

Магистрант Я.Ю. Навроцкий
Науч. рук. доц. Н.В. Пацей
(кафедра программной инженерии, БГТУ)

СРЕДА МОДЕЛИРОВАНИЯ И ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ КЭШИРОВАНИЯ В ICN

Информационно-ориентированная сеть (ICN) – это новая сетевая архитектура, разработанная для замены текущей, широко используемой хост-ориентированной архитектуры, в которой конечная цель данные, а не узел, предоставляющий доступ к ним. Подходы к информационно-ориентированным сетям отличаются с точки зрения реализации, но они имеют одинаковую цель – улучшить производительность и обеспечить более удобный доступ к данным, предоставляя доступ к контенту по имени, а не по адресу. Для того чтобы увеличить производительность в ICN используется внутрисетевое хранилище данных, эффективность которого зависит от политики кэширования и стратегии замены данных в кэше.

Кэш представляет собой аппаратный или программный компонент способный хранить данные, которые могут быть запрошены с большей вероятностью. Доступ к данным в кэше осуществляется быстрее, чем выборка исходных данных из более медленной памяти или удаленного источника, однако объем кэша значительно ограничен по сравнению с хранилищем исходных данных. Данные, хранящиеся в кэше, могут быть результатом более раннего вычисления или дублирования данных, хранящихся в другом месте. Чем больше данных можно забрать из кэша, тем быстрее работает система. Обычно размер кэша относительно невелик, чтобы быть экономически выгодным и обеспечить эффективное использование данных.

Кэш состоит из набора записей. Каждая запись хранит данные, которые являются копией данных из удаленного хранилища. Так же, каждая запись имеет тег, который указывает на оригинал данных из хранилища. Когда клиенту требуется данные, система обработки запроса сначала проверяет кэш. Если в кэше найдена запись с тегом запрашиваемых данных, то система вернет данные из кэша. Ситуация, когда запрашиваемые данные найдены в кэше, называется попаданием кэша. Обратная ситуация, когда происходит обращение к кэшу и кэш не содержит запрашиваемых данных, называется промахом кэша. В случае промаха в кэше данные полученные из удаленного хранилища записываются в кэш, и будут возвращены при следующем запросе. При

поступлении данных в кэш система обычно удаляет определенные записи, чтобы освободить место для новых данных. Эвристика, используемая для выбора удаляемой записи, называется политикой замещения. Для эффективности кэша используют политики замещения в зависимости от размера хранимого содержимого, а также задержек и пропускной способности как для кэша, так и для удаленного хранилища.

Эффективность политики кэширования зависит от того как часто данные берутся из кэша, чем больше данных можно вернуть из кэша, тем эффективнее политика кэширования. Эффективность политики кэширования определяется по формуле:

$$CacheHit = \frac{\sum_{i=1}^N hits_i}{(\sum_{i=1}^N hits_i) + (\sum_{i=1}^N miss_i)};$$

где N – это количество узлов в сети, $hits_i$ – количество попаданий в кэш в узле i , $miss_i$ – это количество промахов в кэше в узле i .

Протяженность – это характеристика, показывающая процент пройденного пути для получения данных, чем меньше пройденный путь, тем эффективнее политика кэширования. Протяженность определяется по формуле:

$$Stretch = \frac{\sum_{i=1}^R hops_walked_i}{\sum_{i=1}^R total_hops_i};$$

где R – это количество запросов, $hops_walked_i$ – количество хопов от клиента до узла предоставляющего данные, $total_hops_i$ – количество хопов от клиента до узла с оригинальными данными.

Так же при анализе характеристик политик кэширования должны учитываться такие параметры как топология, количество запросов, размер каталога, размер кэша.

Для анализа характеристик политик кэширования была написана библиотека классов для среды Unity, позволяющая создать модель контент-ориентированной сети с различными параметрами.

Для возможности переключения между различными политиками кэширования и стратегиями замещения данных в кэше использовался принцип инверсии зависимостей, так идет привязка не к конкретной реализации, а к абстрактной сущности, например, интерфейс.

Информация о том, как проходит процесс оценки политики кэширования, выводится в окне инспектора.

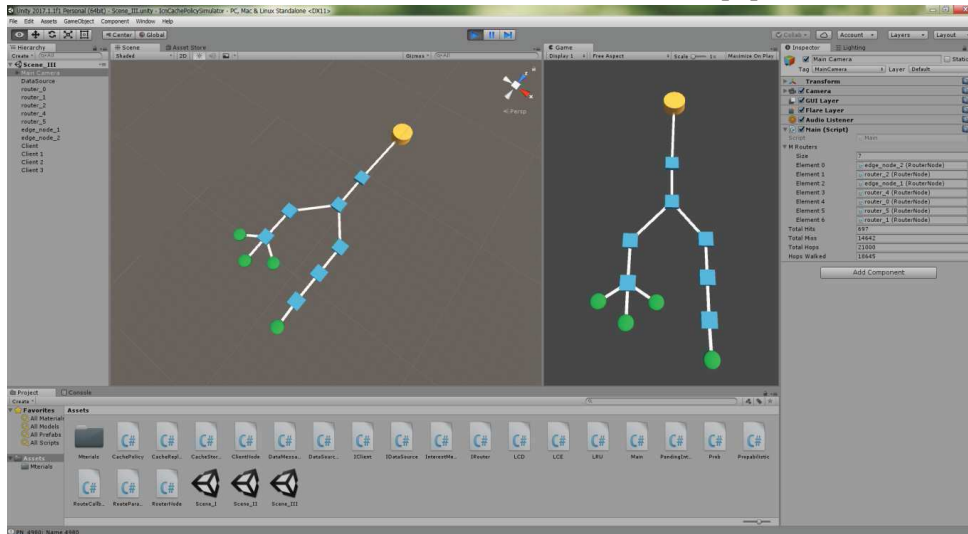


Рисунок 1 – Собранная модель контент ориентированной сети

Main (Script)	
Script	Main
M Routers	
Size	7
Element 0	edge_node_2 (RouterNode)
Element 1	router_2 (RouterNode)
Element 2	edge_node_1 (RouterNode)
Element 3	router_4 (RouterNode)
Element 4	router_0 (RouterNode)
Element 5	router_5 (RouterNode)
Element 6	router_1 (RouterNode)
Total Hits	3
Total Miss	8821
Total Hops	10880
Hops Walked	10877

Рисунок 2 – Результаты процесса оценки политики кэширования

В результате получилось создать однородную среду для оценки политик кэширования в CCN. Данная среда позволяет задавать размер каталога, размер кэша, количество запросов от пользователя, менять политику кэширования и стратегию замены данных в кэше, а так же создать собственную модель сети.

ЛИТЕРАТУРА

1. G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A Survey of Information-Centric Networking Research," IEEE Communications Surveys & Tutorials, vol. 16.

2. I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," ACM SIGCOMM Workshop ICN 2012.

3. G. Zhang, Y. Li, and T. Lin, “Caching in information centric networking: A survey,” *Computer Networks*, vol. 57, no. 16, pp. 3128 – 3141, 2013.
4. G. Rossini and D. Rossi, “A dive into the caching performance of content centric networking.” in *CAMAD*. IEEE, 2012, pp. 105–109.
5. I. Abdullahi, S. Arif, and S. Hassan, “Survey on caching approaches in information centric networking,” *Journal of Network and Computer Applications*, vol. 56, pp. 48 – 59, 2015.

УДК 004.588

Студ. М.Н. Карпович
Науч. рук. доц. Н.В. Пацей
(кафедра программной инженерии, БГТУ)

АНАЛИЗ БИБЛИОТЕК АЛГОРИТМОВ MACHINE LEARNING

Машинное обучение (Machine Learning) - обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться. Существует два типа обучения. Обучение по прецедентам, или индуктивное обучение, основано на выявлении общих закономерностей по частным эмпирическим данным. Дедуктивное обучение предполагает формализацию знаний экспертов и их перенос в компьютер в виде базы знаний. Его принято относить к области экспертных систем.

Существующие инструменты глубокого обучения имеют различный функционал и требуют от пользователя разного уровня знаний и навыков. Правильный выбор инструмента — важная задача, позволяющая добиться необходимого результата за наименьшее время и с меньшей затратой сил.

Краткий обзор инструментов проектирования и обучения нейросетевых моделей. Основное внимание уделено четырем библиотекам: Caffe, Pylearn2, Torch и Theano. Рассмотрим базовые возможности указанных библиотек, примеры их использования. Сравним качество и скорость работы библиотек при конструировании одинаковых топологий нейросетей для решения задачи классификации рукописных цифр (в качестве обучающей и тестовой выборки используется датасет MNIST).

Далее в качестве исследуемого набора данных будет использоваться база изображений рукописных цифр MNIST. Изображения в данной базе имеют разрешение 28x28 и хранятся в формате оттенков серого. Цифры отцентрированы на изображении. Вся база разбита на