурсов на основе MTBF	9
5. Пример планирования ресурсов с учетом инфраструктуры	9
6. Планирование ресурсов между удаленными ЦОДами	10
7. Топологии объединения ЦОДов под управлением Moab	11
8. Независимость от нижележащей платформы	14
9. Планирование IT-бюджета в Moab	14
10. Подключение «вендорских» средств в Moab в качестве ресурсов	16
11. Экономический обзор применения решений на базе Moab	17
12. Примеры клиентских решений на основе Moab	18

1. Портфолио Adaptive Computing

Компания Adaptive Computing развивает два семейства программных решений:

- Moab Cloud Suite,
- Moab HPC Suite.

Оба семейства предназначены для оптимального управления ресурсами ЦОДа в процессе обеспечения пользовательских сервисов при заданных условиях обслуживания пользователей.

Moab Cloud Suite специализирован для управления ресурсами ЦОДов в рамках облачной модели обслуживания. Moab Cloud Suite реализует как частные, так и гибридные облака, с акцентом на модель предоставления сервисов IaaS. В целом, Moab Cloud Suite предназначен для широкого круга задач, естественным образом решаемых в облачной инфраструктуре.

В свою очередь, Moab HPC Suite специализирован для круга задач High Performance Computing (HPC) и родственных задач High Throughput Computing (HTC) и Big Data (посредством интеграции с Intel Hadoop).

Оба семейства решений построены вокруг единого программного компонента, Moab workload management engine. Moab можно неформально назвать «движком» принятия решений по оптимальному выделению ресурсов для обеспечения требуемых сервисов при заданных политиках и ограничениях по использованию ресурсов ЦОДа.

Каждое из двух семейств включает множество модулей, поставляемых по умолчанию или дополнительно, в зависимости от требований конкретного клиента. В качестве примеров таких модулей можно привести веб-портал самообслуживания пользователей, модуль построения отчетности по использованию ресурсов, и многие другие.

Однако, большинство уникальных возможностей решений Adaptive Computing следует из возможностей «движка» Moab, который выступает ядром обоих семейств. В силу исключительной роли «движка» для понимания возможностей продуктов Adaptive Computing, ниже в документе термином Moab обозначается сам «движок», а полным наименованием Moab Cloud Suite либо Moab HPC Suite – то или иное решение, построенное на базе Moab. В зависимости от требований конкретного клиента, комплектация и конфигурация решений на базе Moab может отличаться от общего описания, приведенного ниже.

2. Архитектура Moab

Ключевым архитектурным компонентом, в том числе первым исторически, в решениях Adaptive Computing является «движок» Moab – «движок» принятия решений на основе заданных условий обслуживания (Service Level Agreement, SLA). Весь прочий функционал с

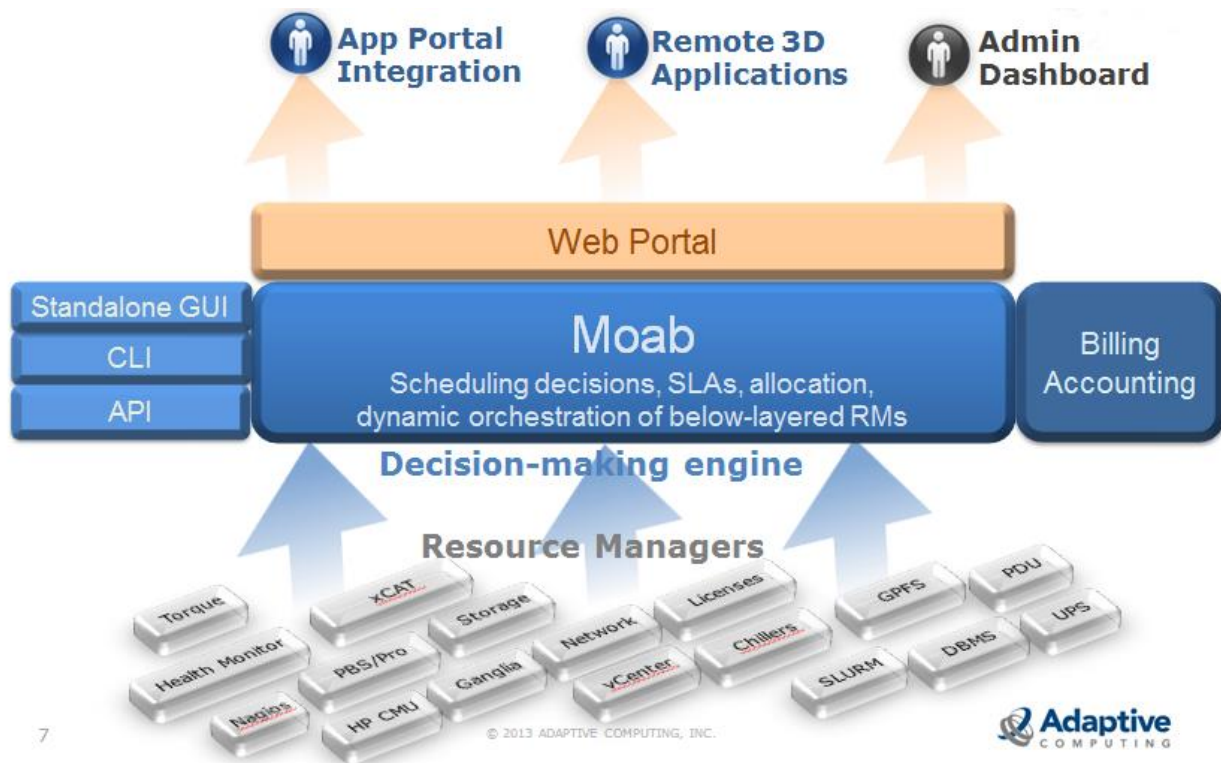
архитектурной точки зрения является надстройками над Moab. Примером противоположного подхода в области Cloud могут послужить коммерческие программные решения для построения виртуализированной облачной инфраструктуры, изначально являвшиеся гипервизором, и позже выросшие из гипервизора до более или менее законченного решения. С другой стороны, в области HPC/HTC Moab отличается уникальная архитектура, изначально построенная вокруг планирования на основе SLA, и не привязанная к конкретным видам управляемых ресурсов.

Как следствие, данные архитектурные особенности делают Moab в равной мере эффективным при построении инфраструктуры как для HPC, так и для Cloud. Универсальность обусловлена проектированием Moab с нацеленностью на решение задач конечных пользователей в терминах, близких человеку при решении бизнес-задач. В частности, отметим простое следствие из архитектуры Moab: оба решения, Moab Cloud Suite и Moab HPC Suite могут интегрироваться для управления одной и той же инфраструктурой ЦОДа. В этом случае, получаем гибкую, конвергентную инфраструктуру, динамически перестраивающуюся под обеспечение классических облачных сервисов либо HPC-сервисов, в соответствии с текущими запросами.

Приведенные особенности архитектуры Moab важны в дальнейшем изложении, поскольку именно из них вытекают характерные преимущества решений Adaptive Computing. Ниже описаны некоторые из этих преимуществ.

3. «Всё есть ресурс»

Moab способен рассматривать почти все измеримые сущности, которые представлены в ЦОДе, в качестве ресурсов, от которых Moab может получать данные, и на основании которых, в свою очередь, может принимать решения по планированию этих ресурсов для пользовательских сервисов:



Приведенная выше иллюстрация описывает базовую трёхуровневую архитектуру Moab:

1. На верхнем уровне «движок» Moab реализует абстрактные алгоритмы принятия решения по планированию абстрактных ресурсов, без привязки к конкретным видам ресурсов.
2. На нижнем уровне расположены конкретные ресурсы ЦОДа.
3. Средний, промежуточный уровень представлен тонкой прослойкой между Moab и реальными ресурсами. Эта прослойка составлена из так называемых ресурсных менеджеров, Resource Managers (RM). Ресурсные менеджеры реализуют интерфейс между Moab и каждым видом ресурса. На практике, ресурсные менеджеры, как правило, представляют собой тривиальные скрипты.

Связь между Moab и конкретными ресурсами в общем случае двунаправленная. С одной стороны, Moab запрашивает у ресурсного менеджера текущее значение управляемого им ресурса. С другой стороны, Moab передает ресурсному менеджеру ультимативные директивы по выполнению тех или иных действий над управляемым ресурсом, в зависимости от решения, принятого Moab. В зависимости от природы конкретного ресурса, ресурсный менеджер может реализовывать лишь часть стандартных сообщений, которые Moab может передавать этому

ресурсному менеджеру. Типичную ситуацию такого вида представляет ресурс, значение которого можно считывать, но управление таким ресурсом технически невозможно либо бессмысленно. Примером является совокупность датчиков температуры в ЦОДе. Moab может отслеживать значения температуры в разных точках ЦОДа, чтобы предотвратить размещение новых сервисов на самых горячих серверах. Однако, управлять датчиками температуры технически невозможно. В таком случае, ресурсный менеджер, связывающий Moab с датчиками температуры, реализовывает для Moab лишь возможность опроса текущих значений ресурса.

К ресурсам Moab, очевидно, относятся процессоры, память, диски. Однако, в равной мере ресурсами Moab, которые принимаются во внимание при обеспечении пользовательских сервисов, могут быть

- Менее очевидные характеристики серверных платформ, такие как относительный уровень загруженности в течение заданного отчетного периода, или усредненный исходящий трафик на заданном сетевом интерфейсе.
- Фабрики сетевых устройств.
- Сетевые хранилища.
- Источники бесперебойного питания (UPS).
- Управляемые модули распределения питания (PDU).
- Внешние системы мониторинга ЦОДа, такие, как Ganglia или Nagios.
- Внешние программные системы инвентаризации ПО.
- Внешние программные системы управления серверами, как правило, «вендорские» решения, поставляемые вместе с аппаратными платформами.
- Ленточные системы резервного копирования.
- Системы охлаждения, в том числе датчики влажности и температуры в ЦОДе.
- Фермы различных гипервизоров, таких, как KVM и VMware.

Список ресурсов, на основании которых Moab принимает решения, не ограничен приведенным выше. В принципе, потенциальный ресурс для Moab – это любая сущность, доступная для Moab по сетевому протоколу с тем, чтобы получить от Moab команду и передать значение ресурса в ответ.

Откуда возникает подобная гибкость включения в Moab сторонних ресурсов?

Она следует из архитектуры Moab: как отмечено выше, Moab – это абстрактный «движок», который реализует политики принятия решений по оптимальному использованию ресурсов. Этот «движок» не привязан к какому-либо конкретному виду ресурса. Moab реализует, не более и не менее, алгоритмы планирования абстрактных ресурсов. В принципе, это означает – любых ресурсов.

Разумеется, законченное решение Moab Cloud Suite включает не только «движок» Moab, но и интегрированные «из коробки» наиболее популярные ресурсы: процессоры, оперативная память, диски, и другие. Однако в дальнейшем, если администратору ЦОДа потребуется подключить к Moab новый вид ресурса, он может сделать это динамически, по мере необходимости.

Задача подключения нового вида ресурса к Moab сводится к написанию нескольких простых скриптов (как правило, от 1 до 4), каждый из которых должен реализовывать одну из стандартных функций: «опросить данные ресурса», «выделить заданное количество ресурса», «освободить указанную долю ресурса». Дополнительно, еще одна команда в конфигурационном файле Moab должна зарегистрировать в Moab новый вид ресурса с заданным именем (например, «температура»). После этого к новому виду ресурса применимы все алгоритмы планирования ресурсов, которыми обладает «движок» Moab. Разумеется, в зависимости от природы ресурса, часть алгоритмов планирования может не иметь смысла применительно к этому виду ресурса. Однако, ценность трехуровневой архитектуры Moab заключается именно в том, что Moab как таковой не накладывает никаких внутренних ограничений на возможности планирования любого вида ресурса.

В качестве простого примера, интегрируем датчики температуры на серверах ЦОДа в Moab в качестве управляемого ресурса. После этого Moab сможет выделять ресурсы под требуемые сервисы, учитывая распределение температуры в ЦОДе.

Шаг 1. Создадим простой скрипт, который получает данные температуры с серверной платформы. Этот скрипт может выглядеть следующим образом:

```
#!/usr/bin/perl
# 'hwctl' outputs information in format '<NODEID> <TEMP>'

open(TQUERY,"/usr/sbin/hwctl -q temp |");
while (<TQUERY>) {
    my $nodeid,$temp = split /\w+/;
    $dstage=GetDSUsage($nodeid);
    print "$nodeid GMETRIC[temp]=$temp
    GMETRIC[dstage]=$dstage";
}
```

Шаг 2. Одной командой в конфигурационном файле Moab регистрируем в Moab новый вид ресурса под названием «temp», указывая для него наш скрипт в качестве функции «опросить данные ресурса»:

```
RMCFG[temp] TYPE=NATIVE CLUSTERQUERYURL=file://<..>/node.query.local.pl
```


В скобках отметим: формат параметра CLUSTERQUERYURL подсказывает, что функция опроса ресурса может быть реализована как в локальном файле, так и на удаленном сетевом ресурсе, доступном по протоколу типа HTTP. Тем самым, достигается максимальная гибкость при интеграции Moab с корпоративными ресурсами, построенными в виде сетевых сервисов. Характерным примером может выступить интеграция серверов лицензий в Moab в качестве управляемого ресурса, что позволит Moab выполнять оптимальное распределение лицензий на дорогостоящее коммерческое ПО на уровне организации в целом – как для департаментов, так и для отдельных пользователей.

Шаг 3. Даем Moab команду перечитать файл конфигурации.

После этого Moab рассматривает температуру на серверах ЦОДа в качестве одного из своих ресурсов. В этом можно удостовериться, опросив состояние одного из серверов. Вывод должен включать показатель нового ресурса “temp”:

```
> checknode cluster013  
...  
Generic Metrics: temp=113.2, dstage=23748  
...
```

В приведенном выше примере Moab обладает информацией о том, что текущая температура на кристалле процессора на сервере cluster013 составляет 113.2 градуса. Этот факт может быть принят во внимание «движком» Moab при планировании будущей нагрузки на сервере cluster013.

Теперь администратор ЦОДа может создавать политики выделения серверов для пользовательских сервисов с учетом температуры. Приведем примеры популярных политик, связанных с температурой, которые администратор может задать в Moab:

- «Поддерживай активные сервисы максимально плотно, в минимально возможном количестве монтажных шкафов, а серверы в неиспользуемых шкафах держи выключенными».
- «Распределяй сервисы по всему ЦОДу так, чтобы температура в ЦОДе была равномерной, без горячих и холодных островов».
- «Если не указано иное, старайся выделять новые сервисы на наиболее холодных серверах».
- «Если температура входного потока воздуха в каком-либо сервере превышает 40 градусов на выходе, переставай давать новые сервисы на этот сервер. Если в течение 5 минут температура на выходе не падает до 35, начинай снимать сервисы. В первую

очередь старайся снимать те, которые занимают наибольшую долю процессора. Если приоритет пользователей 10 наиболее агрессивных по процессору сервисов не позволяет их снять, то распредели их по разным серверам. Если приоритеты пользователей не дают сделать и это, то пробуй снимать 10 сервисов, агрессивных по дискам и памяти. Если и после этого температура не упадет до 35 в течение 5 минут, вышли по электронной почте уведомление администраторам ЦОДа и пользователям сервисов, и выключай этот сервер».

4. Пример планирования ресурсов на основе MTBF

Ресурсом может быть не только температура, но и менее тривиальные параметры, такие, как MTBF (Mean Time Between Failures). Параметр MTBF является родственным принятому в России параметру «наработка на отказ».

Поскольку Moab владеет всей историей использования ресурсов на серверах ЦОДа, то он имеет встроенную возможность планировать ресурсы с учетом MTBF. Вот пример широко используемой политики планирования ресурсов на основе MTBF:

- «Поступающие сервисы в первую очередь распределяй на серверы, минимально использованные в течение предыдущего календарного года. С одним исключением: к серверам IBM Power и системам IBM System z это не относится».

5. Пример планирования ресурсов с учетом инфраструктуры

Другими важными примерами ресурсов, на основании которых Moab может принимать решения, выступают

- Пропускная способность (bandwidth) сетевых хранилищ.
- Пропускная способность фабрик сетевого оборудования, такого, как 1/10/40GbE Ethernet, FC, InfiniBand.
- Латентность (latency) подключений ядра ЦОДа к различным сетевым фабрикам.
- Латентность между «островами» внутри каждой фабрики.
- Один из наиболее важных параметров: коэффициенты «бутылочного горлышка» пропускной способности между разными «островами» сетевой фабрики. Этот параметр может быть исключительно важен при принятии решений относительно миграции по сети тяжеловесных виртуальных машин.

В частности, несложно построить политику, по которой Moab будет распределять сервисы в ЦОДе равномерно с точки зрения подключения к сетевому хранилищу – с тем, чтобы активные сервисы не создали «бутылочного горлышка» ни на одном канале подключения

распределенного сетевого хранилища. То же относится к политикам, предохраняющим от создания «бутылочных горлышек» на сетевых коммутаторах.

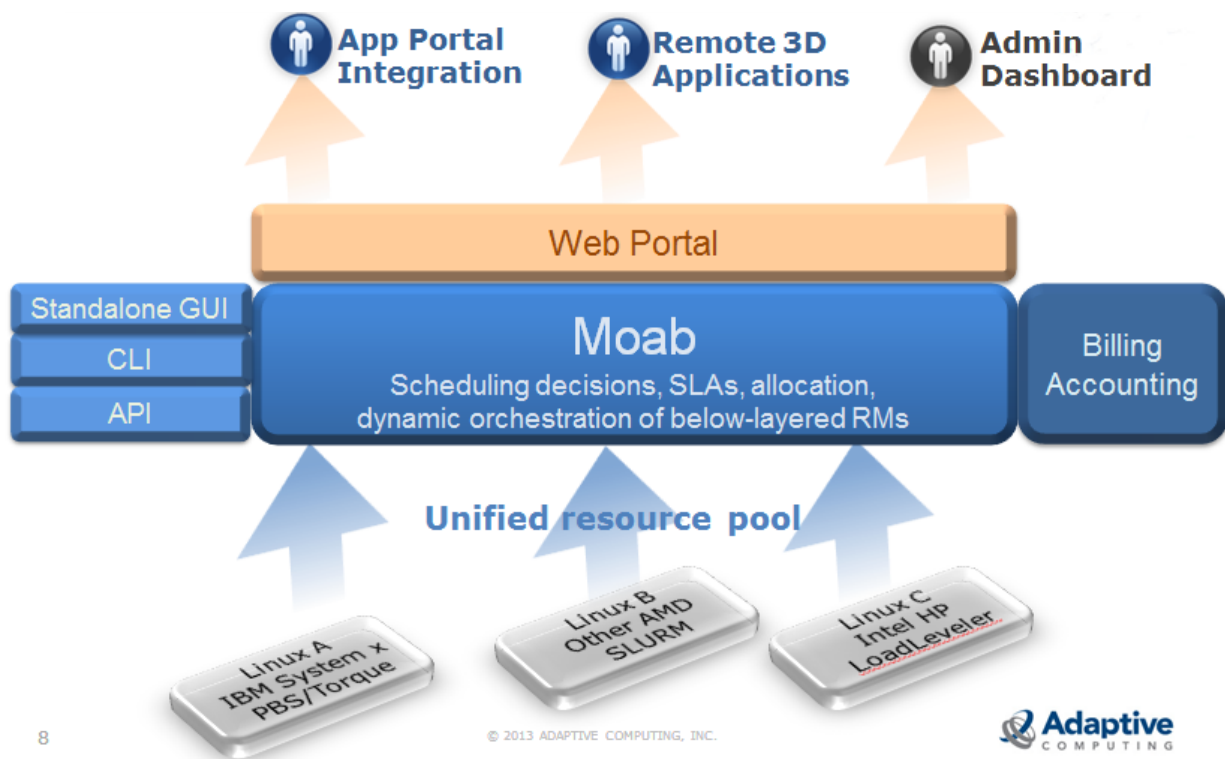
6. Планирование ресурсов между удаленными ЦОДами

Архитектурный принцип Moab отделения абстрактных алгоритмов планирования ресурсов от конкретных видов ресурсов имеет еще одно исключительно важное следствие: отдельно взятый ЦОД как таковой также может быть управляемым ресурсом Moab. Другими словами, Moab может управлять распределением ресурсов между удаленными площадками.

По существу, каждый географически локализованный ЦОД с точки зрения Moab может рассматриваться как один из видов ресурса. Причем, для этого не требуются какие-либо искусственные надстройки, поскольку такая возможность элементарно вытекает из самой архитектуры «движка» Moab.

В качестве примера, пусть ЦОД №1 представляет из себя ферму гипервизоров KVM под управлением oVirt, а ЦОД №2, расположенный в другом городе, представляет собой ферму гипервизоров VMware vSphere под управлением VMware vCenter. Moab интегрирует в себя каждый из ЦОДов благодаря реализации двух ресурсных менеджеров, которые обеспечивают связь, соответственно, с oVirt и vCenter – то есть, с теми сущностями, которые непосредственно управляют фермами гипервизоров, каждая по-своему в каждом из двух ЦОДов. Однако, после того, как Moab «накрыл» собой оба ЦОДа, запросы от пользователей обоих ЦОДов поступают уже в Moab, который получает, хранит и обрабатывает их в своем внутреннем представлении, не зависящем ни от одного, ни от другого типа гипервизора. Следовательно, когда Moab получает запрос от пользователя на выделение сервиса, он имеет возможность решить, в каком ЦОДе и на каком гипервизоре выделить ресурсы, нужные для запуска этого сервиса. При этом пользователь в принципе может не знать, где и как именно его сервис будет запущен.

Более того, в процессе управления очередью запросов от пользователей, Moab может динамически переносить запросы между очередями запросов, изначально нацеленными на разные гипервизоры.



Описанная возможность Moab – прозрачное управление сервисами между разнородными площадками – в общем виде является уникальной.

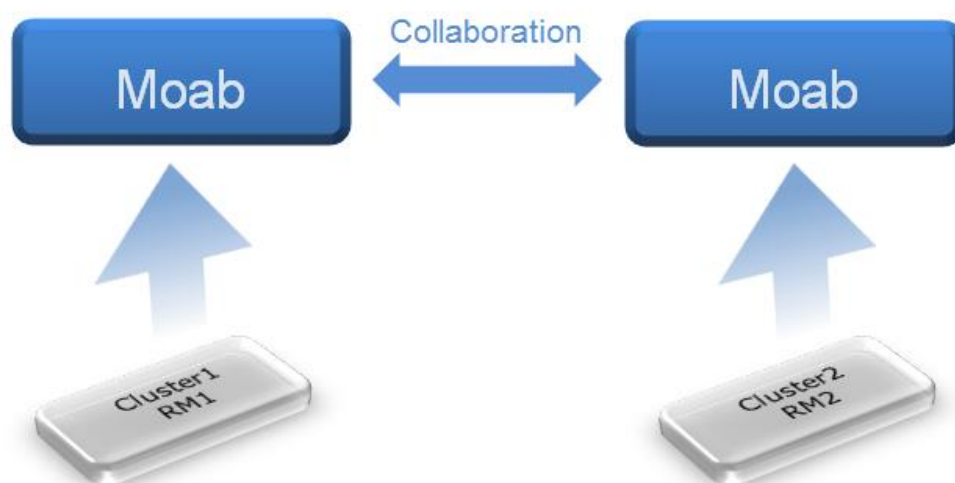
7. Топологии объединения ЦОДов под управлением Moab

При объединении удаленных площадок под управлением Moab, естественным образом выделяются три поддерживаемые топологии (либо сочетания нескольких из этих трех топологий, что также допустимо).

- **Peer-to-Peer.**

Есть ровно два ЦОДа, каждый управляется своим экземпляром Moab. По мере заполнения ресурсов «своего» ЦОДа, Moab этого ЦОДа передает новые запросы на обработку в Moab другого ЦОДа. Также возможна миграция сервисов между ЦОдами и по другим политикам, связанным не только лишь с недостатком локальных ресурсов. Общий смысл топологии Peer-to-Peer заключается в наличии двух равноправных ЦОДов, которые обмениваются сервисами.

Peer-to-Peer



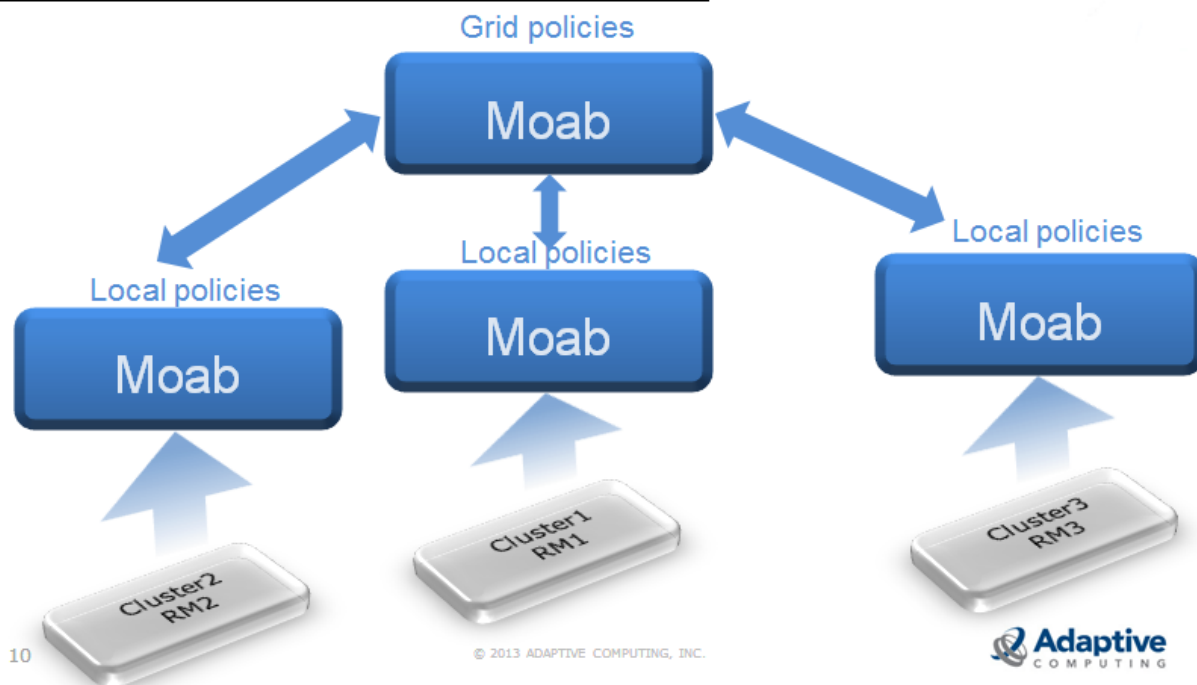
9

© 2013 ADAPTIVE COMPUTING, INC.

- **Source Destination Topology.**

Есть множество ЦОДов, каждый управляется своим экземпляром Moab. Над всеми ЦОДами есть еще один, центральный экземпляр Moab, который управляет распределением ресурсов между ЦОДами. Суть этой топологии в том, что она, в отличие от предыдущей, позволяет реализовывать сколь угодно сложные политики распределения ресурсов между удаленными ЦОДами, в общем случае не равноправными.

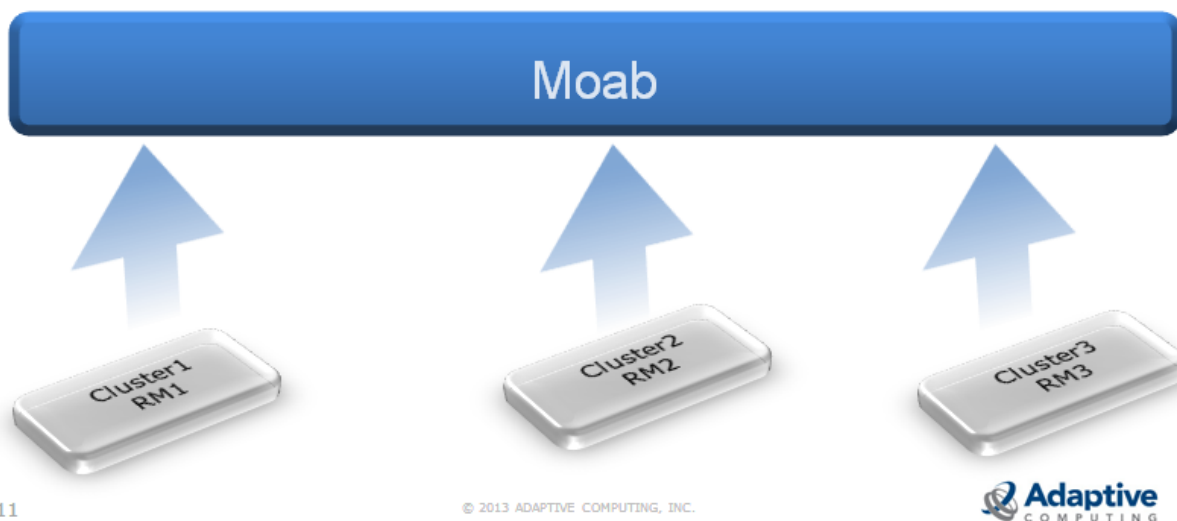
Source Destination Topology



- **Master-Slave.**

Есть один ЦОД, управляемый единственным экземпляром Moab. Однако, под управлением Moab находится множество разнородных ресурсов – настолько разнородных, что пул каждого ресурса можно считать отдельной площадкой. Так, примером Master-Slave, скорее всего, будет ЦОД, в котором Moab управляет пулом серверов x86, IBM Power, IBM System z, HP Integrity Superdome, SGI Altix, Cray и Oracle. Несмотря на то, что Moab в принципе способен переносить сервисы между разными пулами ресурсов, в данном случае перенос сервиса между пулами платформ разного типа едва ли имеет смысл, поскольку каждому виду платформ, скорее всего, соответствует своя специфическая нагрузка. В этом и заключается смысл топологии Master-Slave: ровно один экземпляр Moab управляет множеством логических площадок, каждая из которых образована пулом специфического ресурса.

Master-Slave



8. Независимость от нижележащей платформы

Поскольку Moab максимально отдален от управляемых им ресурсов через прослойку ресурсных менеджеров, то, как следствие, Moab предельно толерантен к аппаратным платформам и операционным системам на управляемых ресурсах. Несмотря на то, что сама по себе серверная часть Moab работает только на x86_64 (если нужно, в режиме high availability), управляемые ресурсы могут реализовываться на самых разных платформах с самыми разными операционными системами, от простейших серверов x86 до узкоспециализированных платформ для бизнес-аналитики, транзакционной нагрузки, обработки массивов неструктурированных данных, и даже высокопроизводительных вычислений. Нужно лишь, чтобы Moab, при посредничестве ресурсного менеджера, имел удаленный доступ на соответствующие системы, с тем, чтобы опросить значение управляемого ресурса.

9. Планирование IT-бюджета в Moab

Поскольку выше сказано, что все алгоритмы планирования, доступные в Moab, одинаково применимы к любому ресурсу, подключенному к Moab, рассмотрим один важный частный случай в качестве примера.

С одной стороны, «движок» Moab реализует такой алгоритм планирования ресурсов, как fair share tree. Это алгоритм планирования, при котором администратор ЦОДа сообщает «движку» Moab процентные доли ресурса, выделенные разным департаментам в организации, и далее Moab выделяет ресурсы с учетом заданного распределения ресурсов по долям между пользователями или департаментами.

С другой стороны, одним из управляемых ресурсов Moab может являться IT-бюджет, выделенный на ЦОД.

Следовательно, администратор ЦОДа может применить алгоритм fair share tree для гарантированного распределения IT-бюджета, выделенного в организации для коммерческой (внешней или внутренней) эксплуатации ЦОДа, в процентном соотношении между подразделениями.

На практике задание fair share tree для распределения бюджета ЦОДа может выглядеть следующим образом:

1. Вся организация владеет 100% годового бюджета эксплуатации ЦОДа.
2. Департаменту геологоразведки выделена гарантированная доля в 50%.
3. Департаменту разработки ПО для внутренних нужд выделено 20%.
4. IT-отделу для хостинга базовых сервисов выделено 10%.
5. Учебному центру выделено 10%.
6. Вспомогательным службам выделено 10%.

После того, как администратор ЦОДа фиксирует эти доли распределения бюджета в Moab, «движок» Moab начинает выделять ресурсы ЦОДа с учетом этих долей. В зависимости от конкретной конфигурации, Moab может применять тонкие настройки, отвечающие на вопросы, неизбежно возникающие при распределении долей бюджета:

- Можно ли отдельным пользователям или департаментам превышать свою долю?
- При каких обстоятельствах отдельным пользователям или департаментам можно или необходимо урезать их долю?
- Допустимо ли применять взаиморасчет или взаимозачет? Если, к примеру, отдел геологоразведки в течение 2 месяцев превышал свою долю за счет доли учебного центра, то до конца текущего финансового года Moab может компенсировать убыток учебного центра, выдав ему недостающую долю ресурса за счет отдела геологоразведки, либо кредитовав учебный центр на следующий учетный период за счет отдела геологоразведки.

Также возможны и другие детали применения алгоритма планирования fair share tree.

Для наиболее удобного использования бюджета как ресурса, Moab Cloud Suite и Moab HPC Suite имеют встроенное средство финансовой отчетности.

10. Подключение «вендорских» средств в Moab в качестве ресурсов

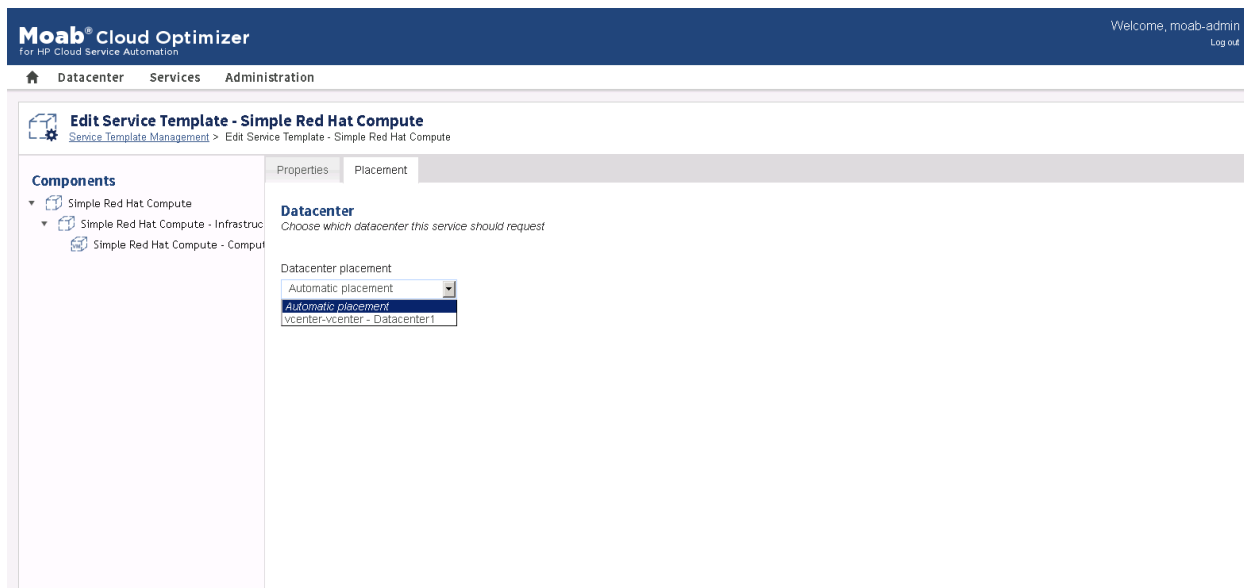
Аналогично примерам выше, Moab может интегрировать в себя «вендорские» средства управления ЦОДом, и использовать их в качестве ресурса. Приведем примеры ресурсных менеджеров, заранее подготовленных для наиболее популярных видов «вендорских» средств:

- VMware vCenter Connector.
- HP Data Center Automation (DCA) Connector.
- HP Cloud Service Automation (CSA) Connector.
- HP Cluster Management Utility (CMU) Connector.
- xCAT Connector.

Дополнительными примерами популярных инструментов на основе open source, широко используемых в связке с Moab, являются Nagios и Ganglia.

Таким образом, Moab не вытесняет «вендорские» средства управления и мониторинга аппаратных платформ, а интегрирует их в себя. Это крайне важно на практике, поскольку во многих сценариях развития ЦОДа перестройка связующего ПО в ЦОДе «с чистого листа» недопустима в силу необходимости непрерывного предоставления сервисов, либо не является экономически оправданной.

Ниже приведен вид интегрированного с Moab средства HP CSA в момент задания политики выбора ЦОДа, в котором по умолчанию должны выделяться виртуальные машины:



11. Экономический обзор применения решений на базе Moab

В целом, настоящий документ описывает конкурентные преимущества продуктов на базе Moab с акцентом на архитектурно-технические особенности. Однако, для более полного понимания механизма получения экономической выгоды из архитектурных преимуществ решений на базе Moab, в текущем разделе кратко приведем экономические выгоды в терминах сокращения расходов и возврата инвестиций.

Снижение капитальных расходов (CapEx) достигается по следующим направлениям:

1. Уменьшение первоначального бюджета на вычислительную инфраструктуру ЦОДа. Это свойство продуктов Adaptive Computing иногда называют принципом “Do More With Less”. На практике оно означает уменьшение расчетного количества серверов, которое требуется закупить для обеспечения заданного уровня сервиса в ЦОДе. Требуемое количество серверов снижается за счет повышения эффективности использования серверов. Именно, применение продуктов Adaptive Computing позволяет повысить устоявшийся уровень загрузки ресурсов (sustained utilization) с 50-60% до 95% и выше, нередко до устоявшегося уровня, близкого к 100%.
2. Консолидация IT-инфраструктуры в ЦОДе с возможностью удаленного доступа.
3. Уменьшение периодических бюджетов на расширение ЦОДа, которое достигается тем же способом: снижение числа серверов, требуемых для предоставления заданного уровня сервиса.

Снижение операционных расходов (OpEx) достигается по следующим направлениям.

1. Снижение энергопотребления ЦОДа. Этот эффект достигается за счет планирования ресурсов с учетом энергопотребления и управления электропитанием аппаратных платформ в зависимости от текущей пользовательской нагрузки в ЦОДе.
2. Уменьшение периодических бюджетов на замену аппаратных платформ и их компонент. Соответствующий эффект достигается за счет планирования ресурсов с учетом параметров наработки на отказ (MTBF). Подробности этого аспекта планирования ресурсов приведены в соответствующем разделе настоящего документа.
3. Уменьшение административного ресурса ЦОДа благодаря автоматизации политик предоставления сервисов.

Кроме того, часть оптимизации выражается не в уменьшении инвестиций, а в увеличении результата, получаемого в рамках основного бизнеса за заданный отчетный период, то есть в данном случае речь идет об оптимизации ROI. К такому виду оптимизации, реализуемой благодаря продуктам Adaptive Computing, относятся

1. Уменьшение затрат на дорогостоящее коммерческое ПО, используемое основными линиями бизнеса. Продукты на базе Moab позволяют достичь этого эффекта за счет оптимизации планирования ресурсов, в том числе лицензий на коммерческое ПО, выделяемых для выполнения заданных SLA. Как следствие, в течение заданного отчетного периода увеличивается коэффициент использования коммерческих лицензий, что в конечном итоге приводит к увеличению результатов основных линий бизнеса за тот же период.
2. Снижение порога вхождения пользователей за счет устранения необходимости в специальных знаниях, специфичных для использования ресурсов ЦОДа. Этот эффект достигается за счет описанной ниже архитектуры продуктов Adaptive Computing: планирование ресурсов в терминах SLA (Service Level Agreement), близких к терминам основных линий бизнеса, в противоположность традиционному планированию в терминах низкоуровневой приоритезации ресурсов. В результате сокращается время выхода основных линий бизнеса, связанных с сервисами ЦОДа, в режим полной продуктивности.

12. Примеры клиентских решений на основе Moab

Решение Moab Cloud Suite управляет частным облаком, являющимся одним из крупнейших в мире. Данная инфраструктура развернута в одном из самых крупных банков США. Масштаб инфраструктуры под управлением Moab составляет 10 географически разнесенных ЦОДов, 10.000 серверов, 100.000 сервисов, и такого же порядка число виртуальных машин. Сценарии использования этого облака включают agile service delivery, automated management, adaptive services across physical and virtual heterogeneous environment.

Результатом внедрения Moab Cloud Suite в данной организации явилась экономия 1 миллиарда долларов США в течение первых 3 лет эксплуатации (CapEx и OpEx в сумме), а также сопутствующее ускорение предоставления сервисов по запросу, повышение пропускной способности всей инфраструктуры ЦОДов в терминах устоявшегося максимального числа сервисов, предоставляемого за 1 рабочий день.

Другим важным примером оптимизации экономии CapEx и OpEx в клиентской организации может выступать одна из крупнейших нефтяных компаний США, использующая решение Moab HPC Suite в своих ЦОДах.

В качестве дополнительного примера горизонтальной расширяемости решений на базе Moab может выступать высокопроизводительная система Titan, построенная на аппаратных платформах производства Cray, Inc. Эта система, установленная в национальной лаборатории Оук-Ридж Министерства энергетики США, занимает 2-е место в списке 500 мощнейших суперкомпьютерных систем в мире (по состоянию на конец 2013-го года).



В системе Titan Moab управляет 18688 процессорами AMD Opteron 6274, а также 18688 графическими ускорителями NVidia Tesla K20x, расположенными на 404 кв.м с общим энергопотреблением 8,2 МВт.