

## Пути к разработке приложений для облачных вычислений с учетом энергии (Часть 3)

Момбекова.С.С., Рахымбек Н.Ж., Шаймерденова Г.С., Колбоев Б.Р.,  
Бибулова Д.А.  
*ЮКГУ им.М.О.Ауезова*

Исследовательская работа нацелена на поддержку энергоэффективности на разных этапах жизненного цикла облачных сервисов. На стадии разработки услуг требования к выявлению требований включают в себя методы захвата, моделирования и анализа с учетом требований к энергии, а также методы, ориентированные на линейку продуктов, для моделирования и обоснования конфигурации системы. Что касается разработки программного обеспечения в отношении потребления энергии, то некоторые исследовательские усилия связаны с осознанием и оптимизацией энергии на уровне приложений и систем, основное внимание уделяется профилированию потребления энергии приложения во время выполнения, итеративно узкой по энергетическим горячим точкам, или рассматривает шаблоны облачной архитектуры для достижения более зрелых бизнес-процессов. На этапе развертывания услуг исследовательские усилия были сосредоточены на стратегиях развертывания соглашений об уровне обслуживания (SLA), особенно в отношении SLA, которые осведомлены об энергопотреблении. путем реализации конкретных политик для экономии энергии, а также технологий развертывания служб, которые играют решающую роль в управлении облачной инфраструктурой и, таким образом, влияют на ее общее потребление энергии. На этапе эксплуатации эффективность энергоэффективности широко изучалась и, например, была сфокусирована на подходах к управлению энергопотреблением для распределенного управления виртуальными машинами в облачных инфраструктурах, целью которых является улучшение использования вычислительных ресурсов и снижение потребления энергии при независимом от нагрузки уровне ограничений обслуживания.

Инструменты управления конфигурацией обеспечивают четыре основных преимущества для управления облаком. Это 1) воспроизводимость и индустриализация конфигурации программного обеспечения, 2) непрерывная бдительность по работающим системам с автоматическим ремонтом и механизмами оповещений, 3) усиление контроля и рационализации крупномасштабных развертываний и 4) способность создавать базу знаний для документировать и отслеживать историю системы по мере ее развития. Наиболее хорошо известные инструменты включают CFEngine, Puppet и Chef.

CFEngine обеспечивает автоматическое управление конфигурацией для крупных сетевых систем и может быть применено для управления различными инфраструктурами, такими как серверы, настольные компьютеры и мобильные / встроенные устройства. Он использует децентрализованные автономные программные агенты для мониторинга, ремонта и обновления отдельных машин. Центральная концепция CFEngine - это идея конвергенции, где описывается конечное желаемое состояние системы, а не шаги, необходимые для ее получения. Это позволяет CFEngine запускать любое начальное состояние системы с предсказуемыми конечными результатами. Недостатком этого подхода является то, что только с соблюдением статистики или наилучшими усилиями может быть достигнута заданная политика конфигурации, когда система не может быть гарантирована в конечном итоге в желаемом состоянии, но медленно сходится со скоростью, определяемой отношением изменения окружающей среды к скорости, с которой выполняется CFEngine. Кукольный был раздвоен из CFEngine и предоставляет графические и модельные подходы к управлению конфигурацией с помощью упрощенного декларативного языка, специфичного для домена, который был разработан для чтения человеком.

Открытый формат виртуализации (OVF) является открытым стандартом для определения, упаковки и распространения виртуальных устройств, которые могут работать с виртуализацией в облаке. Его использование в составе дескриптора службы для определения требований приложений не является новым и было реализовано в OPTIMIS Toolkit, где OVF-фрагмент находится в нестандартной схеме манифеста службы на основе XML. Одной из проблем с этим подходом является влияние на интероперабельность с облачными провайдерами, которые должны поддерживать эту схему, чтобы обеспечить развертывание приложений. Это по сравнению с решением, которое представлено в этой статье, где используется чистый документ OVF, расширенный и реализованный в соответствии с возможностями спецификации 1.1.1 OVF Specification, делает наше решение на 100% совместимым с облачными провайдерами и технологиями, которые уже поддерживают OVF.

1. Pawlish M, Varde AS, Robila SA (2012) Cloud Computing для экологически чистых центров обработки данных В: Материалы четвертого международного семинара по управлению облачными данными, CloudDB &acute;12, 43-48 .. ACM, Нью-Йорк