

# Разработка нагрузочного теста для исследования производительности распределенных файловых систем на основе подсчета ортогональных пар диагональных латинских квадратов

С.Н. Попов

Самарский национальный исследовательский университет  
им. академика С.П. Королева  
Самара, Россия  
stef4n.popov@gmail.com

С.В. Востокин

Самарский национальный исследовательский университет  
им. академика С.П. Королева  
Самара, Россия  
eastst@mail.ru

**Аннотация**—В статье описан опыт разработки нагрузочного теста для исследования производительности распределенных файловых систем. В качестве предмета исследования использовался подсчет ортогональных пар диагональных латинских квадратов. Выполнена программная реализация данного теста, пригодная для исполнения на произвольных невыделенных вычислительных ресурсах сети Интернет. Работоспособность программной реализации теста проверена в конфигурации на VPN-сервере с установленной системой TLJH, пользователи которой имитировали вычислительные ресурсы. Экспериментально проверена корректность выполнения тестов для различного количества разбиений массива ДЛК на файлы, а также приведен результат ускорения.

**Ключевые слова**— *распределенная файловая система, диагональные латинские квадраты, невыделенные вычислительные ресурсы, TLJH, VPN-сервер.*

## 1. ВВЕДЕНИЕ

Одним из направлений сокращения стоимости вычислений является использование так называемых невыделенных (non-dedicated) вычислительных ресурсов. Примерами таких ресурсов являются: машины добровольцев, предоставляемые ими временно для научных проектов [1]; свободные от основной нагрузки узлы кластеров [2] или суперкомпьютеров [3]; бесплатные или недорогие виртуальные машины облачных провайдеров [4]. Для учета специфики перечисленных окружений в вычислительных приложениях (compute-intensive) достаточно разбить вычисления на множество задач. Однако для использования невыделенных ресурсов в задачах обработки данных (data-intensive) необходимо также подобрать систему распределенного хранения, так как производительность системы хранения будет оказывать существенное влияние на эффективность вычислений в целом.

В связи с этим актуальными являются вопросы о выборе подходящей системы распределенного хранения, предсказания производительности приложения с выбранной системой, тонкой настройки приложения и т. п.

В качестве подхода к решению перечисленных вопросов в работе рассматривается нагрузочный тест и

его программная реализация, построенные на основе задачи подсчета пар диагональных латинских квадратов 7-ого порядка [5].

## 2. МЕТОДИКА ПОСТРОЕНИЯ ТЕСТА

Идея теста состоит в попарном сопоставлении специальных математических объектов — нормализованных диагональных латинских квадратов (ДЛК) 7-ого порядка. Пару ДЛК можно объединить в новый квадрат, каждый элемент которого получен объединением элементов исходных ДЛК в совпадающих позициях. При этом новый квадрат может также обладать свойством ДЛК: не содержать повторяющихся элементов во всех строках, столбцах, главной и побочной диагоналях. В этом случае говорят об ортогональности исходной пары ДЛК.

Для построения теста существенно, что вычислительный процесс, реализующий подсчет ортогональных пар латинских квадратов, (а) моделирует часто встречающийся на практике типовой информационный процесс попарной обработки данных в независимых задачах AllPairs; (б) имеет простой критерий корректности результата, так как число пар для ДЛК данного порядка известно; (в) существенно зависит от производительности от внутреннего алгоритма работы системы хранения; (г) массив данных для теста, включающий 171200 нормализованных ДЛК 7-ого порядка, уникален и генерируется программно.

Тривиальная стратегия хранения данных — размещение файлов с ДЛК на файловом сервере на некотором узле. При этом вычислительные узлы, выполняющие сопоставление пар ДЛК, будут каждый раз запрашивать файлы с файлового сервера. Простая оптимизация заключается в кэшировании данных на вычислительных узлах после их получения с файлового сервера, что позволяет избежать повторных передач файлов, если они переданы на узел ранее для другой задачи. Еще одной оптимизацией является прямая передача данных между кэшами систем хранения вычислительных узлов, минуя узел файлового сервера, что частично разгружает его канал передачи данных. Производительность процесса AllPairs улучшится за счет перечисленных оптимизаций, если они автоматически поддерживаются исследуемой системой хранения данных.

Тест проводится для различных разбиений множества ДЛК на файлы: все ДЛК в одном файле, ДЛК в 2х файлах, ДЛК в 3-х файлах и так далее. Файлы с ДЛК содержат примерно равное (разница не более чем на один квадрат) число квадратов. В тестах измеряется среднее время выполнения поиска ортогональных пар ДЛК. Тестирование разных разбиений множества ДЛК проводится до тех пор, пока не будет достигнут минимум времени вычисления. Снижение времени должно наблюдаться по причине распараллеливания, а повышение времени вычисления после достижения минимального значения происходит из-за снижения трудоёмкости обработки, приходящейся на единицу обрабатываемых данных (гранулярность) и, следовательно, росту доли накладных затрат на единицу обрабатываемых данных. Лучшая система распределенного хранения обеспечивает минимум времени счета и большее ускорение при прочих равных условиях, а также эффективнее работает с менее гранулярными задачами.

Разработанное приложение для тестирования использует генератор ДЛК, предоставленный авторами работы; подсистему запуска задач на основе Templet SDK [6, 7]; сервер платформы Everest [8, 9], распределяющий задачи среди программ-агентов Everest, установленных на ресурсах. Также на ресурсах устанавливаются клиенты исследуемой системы хранения. Развертывание и тестирование описанной программной инфраструктуры проведено на VPN-сервере с установленной системой TLJH [10]. Для имитации ресурсов использовались отдельные сессии TLJH, для имитации системы хранения использовался разделяемый пользователями TLJH каталог файловой системы VPN-сервера. Экспериментально проверена корректность выполнения тестов для различного количества разбиений массива ДЛК на файлы; получено ускорение 1.5 для разбиения массива на 2 части на 10 виртуальных процессорах.

### 3. ЗАКЛЮЧЕНИЕ

В работе предложен нагрузочный тест для исследования производительности распределенных файловых систем на основе алгоритма поиска ортогональных пар диагональных латинских квадратов 7-ого порядка. Выполнена программная реализация данного теста, пригодная для исполнения на

произвольных невыделенных вычислительных ресурсах сети Интернет. Работоспособность программной реализации теста проверена в конфигурации на VPN-сервере с установленной системой TLJH, пользователи которой имитировали вычислительные ресурсы. В дальнейшем тест планируется использовать для сравнительного исследования системы хранения Everest и системы IPFS [11] на предмет применимости в практических задачах обработки данных на невыделенных вычислительных ресурсах.

### ЛИТЕРАТУРА

- [1] Mengistu, T. Survey and Taxonomy of Volunteer Computing / T. Mengistu, D. Che // *ACM Computing Surveys*. – 2019. – P. 52. DOI: 10.1145/3320073.
- [2] Reddy, C.K.K. High Performance Computing Cluster System and its Future Aspects in Processing Big Data / C.K.K. Reddy, K.E. Bala Chandrudu, P.R. Anisha, G.V.S. Raju // *International Conference on Computational Intelligence and Communication Networks (CICN)*. – 2015. – P. 881-885. DOI: 10.1109/CICN.2015.173.
- [3] Ungurean, I. High-performance computing on a supercomputer based on new-generation processors / I. Ungurean, I. Rusu, S. Pentiu / *5th Romania Tier 2 Federation Grid, Cloud & High Performance Computing Science (RQLCG)*. – 2012. – P. 96-99.
- [4] Roloff, E. High Performance Computing in the cloud: Deployment, performance and cost efficiency / E. Roloff, M. Diener, A. Carissimi, P.O.A. Navaux // *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. – 2012. – P. 371-378. DOI: 10.1109/CloudCom.2012.6427549.
- [5] Kochemazov, S.E. Fast Algorithm for Enumerating Diagonal Latin Squares of Small Order / S.E. Kochemazov, E.I. Vatutin, O.S. Zaikin / *ArXiv preprint: 1709.02599*, 2017.
- [6] Vostokin, S.V. Templet: A markup language for concurrent actor-oriented programming / S.V. Vostokin // *CEUR Workshop Proceedings*. – 2016. – Vol. 1638. – P. 460-468.
- [7] The Templet Project [Electronic resource]. — Access mode: <https://github.com/the-templet-project> (06.02.2022).
- [8] The Everest Project [Electronic resource]. — Access mode: <http://everest.distcomp.org> (06.02.2022).
- [9] Sukhoroslov, O. Web-Based Platform for Publication and Distributed Execution of Computing Applications / O. Sukhoroslov, S. Volkov, A.A. Afanasiev // *14th International Symposium on Parallel and Distributed Computing (ISPDC)*. IEEE. – 2015. – P. 175-184.
- [10] The Littlest Jupyter rHub project [Electronic resource]. — Access mode: <https://tljh.jupyter.org> (06.02.2022).
- [11] Muralidharan, S. An InterPlanetary File System (IPFS) based IoT framework / S. Muralidharan, H. Ko. // *IEEE International Conference on Consumer Electronics (ICCE)*. – 2019. – P. 1-2. DOI: 10.1109/ICCE.2019.8662002.