

О ПРИМЕНЕНИИ ИИ В ВЫЧИСЛИТЕЛЬНОЙ ИНФРАСТРУКТУРЕ НОВОГО ПОКОЛЕНИЯ

Р.Л. Смелянский

МГУ имени М.В. Ломоносова
Россия, 119991, Москва, Ленинские горы, д. 1, стр. 52
E-mail: smel@cs.msu.su

В.Ю. Королев

МГУ имени М.В. Ломоносова
Россия, 119991, Москва, Ленинские горы, д. 1, стр. 52
E-mail: bruce27@yandex.ru

Е.П. Степанов

МГУ имени М.В. Ломоносова
Россия, 119991, Москва, Ленинские горы, д. 1, стр. 52
E-mail: estepanov@lvk.cs.msu.ru

Ключевые слова: сетевая вычислительная инфраструктура, методы машинного обучения, мультиагентные методы.

Аннотация: На основе анализа свойств современных приложений (прикладных программных систем) показано, что существующая вычислительная инфраструктура им не удовлетворяет. В докладе представлена архитектура сетевой вычислительной инфраструктуры нового поколения и на конкретных примерах показано, как средствами управления ее ресурсами на основе методов машинного обучения можно удовлетворить требования современных приложений.

1. Введение

Технология вычислений – одна из основ современной цивилизации. Поэтому важно понимать основные тенденции и направление ее развития. В настоящее время вычислительная инфраструктура переходит от концепции «создай сам» к концепции — «потребляй как услуга», когда не нужно покупать и развивать собственную вычислительную инфраструктуру, арендовать каналы для подключения с общедоступной сетью, нанимать дорогих специалистов для системного и сетевого администрирования. В новой парадигме можно запрашивать и получать ресурсы и услуги по мере необходимости. Однако такая возможность сегодня ограничена рамками конкретного центра обработки данных или высокопроизводительной вычислительной установки.

Если мы посмотрим на историю развития вычислительной инфраструктуры, то главный урок этой истории заключается в том, что основным двигателем этого прогресса были требования приложений. Именно они стимулировали три основные движущие силы развития вычислительной инфраструктуры: микроэлектронику, инженерию программного обеспечения и телекоммуникацию.

В 2018 году лауреаты премии Тьюринга – профессора Массачусетского технологического института Дж. Л. Хеннесси и Д.А. Паттерсон в своей лекции [1] дали прекрасный обзор истории компьютерной архитектуры и уроков этой истории. В заключение этой статьи они написали: “The next decade will see a Cambrian explosion of novel computer architectures, meaning exciting times for computer architects in academia and in industry”. Этот «взрыв» открывает большие возможности для вычислительной инфраструктуры.

Появление новых технологий виртуализации вычислительных ресурсов, программно-конфигурируемых сетей и виртуализации сетевых функций открыло новые возможности для соединения ресурсов отдельных центров обработки данных (ЦОД). Теперь не столь важно, сколько серверов и стоек в ЦОД, а то, насколько эффективно мы умеем сопоставить виртуальные ресурсы физическим.

Для определения требований к вычислительной инфраструктуре в работе [2] были проанализированы свойства современных приложений. Основные из них: распределенность, самодостаточность, режим реального времени, эластичность, кроссплатформенность и другие. Определения этих терминов можно найти в [2]. Тем не менее, для лучшего понимания дальнейшего текста необходимо пояснить термин самодостаточность, который определен в [2] как: «приложение уже не представляется только кодом и исходными данными, оно сопровождается описанием структуры взаимосвязи компонентов (далее сервисов приложений), составляющих приложение; заданием требуемого уровня их производительности, явно сформулированными требованиями к вычислительным и сетевым ресурсам, хранению данных и доступу к ресурсам данных, предполагаемым временным ограничениям на время вычислений и передачи данных в виде соглашения об уровне обслуживания (SLA), процедуры запуска приложений. Для таких описаний сейчас активно развиваются специальные языки. Ярким примером прообраза такого языка может служить язык TOSCA [3] (далее такое описание будет называться Application Operation Specification — AOS)».

На основе анализа свойств приложений были сформулированы требования к перспективной вычислительной инфраструктуре. Вот основные из них [2]:

- Повсеместно связанные и доступные вычислительные ресурсы, не ограниченные рамками локальных ЦОДов или вычислительных комплексов;
- Детерминированное качество сервисов передачи данных;
- Наличие оперативной информации о доступных вычислительных мощностях и ресурсах сети передачи данных;
- Виртуализация, масштабируемость, бессерверность;
- Доступность, надежность и отказоустойчивость;
- Эффективность и справедливость при распределении ресурсов;
- Безопасность.

Вычислительную инфраструктуру с указанными выше свойствами мы будем называть Сетевой Вычислительной Средой или по-английски – Network Powered by Computing (NPC). NPC — это открытая, программно-управляемая, полностью виртуализированная инфраструктура, представляющая собой тесную программно-ориентированную интеграцию различных вычислительных установок и высокоскоростной сети передачи данных (DTN). Важно подчеркнуть, что эта инфраструктура не ограничена рамками ЦОДа, локального комплекса вычислительных установок, сколь мощными они не были. Здесь можно провести аналогию с инфраструктурой генерации и управления распределением электрической энергии в любом современном экономически развитом государстве.

Состав функциональной архитектуры NPC кратко может быть описан следующим образом:

- совокупность вычислительных установок (ЦОДов, периферийных вычислительных ресурсов (Edge), НРС установок), состав которой может динамично изменяться;
- SD-WAN наложенная сеть передачи данных, соединяющая между собой вычислительные ресурсы НРС;
- контур управления приложениями и их данными, вычислительными ресурсами (распределением ресурсов в соответствии с SLA);
- контур управления SD-WAN сетью передачи данных, которая представляет собой наложенную сеть на физической сети оператора.

Технологическую основу НРС составляют технологии программно-конфигурируемого управления и виртуализации сетевых функций [4].

Современные скорости передачи данных, вычислений, масштабы потоков приложений и запросов к ним с учетом ограничений на время их выполнения/реакции на запрос, не позволяют применять классические методы оптимизации в силу их высокой вычислительной сложности. Решение приходится искать в области методов много агентной оптимизации (МА оптимизации), методов управления с машинным обучением. Этот тезис демонстрируется на решении трех задач, без решения которых невозможна НРС инфраструктура:

- оценка ожидаемого времени выполнения приложения на определенном вычислителе НРС;
- оптимального распределения/балансировки потоков данных в транспортной среде НРС;
- выбор оптимального транспортного/наложенного канала для передачи данных.

2. Оценка времени выполнения программы

Задача оценки времени выполнения приложения, применительно к MPI программ для НРС вычислителей рассмотрена в статье [2]. Такая оценка необходима для эффективного распределения потока приложений между неоднородными вычислителями НРС. Задача предсказания времени представлена как задача рекомендательной системы, когда известен набор MPI программ, представленных набором определенных параметров, и набор вычислителей, на некоторых из которых выполнялось подмножество программ из этого набора. Задача предсказания состоит в том, чтобы спрогнозировать ожидаемое время работы программы на вычислителе, на котором она до этого не выполнялась.

Задача оценки времени выполнения программы на определенном вычислителе хорошо известна и является классической. Например, время выполнения программы на конкретном вычислителе можно спрогнозировать на основе истории ее запусков на этом вычислителе. Для этого могут быть использованы многие алгоритмы экстраполяции, регрессии или их комбинации в виде ансамбля методов. Основным недостатком всех этих алгоритмов является то, что они применимы только к одному и тому же вычислителю. Но особенность рассматриваемой задачи заключается в том, чтобы получить оценку времени выполнения приложения на определенном наборе неоднородных вычислителей, когда история запуска каждой программы на каждом вычислителе из этого набора не доступна.

3. Балансировка потоков данных в НРС

Задача управления потоками данных (далее сетевым трафиком) в НРС, рассмотрена как задача балансировки трафика по каналам сети передачи данных так, чтобы не

вызывать перегрузки и минимизировать задержки при передаче. Предложенное решение [5] основано на комбинации методов много агентной оптимизации и обучением с подкреплением. Существует три подхода к реализации много агентной оптимизации: централизованный, децентрализованный с взаимодействием между агентами и полностью децентрализованный. Во всех этих подходах предполагается, что агенты формируют свое локальное состояние на основе наблюдений за окружающей средой. Однако поведение агентов в этих подходах различно:

- централизованный подход предполагает наличие в среде одного объекта (центр управления), который координирует работу каждого агента. Этот центр управления собирает данные о локальных состояниях от всех агентов и определяет действия для каждого из них на основе решения задачи оптимизации.
- децентрализованный подход с взаимодействием между агентами (или просто децентрализованный подход) предполагает, что каждый агент обменивается с другими агентами своим локальным состоянием. На основе собранной информации агент выбирает оптимальное действие.
- полностью децентрализованный подход не предполагает какого-либо общения между агентами. Каждый агент принимает решение только на основе анализа истории своего поведения.

В докладе рассмотрен метод Multi-Agent Routing using Hashing (MARON), предложенный в [5]. Метод MARON основан на объединении идей децентрализованного мультиагентного обучения с подкреплением и алгоритмов консистентного хеширования. Эта комбинация позволила обеспечить справедливое распределение трафика между сетевыми ресурсами и эффективное их использование. В среде NPC предполагается наличие специального программно-аппаратного устройства NPC Router (NPCR) – это шлюз наложенной сети передачи данных в NPC, обеспечивающий доступ к локальному вычислительному ресурсу, например, ЦОД. Предложенный метод в каждый момент времени каждому выходному порту NPCR присваивает вес, который потом используется консистентной хэш-функцией для распределения потоков данных между наложенными каналами. Таким образом, ключевыми проблемами являются то, как присвоить веса выходным портам NPCR и как построить надлежащую хеш-функцию.

Чтобы избежать циклов на маршрутах, для выбора порта до надлежащего места назначения в методе MARON был разработан специальный алгоритм построения маршрутов, называемый выбором следующего перехода (NHS) [5]. Согласно этому алгоритму, строится два ациклических направленных остовных подграфа топологии NPC и лишь один раз можно перейти между этими подграфами при построении маршрута, что гарантирует отсутствие циклов.

4. Выбор оптимального наложенного канала

Задача выбора оптимального канала рассматривается в следующей постановке. Даны две вычислительные установки, расположенные за разными NPCR, которые соединены разными по качеству наложенными каналами. Под качеством наложенного канала (далее просто канала) понимается набор величин, определяющих вариацию доступной пропускной способности канала, вариации джиттера и задержки пакета, вероятность потери пакета в канале и стоимость канала за единицу времени. Каждый поток данных, поступающий на NPCR со стороны вычислителя, характеризуется своими требованиями к качеству канала (SLA). SLA представлен четырьмя величинами: минимально допустимой скоростью, максимально допустимой вероятностью потери пакета, максимально допустимыми джиттером и задержкой в

канале. Требуется выбрать канал, который будет минимальным по стоимости и удовлетворять ограничениям SLA. Для выбранного канала также требуется определить параметры транспортного соединения, устанавливаемого между NPCR.

Эта задача разбивается на три:

- переход от Backward Error Correction (повторные передачи пакетов в TCP в случае возникновения потери) к Forward Error Correction за счет передачи пакетов блоками и введение избыточности в каждый блок;
- выбор оптимальных параметров в алгоритме управления перегрузкой (в качестве такового был рассмотрен BBR [6]);
- оценка ожидаемой пропускной способности в канале с заданными показателями качества, при соблюдении ограничений, накладываемых SLA потока.

Вторая и третья задачи решаются с применением методов регрессионного анализа и построения ансамбля алгоритмов. На экспериментальных данных показано, что предложенное решение обеспечивает не менее чем в 90% случаев выбор оптимального канала, соответствующего требованиям SLA потока. Представлены результаты сравнительного анализа предложенного метода с другими методами выбора канала, а именно: без использования техники FEC, со случайным выбором канала, выбором канала с минимальным RTT, минимальной вероятностью потери пакета и минимальным значением метрики vQoE, предложенной компанией Cisco Systems.

Также в докладе представлены экспериментальные оценки потребляемых ресурсов.

5. Заключение

На основе анализа свойств современных приложений сформулированы требования к вычислительной инфраструктуре нового поколения такие, как:

- повсеместно связанные и доступные вычислительные ресурсы, детерминированное качество сервисов передачи данных;
- наличие оперативной информации о доступных вычислительных мощностях и ресурсах сети передачи данных;
- виртуализация, масштабируемость, бессерверность;
- доступность, надежность и отказоустойчивость;
- эффективность и справедливость при распределении ресурсов;
- безопасность.

На примере задач оценки времени выполнения программ, балансировки потоков данных и выбора оптимального наложенного канала было показано, как с помощью методов искусственного интеллекта можно обеспечить управления ресурсами предложенной вычислительной инфраструктуры нового поколения так, чтобы удовлетворить эти требования.

Работа выполнена при поддержке Программы развития МГУ, проект №23-Ш03-03.

Список литературы

1. John L. Hennessy, David A. Patterson Communications of the ACM A New Golden Age for Computer Architecture // Communication of the ACM. 2019. Vol. 62, No. 2. P. 48-60.
2. Smeliansky R. Network Powered by Computing: Next Generation of Computational Infrastructure. In "Edge Computing – Technology, Management and Integration" // IntechOpen. 2023, DOI: 10.5772/intechopen.110178.
3. Topology and Orchestration Specification for Cloud Applications. [Internet] Available from: <http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html> [Дата обращения: 2023-14-11].
4. Смелянский Р.Л. Антоненко В.А. Концепции программного управления и виртуализации сетевых сервисов в современных сетях передачи данных. М.: Курс, 2020.

5. Stepanov E. et al. On Fair Traffic allocation and Efficient Utilization of Network Resources based on MARL // Preliminary on ResearchGate, https://www.researchgate.net/publication/371166584_On_Fair_Traffic_allocation_and_Efficient_Utilization_of_Network_Resources_based_on_MARL [Дата обращения 2023-14-11].
6. BBR Congestion Control. <https://www.ietf.org/archive/id/draft-cardwell-iccrp-bbr-congestion-control-01.html> [Дата обращения 2023-11-19].