УДК 004.4

Использование, возможности и задачи технологии SDI

И.В. Филиппов[1,2], А.Ф. Мелик-Адамян[1]

[1]АО «Интел А/О»

[2]Московский физико-технический институт (государственный университет)

Современные массивно-параллельные системы, как облачные, так и вычислительные, достаточно широко используют технологии автоматического программного управления вычислительными ресурсами. К таким решениям можно отнести, например, программно-определяемые сети (SDN). SDN предполагает явное разделение пересылающей аппаратуры (data plain) и управляющей аппаратуры (control plane). В любой момент времени программа деятельности любого пересылающего устройства может быть изменена из центра управления для лучшей адаптации к текущей задаче. Другой пример управляемой аппаратуры – программно-управляемое хранилище данных (SDS). SDS регулирует возможности хранилища и предоставляемые им сервисы, подстраивая их под конкретную задачу. Кроме SDN и SDS, существует технология программно-управляемого дата-центра (SDDC). Под SDDC понимается дата-центр, вся аппаратура которого виртуализована и предоставляется как различные сервисы, контроль за которыми полностью берёт на себя управляющая система, не зависящая от конкретной архитектуры.

Сейчас данные технологии объединяются под названием программно-конфигурируемая инфраструктура (SDI), в которой декларируется возможность программно управлять любой частью системы от памяти до сети, вне зависимости от конкретной инфраструктуры её компонентов. Существуют три основные парадигмы облачных вычислений: архитектура как сервис (IaaS), платформа как сервис (PaaS) и приложение как сервис (SaaS). В докладе будут рассмотрены основные возможности технологии SDI в рамках парадигмы PaaS, как предоставляющей наибольшие возможности для динамического конфигурирования.

Приложения, использующие кластеры, сильно различаются по типу: высокопроизводительные расчёты, клиентоориентированные сервисы, распределённые хранилища данных и так далее. Для каждого типа задач необходимы различные стеки окружения (среды), в которых они выполняются[1]. Некоторые уровни стеков совпадают (например, операционная система), некоторые сильно различаются, например, в анализе больших данных часто используются Hadoop и MapReduce, а в научных расчётах различные технологии ускорения, например, OpenMP и Intel MPI. Нужно отметить, что на данный

момент все стеки не реконфигурируемы, что означает, что облачная система не может эффективно выполнять вычислительные задачи, и наоборот. Возникает задача динамической адаптации сервера под стек окружения конкретной задачи.

Более того, пользователь может использовать нестандартное окружение, например, конкретные версии библиотек, или Java машину, или запущенное стороннее приложение. Для таких задач используются технологии контейнеризации, например Docker [2]. Однако для создания контейнера необходимо сконфигурировать все необходимые части вручную. Практической задачей становится возможность задать требуемое окружение, которое автоматически разворачивалось бы на выделенном для запуска приложения сервере или группе серверов. В качестве продолжения этой задачи можно назвать возможность автоматически конфигурировать окружение в зависимости от сценария работы приложения.

Системы управления контейнерами, например, Kubernetes, не учитывают особенности гетерогенности на уровне аппаратуры. Сервера могут содержать CPU, GPU, FPGA, и другие специальные ускорители. В гетерогенной системе возникает задача исполнения пользовательского приложения на серверах с соответствующими ресурсами. Данная задача является продолжением технологии offload.

Таким образом, SDI предоставляет кластер, который может состоять из десятков тысяч серверов, динамически конфигуриремый под необходимое каждой задаче окружение, динамически адаптируемый для максимальной производительности каждой задачи с учётом гетерогенности ядер. В этих условиях важнейшей задачей становится задача планирования исполнения.

В результате данного исследования были выявлены высокоуровневые задачи для дальнейшей работы на основе системы управления контейнерами Kubernetes. Было решено расширить ее новыми возможности, позволяющими выбирать необходимое аппаратное обеспечение и устанавливать необходимое программное обеспечение с учётом требований каждого приложения (self-provisioning). Добавление этих возможностей подразумевает значительные изменения в системе планирования выполнения (scheduling), оркестрирования сервисов (orchestrating) и изменение взаимодействия с соседними системами в программном стеке облачной инфраструктуры.

Литература

1. *Reed, D., & Dongarra, J.* Exascale computing and big data. – Communications of the ACM – 2015 – pp. 56-68.
2. *Anderson, Charles* Docker [Software engineering] – Software, IEEE vol.32, no.3 – May-June 2015 – pp.102-c3.