

УДК 004.623

## ОПТИМИЗАЦИЯ ВРЕМЕНИ СБОРА ДАННЫХ О ТРАНЗАКЦИЯХ В СЕТИ SOLANA



**А.Г. Бокун**  
Ассистент кафедры  
информатики, БГУИР.  
a.bokun@bsuir.by



**А.В. Герчик**  
Ассистент кафедры  
информатики, БГУИР.  
a.gerchik@bsuir.by

### **А.Г. Бокун**

Окончил Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов связана с информатикой, теоретической и прикладной криптографией, а также технологиями шифрования и защиты информации.

### **А.В. Герчик**

Окончил Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов связана с исследованием роли и перспективы интеграций технологии дополненной реальности в различных отраслях современного мира.

**Аннотация.** В статье исследована работа одной из составляющих систем валидатора сети Solana – модуля сбора и трансляции данных о прошедших транзакциях *geyser*. Во введении обозначена важность и актуальность вопроса сбора данных в высокопроизводительных сетях. В основной части статьи приведен эксперимент, позволяющий больше узнать о тонкостях работы стандартного решения. Практическая ценность полученных результатов оценивается в заключении.

**Ключевые слова:** Solana, блокчейн, криптография, *geyser*, валидация, транзакции.

**Введение.** Последние разработки в области прикладной криптографии, блокчейн-сетей и децентрализованных финансов получили новый импульс для активного развития ввиду роста рынка криптовалют и институционального признания токенов с наибольшей капитализацией (BTC и ETH) через создание ETF, выпускающего публично торгуемые ценные бумаги, которые отражают движение цен на фьючерсные контракты целевой криптовалюты. Это показывает заинтересованность традиционного финансового капитала в работе со средствами децентрализованной торговли и накопления ликвидности и делает более актуальными и экономически обоснованными исследования в этой области.

Актуальной задачей является разработка инструментов сбора и анализа большого объема данных о прохождении транзакций в блокчейн-сетях. Агрегация подобной информации решает сразу несколько важных задач: анализ движения активов на рынке, оценка и мониторинг работоспособности сети и облегчение процесса разработки и внедрения смарт-контрактов.

**Блокчейн Solana.** В работе рассматривается блокчейн-сеть Solana. Она обладает крайне высокой скоростью работы: актуальный TPS (transaction per second) основной сети (mainnet) составляет 4000, а время создания блока – 0.4 секунды. Данные показатели делают сеть крайне чувствительной к любым внедрениям в процесс валидации транзакций, так как это может значительно повлиять на ее TPS [1].

Для сбора информации о проходящих в сети транзакциях валидаторы Solana имеют поддержку плагина *geyser*. Данный плагин предоставляет разработчикам интерфейс

взаимодействия с валидатором в части получения данных об интересующих сущностях сети. На данный момент `geyser` может отправлять данные об аккаунтах, транзакциях, слотах, блоках и записях. Для дальнейшей работы необходимо четко определить следующие термины, опуская некоторые подробности реализации сети.

– блок – непрерывный набор записей в списке валидатора, покрываемый голосованием. Лидирующий валидатор производит не более одного блока за слот.

– запись – часть блока, содержащая в себе информацию об обработанной транзакции. В блоке должно быть как минимум 64 записи.

– слот – период времени, в течение которого каждый лидирующий валидатор принимает транзакции и создает блок (на данный момент равен 0.4 секунды).

– транзакция – одна или несколько инструкций, подписанных клиентом с помощью одной или нескольких пар ключей и выполняемых атомарно лишь с двумя возможными исходами: успех или неудача [2].

**Постановка вопроса.** Уже было отмечено, что для ретрансляции информации о сущностях сети валидаторы используют интерфейс `geyser` к которому могут подключаться внешние программы с помощью `extern “C”` метода с особым именем `_create_plugin()`, как это делает, например, `solana-accountsdb-plugin-postgres` – реализация доступа к `geyser`-интерфейсу с записью информации в базу данных PostgreSQL. Однако не для всех разработчиков может быть приемлема работа только лишь со стандартным интерфейсом, в таком случае ставится вопрос: в какой момент наиболее эффективно и безопасно проводить сбор данных о прошедшей через валидатор транзакции? Было принято четыре возможной гипотезы, отвечающей на этот вопрос:

- после выпуска блока (окончания слота), которому принадлежит транзакция;
- после добавления транзакции в запись и закрытия записи;
- после добавления транзакции в запись;
- перед добавлением транзакции в запись.

Также было принято за аксиому, что наиболее оптимальным и безопасным для работоспособности сети будет тот вариант, который был реализован разработчиками в стандартном интерфейсе.

**Эксперимент.** Для проведения исследования локально был развернут тестовый валидатор и установлен плагин `solana-accountsdb-plugin-postgres` для сохранения данных в PostgreSQL. К сожалению данная реализация плагина (как и любая другая из существующих на сегодняшний день) не собирает данные о записях, поэтому с его помощью получится проверить только первую гипотезу. Плагин создает в базе данных отношения с большим количеством полей, однако в рамках эксперимента большинство из них не имеет значения. Ключевыми для эксперимента являются поля:

- для транзакции: `slot` (номер слота локальной сети, в который попала транзакция), `updatet_on` (время изменения записи о транзакции) и `index` (номер транзакции в блоке);
- для блока: `slot` (номер слота локальной сети, в который попал блок), `updatet_on` (время изменения записи о блоке).

Кроме этого в исследовании будет использоваться собственное программное обеспечение, предназначенное для генерации потока транзакций на заданном валидаторе. Также стоит отметить, что название поля `updatet_on` указывает на то, что временная отметка будет меняться при изменении записи плагином. Чтобы избежать искажения данных было принято решение создать дополнительные отношения и триггеры, которые при добавлении новых кортежей в отношения `transaction` и `block` будет копировать ключевые для эксперимента поля в новые отношения. Это будет гарантировать, что временные метки не были изменены в процессе работы плагина и валидатора.

Для первого этапа эксперимента будет сгенерирован поток из десяти транзакций, которые должны будут попасть на один слот (то есть в один блок).

Все транзакции попали на слот 14404 локального валидатора. Блок, принадлежащий этому слоту был создан в НН:ММ:00.892. В таблице 1 будет представлена информация о транзакциях (10 сгенерированных транзакций + 1 голосующая, создаваемая валидатором автоматически), принадлежащих слоту 14404.

Таблица 1. Информация о транзакциях слота 14404.

Индекс	Время создания	Разница с временем создания блока (с)
0	НН:ММ:00.493	0.399
1	НН:ММ:00.550	0.342
2	НН:ММ:00.563	0.329
3	НН:ММ:00.563	0.329
4	НН:ММ:00.569	0.323
5	НН:ММ:00.587	0.305
6	НН:ММ:00.593	0.299
7	НН:ММ:00.593	0.299
8	НН:ММ:00.600	0.292
9	НН:ММ:00.600	0.292
10	НН:ММ:00.610	0.282

Данные свидетельствуют, что все транзакции попали в блокчейн и получили индексы внутри блока корректно. Также видно, что между добавлением информации о последней транзакции и добавлением информации о блоке прошло 0.282 секунды.

Предварительные результаты эксперимента говорят о несостоятельности первой гипотезы. Второй этап предполагает проведение исследования на большем количестве слотов, при большем потоке сгенерированных транзакций. В данном случае была сгенерирована нагрузка из 1000 транзакций, они попали на слоты локальной сети 14602-14608. Ввиду этого для визуализации результат будет принята сквозная индексация транзакций между блоками исследуемых слотов. На рисунке 1 представлен результат второго этапа исследования в виде графика зависимости времени добавления сущности сети в базу данных от ее сквозного индекса.

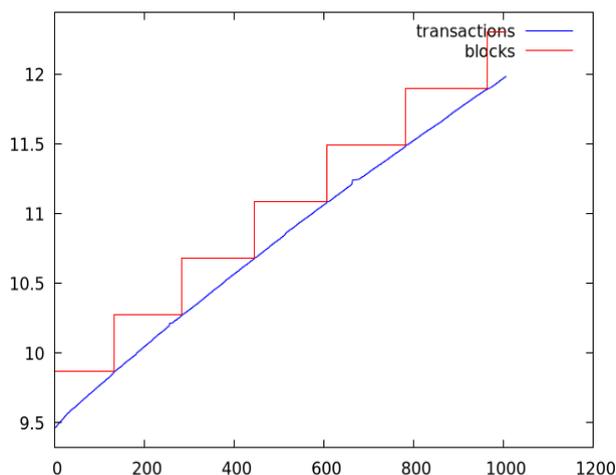


Рисунок 1. График зависимости времени добавления сущностей сети (транзакций и блоков) в базу данных от ее сквозного индекса

Точки графика, относящегося к транзакциям всегда принадлежат площади фигуры, находящейся под графиком, относящемся к блокам. Точки времени добавления блоков максимально приближаются к графику транзакций, однако не достигают его. На рисунке 2 показан участок приближения точки добавления блока, принадлежащего слоту 14602, к графику транзакций.

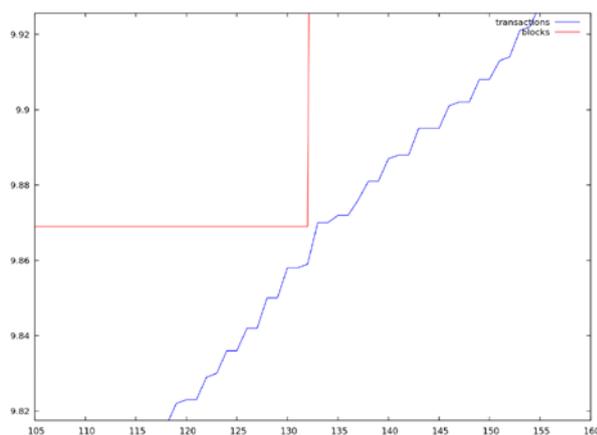


Рисунок 2. Участок приближения точки добавления блока, принадлежащего слоту 14602, к графику транзакций

Полученные экспериментальные данные позволяют опровергнуть первую гипотезу.

Ранее было отмечено, что на сегодняшний день нет открытых плагинов, которые были бы заинтересованы в сборе данных о записях через интерфейс *geyser*. В связи с этим необходимо модифицировать интерфейс таким образом, чтобы он вызывал методы оповещения о записях даже когда нет вызывающего плагина, которому нужно отправить данные. Для этого необходимо модифицировать файлы *geyser\_plugin\_interface.rs* и *geyser\_plugin\_manager.rs*. В первом случае надо заменить возвращаемое значение функции *transaction\_notifications\_enabled* на *true*, эта функция оповестит интерфейс, что есть существуют интересанты получения данных о записях. Во втором случае необходимо изменить функцию *entry\_notifications\_enabled* таким образом, чтобы она во всех возможных ветках своего выполнения возвращала *true*, это позволит эмулировать поведение внешнего плагина, запрашивающего данные.

Далее необходимо добавить в начало функции *notify\_entry*, принадлежащей типу *EntryNotifier* в имплементации этого типажа для структуры *EntryNotifierImpl* код, представленный в листинге 1.

```
let now = Local::now();
let mut file = OpenOptions::new().append(true).open("entry.txt").unwrap();
file.write_all(
    String::from(format!(
        "slot: {}, index: {}, entry (num_transactions): {}, hash: {}, timestamp:
    {})\n",
        slot,
        index,
        entry.num_transactions,
        entry.hash.to_string(),
        now.format("%Y-%m-%d %H:%M:%S%.9f"),)).as_bytes(),);
```

*Листинг 1. Вставка в функцию notify\_entry*

Код данной вставки будет получать входящий поток оповещений о записях и сохранять информацию о записях, а именно:

- слот, которому принадлежит запись;
- индекс записи в блоке;
- количество транзакций в записи;
- хэш записи;
- временная отметка;
  - построчно в указанный файл.

Скорость предложенного способа сбора данных будет значительно отличаться от скорости отправки информации в базу данных плагином, так как здесь исключаются накладные расходы на пересылку данных по сети, но появляются издержки работы с файлами. Для сохранения корректности измерений (в первую очередь временных) было принято решение на текущем этапе эксперимента собирать данные о транзакциях аналогичным образом (сохранять данные в файл, эмулируя наличие плагина-интересанта). В таблице 2 будет представлена информация о транзакциях и записях, которые содержат в себе эти транзакции, все данные принадлежащих слоту 14704.

Таблица 2. Информация о транзакциях и записях

Индекс транзакции	Время добавления транзакции (Т)	Индекс записи	Время добавления записи (Е)	Разница (Т - Е, с)
0	НН:ММ:40.961892434	1	НН:ММ:40.961703902	0.000188532
1	НН:ММ:41.276283850	52	НН:ММ:41.275962329	0.000321521
2	НН:ММ:41.288145746	54	НН:ММ:41.288069011	0.000076735
3	НН:ММ:41.288608823	56	НН:ММ:41.288494233	0.00011459
4	НН:ММ:41.295746669	58	НН:ММ:41.295608933	0.000137736
5	НН:ММ:41.296041318	59	НН:ММ:41.295960419	0.000080899
6	НН:ММ:41.312644970	62	НН:ММ:41.312569852	0.000075118
7	НН:ММ:41.313114525	63	НН:ММ:41.313040279	0.000074246
8	НН:ММ:41.320464516	66	НН:ММ:41.320357877	0.000106639

9	НН:ММ:41.326665713	68	НН:ММ:41.326576463	0.00008925
10	НН:ММ:41.331976224	69	НН:ММ:41.331941086	0.00003513

Полученные данные демонстрируют, что информацию о записи интерфейс отдает раньше, чем информацию о транзакциях, которые этой записи принадлежат. То есть сначала происходит окончательное формирование и закрытие записи (вывод сделан на основе того, что в число получаемой информации о записи входят данные, которые могут быть доступны только после окончательного формирования, например количество транзакций в записи или хэш), отправка оповещения о записи, а после этого отправка оповещений о всех транзакциях.

Для проведения второго этапа исследования также был сгенерирован поток из 1000 транзакций, которые попали на слоты локальной сети 20468-20475. Стоит отметить, что по правилам сети любой блок должен содержать в себе как минимум 64 записи, в связи с этим валидатор на каждом слоте, помимо прочих, генерирует 64 пустых записи. В целях повышения соответствия данных эти записи были исключены из исследовательской выборки. На рисунке 3 представлен результат второго этапа исследования в виде графика (с использованием сквозного индексирования).

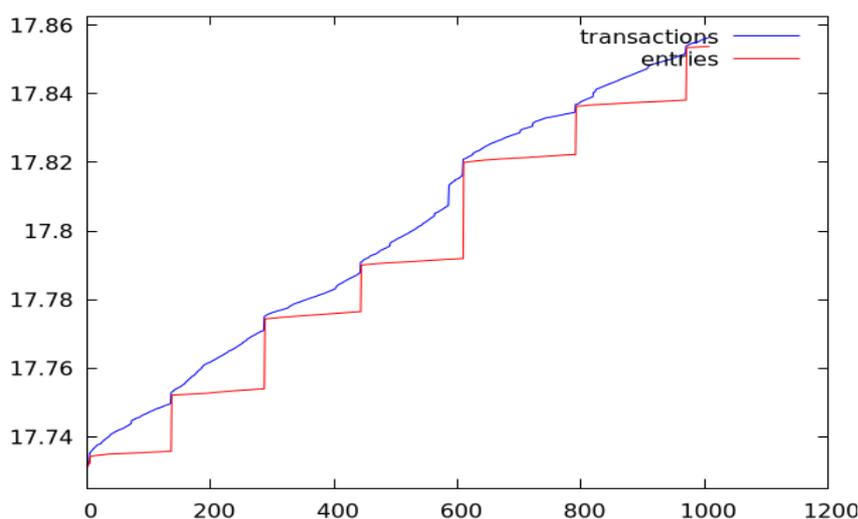
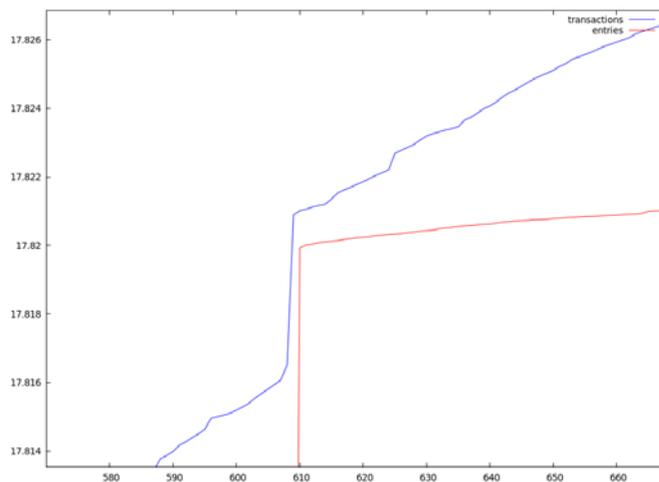


Рисунок 3. График зависимости времени добавления записей и транзакций в файл от ее сквозного индекса

Ступенчатый вид графика записей объясняется исключением из выборки пустых записей. В целом, полученные данные позволяют подтвердить сделанное ранее предположение о времени отправки оповещения о транзакции. Как и в случае с графиком на рисунке 1 видны точки потенциального касания графиков, однако ни в одной из точек касания не происходит. На рисунке 4 показан участок приближения точки добавления нулевой транзакции, принадлежащей слоту 20473, к графику записей.



*Рисунок 4.* Участок приближения точки добавления нулевой транзакции, принадлежащей слоту 20473, к графику записей

Таким образом было подтверждено, что информация о транзакциях отправляется интерфейсом geyser после закрытия записи, которой эта транзакция принадлежит.

**Вывод.** На основе практического исследования ответом на вопрос, который был поставлен в начале статьи, является гипотеза номер 2. Информация о транзакциях интерфейсом валидатора Solana geyser отправляется после добавления транзакции в запись и закрытия записи. Полученный результат будет ценен при проектировании прокси-узлов для валидаторов и инструментов анализа сети Solana.

### Список литературы

- [1] Mishra D.P. и др. Solana blockchain technology: a review // International Journal of Informatics and Communication Technology (IJ-ICT). Institute of Advanced Engineering and Science. – 2024. – Т. 13, № 2. – С. 197.  
[2] Qiu J. A Hybrid Data Decomposition and Deep Learning Approach for Solana Price Prediction Incorporating Market Factors // Advances in Economics, Management and Political Sciences. EWA Publishing. – 2025. – Т. 162, № 1. – С. 6–14.

### Авторский вклад

**Бокун Артем Геннадьевич** – составление плана эксперимента, сбор и анализ данных о транзакциях на локальном валидаторе с использованием специального программного обеспечения, интерпретация и визуализация результатов исследования.

**Герчик Артем Вадимович** – проведение этапов исследования с использованием программного обеспечения, предназначенного для генерации потока транзакций, сбор данных, анализ полученных результатов, формирование структуры статьи.

## OPTIMIZING TRANSACTION DATA COLLECTION TIME IN THE SOLANA NETWORK

*A.G. Bokun*

*Assistant, Department of Computer  
Science, BSUIR.*

*A.V. Gerchik*

*Assistant, Department of  
Computer Science, BSUIR.*

**Abstract.** The paper studies the work of one of the components of the Solana network validator systems – geyser, the module for collecting and broadcasting data on past transactions. In the introduction the importance and relevance of the issue of data collection in high-performance networks is outlined. The main part of the article contains an experiment that allows us to learn more about the subtleties of the standard solution. The practical value of the obtained results is evaluated in the conclusion.

**Keywords:** Solana, blockchain, cryptography, geyser, validation, transactions.