

## Последовательная и параллельная реализации гибридной жидкостной модели информационных потоков в компьютерных сетях со сложной топологией

Д. А. Басавин\*, С. В. Поршнева\*\*

\*Белгородский государственный аграрный университет имени В. Я. Горина  
308503, Белгородская обл., п. Майский, ул. Вавилова, 1

\*\*Уральский федеральный университет  
имени первого Президента России Б. Н. Ельцина  
620000, Екатеринбург, ул. Ленина, 51

e-mail: s.v.porshnev@urfu.ru

*Аннотация.* В статье обсуждаются результаты моделирования информационных потоков в компьютерных сетях (КС) со сложной топологией, полученные с помощью последовательной (написанной на языке пакета MATLAB) и параллельной (на основе технологии GPGPU) программных реализаций гибридной жидкостной модели (ГЖМ) интернет-трафика. Получены оценки скорости вычислений обеих программных реализаций, подтверждающие более высокую производительность параллельной программной реализации ГЖМ. Предложены направления дальнейших исследований с целью развития параллельной программной реализации ГЖМ.

*Ключевые слова:* параллельная гибридная жидкостная модель, гибридная жидкостная модель, модель интернет-трафика, компьютерные сети, GPGPU.

### 1. Введение

Для исследования особенностей информационных потоков, создаваемых в современных КС большим числом различных устройств и приложений, используются жидкостная модель (ЖМ) сетевого трафика [1] и ее дальнейшая модификация — гибридная жидкостная модель (ГЖМ), предложенная в [2]. ЖМ и ГЖМ представляют собой систему дифференциальных уравнений (СДУ) вида:

$$\frac{dW_i(t)}{dt} = \frac{l_{(w_i(t)-M_i)}}{R_i(t)} - \frac{W_i(t)}{2} \cdot \lambda_i(t), \quad (1)$$

$$\frac{dq_l(t)}{dt} = -l_{q(t)} \cdot C_l + \sum_{i \in N_l} A_i^l(t), \quad (2)$$

где  $W_i(t)$  — скорость передачи данных  $i$ -го потока;  $q_i(t)$  — длина очереди на входе в  $l$ -й канал;  $l_{f(t)}$  — функция Хевисайда<sup>1</sup>;  $R_i(q_i(t))$  — время от момента передачи «кванта»  $i$ -го информационного потока (например, пакета) его источником до момента поступления подтверждения о его получении от его приемника информации;  $\lambda_i(A_i^i(t), p_i^i(t))$  — общая скорость потерь пакетов  $i$ -го класса *TCP*-источников;  $C_l$  — пропускная способность  $l$ -го канала (скорость передачи данных);  $p_i^i(t)$  — вероятность сброса пакета  $i$ -го класса *TCP*-источников в  $l$ -м канале;  $A_i^i(t) = W_i^i(t)/R_i^i(t)$  — скорость передачи  $i$ -го потока в  $l$ -м канале. В ЖМ и ГЖМ учитываются конечность времени передачи данных от одного узла сети к другому (запаздывание), а также обратная связь, предусмотренная протоколом *TCP* [3], политики управления скоростью передачи данных в КС, механизмы сброса пакетов на маршрутизаторе, используемые для предотвращения его перегрузки, а также влияние потоков данных, создаваемых различными пользователями друг на друга. Вид функции  $\lambda_i(A_i^i(t), p_i^i(t))$  определяется выбираемыми характеристиками канала и механизмами ограничения скорости передачи данных (например, алгоритмы *Rate limiting*<sup>2</sup> и *AQM*<sup>3</sup>). С физической точки зрения данные модели представляют собой уравнения баланса потоков, входящих в соответствующий узел КС и выходящих из него.

В связи с отсутствием аналитических решений СДУ, используемых в ГЖМ, возникает необходимость нахождения их численных решений СДУ и, следовательно, разработки соответствующих программных реализаций. Известен вариант программной реализации ГЖМ с последовательно исполняемым кодом [4]. Однако опыт ее практического использования показал, что скорость вычисления характеристик информационных потоков даже для относительно простых конфигураций КС оказывалась весьма низкой. В этой связи в [5] была высказана гипотеза о возможности существенного увеличения скорости вычислений за счет использования технологий *GPGPU*. Для ее подтверждения авторами была разработана параллельная программная реализация ГЖМ, использующая *GPGPU* для вариантов сетей с

<sup>1</sup> Рассчитывается в соответствии с:

$$l_{f(t)} = \begin{cases} 1, & f(t) \geq 0 \\ 0, & f(t) < 0 \end{cases}$$

<sup>2</sup> *Rate limiting* (англ. — ограничение скорости доступа) или *Traffic shaping* (англ. — формирование трафика) — ограничение пропускной способности канала для отдельного узла сети ниже технических возможностей канала до узла. Шейпинг обычно используется как средство ограничения максимального потребления трафика со стороны узла сети.

<sup>3</sup> *Active Queue Management (AQM)*, (англ. — активное управление очередью) — технология в сетевых маршрутизаторах, заключающаяся в отбрасывании или установлении флага *ECN* пакетов до того, как очередь маршрутизатора заполнится.

одним маршрутизатором, анализ производительности которой был представлен в [6].

В статье обсуждаются результаты сравнения скорости расчетов параллельной программной реализации ГЖМ и соответствующей программной реализации ГЖМ с последовательно выполняемым кодом для многомаршрутизаторных вариантов КС. В проведенных исследованиях в экспериментах использовалась последовательная программная реализация ГЖМ, написанная на языке MATLAB [2], и параллельная программная реализация, написанная с использованием технологии GPGPU (язык C и библиотека OpenCL). В связи с тем, что соответствие количественных характеристик результатов расчетов, полученных с помощью последовательной программной реализации ГЖМ, и характеристик реального трафика подтверждено ранее в [2], в качестве критерия, свидетельствующего о правильности работы параллельной программной реализации ГЖМ, была проверена сходимость результатов расчетов, полученных при использовании каждой из программных реализаций ГЖМ.

## 2. Методика проведения экспериментов

В экспериментах проведено моделирование информационных потоков в КС, концептуальная схема топологии которой представлена на рис. 1.

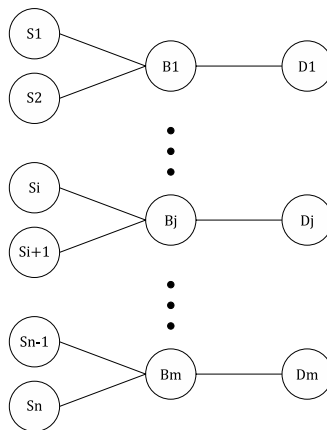


Рисунок 1. Схема топологии моделируемых сетей для эксперимента  $k$ ,  
 $S$  — TCP-источник интернет-трафика (генератор нагрузки),  
 $B$  — коммутационный узел (маршрутизатор),  $D$  — приемник нагрузки

Из рис. 1 видно, что в обсуждаемых экспериментах топология сети представляла собой совокупность групп небольших сетей, каждая из которых являлась подсетью (сегментом), состоящим из двух TCP-классов (генераторов нагрузки), одного

коммутационного узла (маршрутизатора), одного приемника и трех каналов связи. Общее количество сегментов сети для  $k$ -го эксперимента составляло  $m = 2^{k-1}$ ;  $k = 1, \dots, 9$ .

Для обеспечения максимальной нагрузки на вычислительные алгоритмы и аппаратные ресурсы с целью оценки и анализа производительности программных реализаций ГЖМ расчеты проводились для случая, когда каждый ТСР-класс генерировал нагрузку на протяжении всего времени моделирования. Варьируемыми параметрами ГЖМ в рамках одного эксперимента являлись: количество моделируемых ТСР-источников интернет-трафика, маршрутизаторов и приемников. Остальные параметры ГЖМ оставались неизменными. Данный подход позволил проводить идентичные эксперименты на последовательной и параллельных реализациях ГЖМ.

В  $k$ -м эксперименте использовались следующие параметры ГЖМ:

- время моделирования — 5 с;
- количество пользователей —  $n = 2^k$ ;
- количество маршрутизаторов —  $m = 2^{k-1}$ ;
- среднее время задержки распространения сигнала для каждого канала (RTT) — 4 мс;
- пропускная способность каналов — 1 Мбит/с;
- лимит скорости передачи для пользователей 1 Мбит/с;
- согласованный всплеск (BC) — 100 Кб;
- расширенный всплеск (BE) — 1000 Кб.

Замеры времени вычислений производились перед началом моделирования и после его окончания без учета операций инициализации GPU. Для каждого набора параметров вычислялись 20 независимых решений ГЖМ. Далее для каждого набора параметров вычислялись средние по ансамблю решения ГЖМ, полученные с помощью последовательной и параллельной реализаций ГЖМ, которые далее сравнивались друг с другом, а также время вычислений.

Для исключения влияния внешних факторов на производительность последовательной и параллельной программных реализаций ГЖМ в экспериментах использовалось одинаковое аппаратное обеспечение и операционная система.

### 3. Анализ результатов экспериментов

Сравнение усредненных по ансамблю независимых реализаций решений СДУ (1)–(2), полученных с помощью последовательной и параллельной реализаций ГЖМ, показали, что их отличия друг от друга оказываются в пределах погрешности, обусловленной конечной точностью компьютерных вычислений.

Усредненные по ансамблю реализаций замеры времени моделирования для 8 вариантов экспериментов представлены на рис. 2.

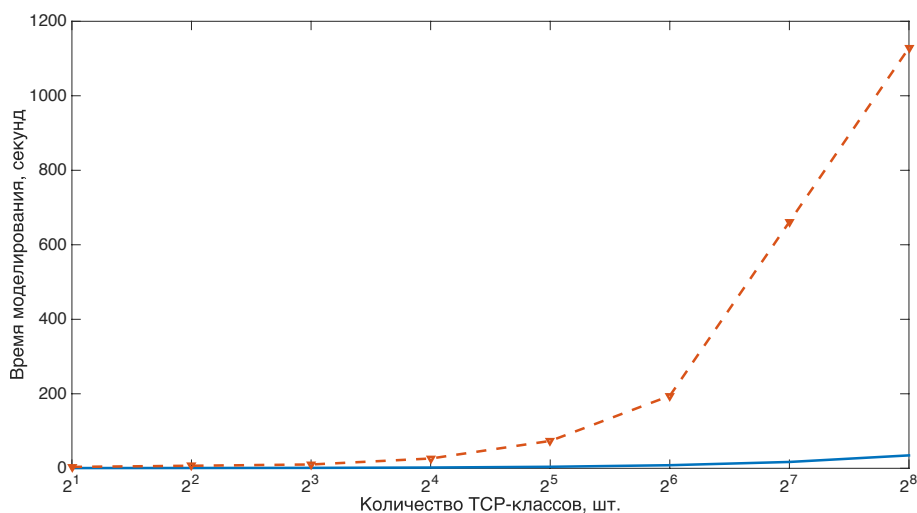


Рисунок 2. Зависимость времени решения СДУ (1)–(2) от числа TCP-классов: пунктирная линия — последовательная реализация ГЖМ; сплошная — параллельная реализация ГЖМ

Из рис. 2 видно, что время вычислений последовательной программной реализации ГЖМ при любом количестве TCP-источников превышает аналогичную величину, измеренную в экспериментах с параллельной реализацией ГЖМ, что свидетельствует о более высокой производительности последней. Для количественного подтверждения данного вывода в таблице приведены соответствующие значения времени вычислений.

Таблица. Сравнительный анализ программных реализаций

Количество TCP-классов	Время моделирования на GPU, сек	Время моделирования в MATLAB, сек	Ускорение, раз
2 <sup>1</sup>	0.684	3.760	5.494
2 <sup>2</sup>	0.818	6.823	8.338
2 <sup>3</sup>	1.186	10.245	8.635
2 <sup>4</sup>	2.084	26.224	12.583
2 <sup>5</sup>	4.083	73.505	18.005
2 <sup>6</sup>	8.329	194.542	23.357
2 <sup>7</sup>	16.960	661.531	39.005
2 <sup>8</sup>	34.699	1128.521	32.524

Отметим, что при использовании параллельных алгоритмов обработки на центральном процессоре (GPU) (например, встроенных в язык MATLAB) скорость вы-

числений в лучшем случае будет уменьшаться обратно пропорционально количеству ядер процессора. Однако даже при использовании 8 GPU оно окажется заметно больше аналогичной величины, получаемой при использовании параллельной реализации ГЖМ. Анализ опыта использования последовательной программной реализации ГЖМ, написанной на языке C [6], показывает, что она обеспечивает более высокую скорость вычислений в сравнении с программной реализацией, написанной на языке MATLAB, однако уступает параллельной программной реализации ГЖМ. Однако немаловажным фактом является тот, что при использовании последовательных вычислений на CPU приложению доступны большие объемы памяти (ОЗУ) для хранения параметров и результатов расчетов и, следовательно, возможно моделирование большего числа источников TCP-источников интернет-трафика (по нашим оценкам, их число может достигать  $2^{17}$ ) и КС со сложной топологией.

В рамках дальнейшего совершенствования программной реализации планируется разработка архитектуры системы, позволяющей использовать динамически подключаемые модули (плагины), для реализации новых алгоритмов AQM, TCP и др.

#### 4. Заключение

Подтверждена работоспособность параллельной программной реализации ГЖМ, разработанной на основе использования технологий GPGPU. Показано, что ее использование обеспечивает существенное увеличение скорости вычислений в сравнении с последовательной программной реализацией ГЖМ, созданной на языке MATLAB. Созданная авторами параллельная реализация ГЖМ, в отличие от пакетных симуляторов, обеспечивает возможность моделирования высокоскоростных КС со сложной топологией.

#### Литература

- [1] *Misra V., Gong W.-B., Towsley D.* Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED // Proceedings of ACM Sigcomm. — Stockholm, Sweden, August, 2000. P. 151–160.
- [2] *Гребенкин М. К., Поршнев С. В.* Гибридная жидкостная модель магистрального Интернет-канала. — Saarbrücken : LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012. 163 с.
- [3] Transmission Control Protocol [Электронный ресурс]. URL: <https://tools.ietf.org/html/rfc793>

- [4] Гребенкин М. К., Поршнев С. В. Программная реализация гибридной жидкостной модели информационных потоков в высокоскоростных магистральных интернет-каналах / Свидетельство о гос. регистрации программ для ЭВМ № 2012616118 от 4 июля 2012 г.
- [5] Басавин Д. А., Поршнев С. В. О целесообразности использования графических процессоров при моделировании крупных телекоммуникационных систем // Актуальные вопросы в научной работе и образовательной деятельности: сб. научн. тр. по материалам Междунар. науч.-практ. конф. 31 января 2013 г. в 13 ч. Ч. 2. — Тамбов : ТРОО «Бизнес-Наука-Общество», 2013. С. 12–15.
- [6] Басавин Д. А., Поршнев С. В. Оптимизация параллельной гибридной жидкостной модели высокоскоростных информационных потоков в магистральных интернет-каналах // Материалы Междунар. Крымской конф. «СВЧ-техника и телекоммуникационные технологии» (КрыМиКо), 2013.

**Авторы:**

Дмитрий Андреевич Басавин — ассистент кафедры «Информатика и информационные технологии», Белгородский государственный аграрный университет имени В. Я. Горина

Сергей Владимирович Поршнев — доктор технических наук, профессор, заведующий кафедрой радиоэлектроники информационных систем, Уральский федеральный университет имени первого Президента России Б. Н. Ельцина

## Serial and parallel implementations of hybrid fluid model of information flows in networks with complex topology

*D. A. Basavin\*, S. V. Porshnev\*\**

*\*Belgorod State Agricultural University named after V. Gorin  
Vavilova st., 1, Mayskiy, Belgorod region, 308503*

*\*\*Ural Federal University  
Lenina ave., 51, Ekaterinburg, 620000*

*e-mail: s.v.porshnev@urfu.ru*

*Abstract.* This paper discusses the results of modelling of information flows in networks with complex topologies and it depicts a comparison of the sequential (written in the MATLAB language) and parallel (based on GPGPU technology) software implementations of the hybrid fluid model (HFM) of Internet traffic. Obtained performance estimates of both software implementations indicate a higher performance of parallel software implementation HFM. The directions of further research, the results of which will be the basis for the later development of parallel software implementation HFM are proposed  
*Key words:* parallel hybrid fluid model, hybrid fluid model, Internet traffic model, computer networks, GPGPU.

### References

- [1] *Misra V., Gong W.-B., Towsley D.* (2000) Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. Proceedings of ACM Sigcomm. Stockholm, Sweden, August, pp. 151–160.
- [2] *Grebenkin M. K., Porshnev S. V.* (2012) Gibridnaya zhidkostnaya model' magistral'nogo Internet-kanala. Saarbrucken, LAP LAMBERT Academic Publishing GmbH & Co. KG [In Rus]
- [3] <https://tools.ietf.org/html/rfc793>
- [4] *Grebenkin M. K., Porshnev S. V.* Programmnaya realizatsiya gibridnoy zhidkostnoy modeli informatsionnykh potokov v vysokoskorostnykh magistral'nykh internet-kanalakh. Patent software RUS № 2012616118 4.07.2012 [In Rus]
- [5] *Basavin D. A., Porshnev S. V.* (2013) O tselesoobraznosti ispol'zovaniya graficheskikh protsessorov pri modelirovanii krupnykh telekommunikatsionnykh sistem. In Conf. Aktual'nyye voprosy v nauchnoy rabote i obrazovatel'noy deyatel'nosti, vol. 2. Tambov, TROO «Biznes-Nauka-Obshchestvo», pp. 12–15. [In Rus]
- [6] *Basavin D. A., Porshnev S. V.* (2013) Optimizatsiya paralel'noy gibridnoy zhidkostnoy modeli vysokoskorostnykh informatsionnykh potokov v magistral'nykh internet-kanalakh. In Conf. SVCH-tehnika i telekommunikatsionnyye tekhnologii (KryMiKo) [In Rus]