



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики
Кафедра Суперкомпьютеров и Квантовой Информатики

КУРСОВАЯ РАБОТА

Исследование влияния системы мониторинга производительности на выполнение коллективных MPI операций

Выполнил:

Скрябин Глеб Денисович

Научный руководитель:

Стефанов Константин Сергеевич

Москва 2022

Содержание

1. Введение	3
2. Цель работы.....	3
3. Метод проверки и коллективные MPI операции	3
4. Критерий обнаружения шума системы мониторинга.....	4
5. Влияние шума на детектор	5
6. Влияние шума на детектор, привязка к ядрам	8
7. Ссылки	10

1. Введение

Параллельное приложение, запускаемое на суперкомпьютере, может быть исследовано с помощью системы мониторинга производительности. Агент системы мониторинга собирает информацию о состоянии программно-аппаратной среды во время запуска параллельного приложения. Однако агент системы мониторинга производительности оказывает дополнительную нагрузку, помехи на пользовательскую задачу, так как для сбора данных также использует ресурсы вычислительной системы. Помехи в дальнейшем будем называть шумом системы мониторинга производительности.

В дипломной работе, выбранной для повторных исследований, изучается влияние агента системы мониторинга производительности суперкомпьютера Ломоносов-2 на представленный детектор шума, чувствительному малому шуму этой системы. Данный детектор основан на выполнении коллективных MPI операций All-to-All и Barrier.

2. Цель работы

Целью работы является проверка исследований влияния системы мониторинга производительности суперкомпьютера Ломоносов-2 на сильно синхронизированные программы, содержащие коллективные MPI операций All-to-All и Barrier и повторение всех опытов для подтверждения и сравнения результатов.

3. Метод проверки и коллективные MPI операции

Проверка состоит в ~~повторном исследовании~~ влияния системы мониторинга производительности суперкомпьютера на сильно синхронизированные программы, используя те же методы и похожие параметры программ.

Запуски программ проводились в ~~те~~стовом разделе, что несколько затрудняет работу, так как имеется ограничение по времени работы программ, чего не было во всех изначальных опытах. Для того, чтобы в таких условиях получить результаты, схожие с изначальными, требуется уменьшить параметры, задающие уровень нагрузки на вычислительный узел суперкомпьютера. Это может повлиять на точность, но при этом сравнения результатов все еще возможны.

Для исследования используются программы, содержащие большое количество коллективных MPI операций All-to-All и Barrier. Такие программы моделируются с помощью детектора шума, в основе которого лежит цикл, содержащий MPI операции All-to-All или Barrier. В изначальной ВКР было показано, что такой детектор является чувствительным к небольшому искусственному шуму.

Используемый детектор шума чувствителен к реальному шуму системы мониторинга производительности суперкомпьютера Ломоносов-2, на котором установлена система мониторинга производительности dimmon. Стандартный сбор данных для агента этой системы производится с частотой 1 раз в секунду. При такой частоте сбора данных агент системы мониторинга dimmon занимает до 0,3% процессорного времени, которое можно считать малым.

4. Критерий обнаружения шума системы мониторинга

При запуске используемого детектора шума при длительности 10 - 15 минут, когда в цикле выполняется большое число коллективных операций, можно собрать сравнительно небольшую статистику времени работы детектора. В таком случае будем считать, что детектор обнаруживает шум системы мониторинга, если 99% доверительные интервалы времени запусков детектора без и с шумом удалены друг от друга и не пересекаются. Расстояние между доверительными интервалами называется *boundary gap*. Влияние детектора может быть обнаружено статистически, если *boundary gap* имеет положительное значение.

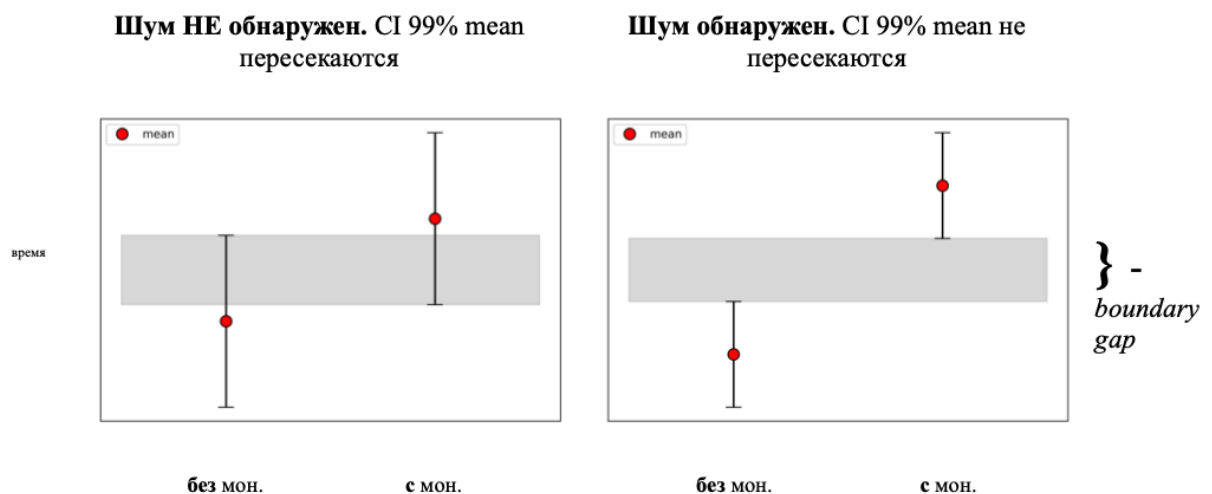


Рис. 1: Варианты расположения доверительных интервалов, представленные в исследуемой работе.

Здесь и далее, графики, описывающие повторные опыты для сравнения будут представлены вместе с изначальными результатами. На графиках по ординатам - шкала времени, которая указана в секундах; синим крестиком обозначено среднее значение времени запусков детектора шума. По координатам - конфигурации запусков, обозначена в виде [количество узлов]_[количество логических ядер на узел].

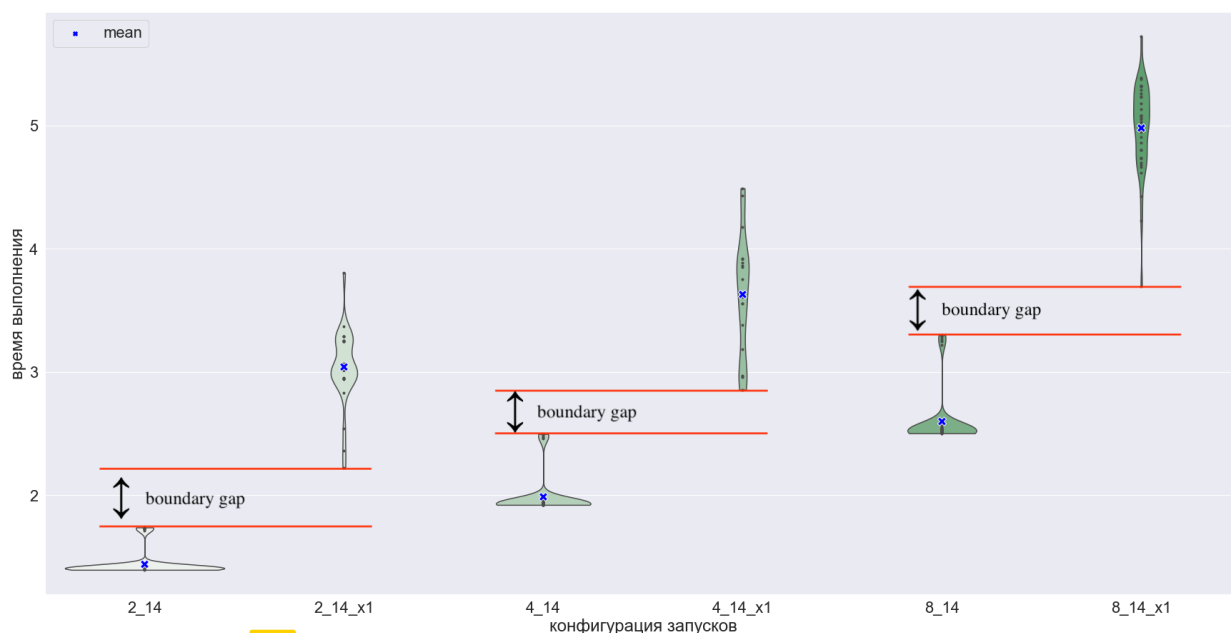


Рис. 2: Пример обнаружения шума при использовании 2, 4 и 8 узлов

5. Влияние шума на детектор

Приведем результаты запусков детекторов с операцией All-to-All и Barrier на 2-х и 4-х узлах. Детектор запущен на 14 ядрах каждого узла. Во всех четырех случаях значение boundary gap положительное. Это означает, что детектор обнаруживает агент системы мониторинга, работающий в стандартном режиме с частотой 1 Гц, что соответствует результатам изначальных исследований. Так же надо отметить, что в данной работе не использовались конфигурации с 28 логическими ядрами на узел, в некоторых случаях требовалось вручную поставить выполнение детектора на те же узлы, на которых выполняется задача.

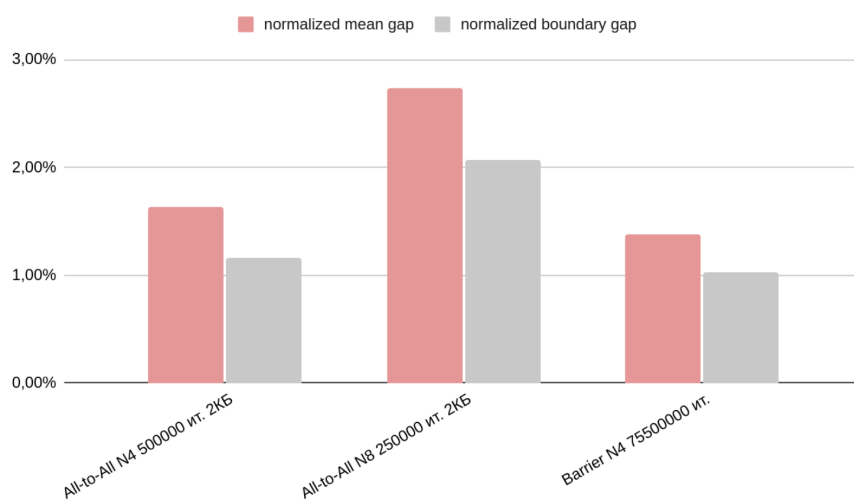


Рис. 3: Результаты запусков детектора. 28 ядер на узел, представленные в исследуемой работе.

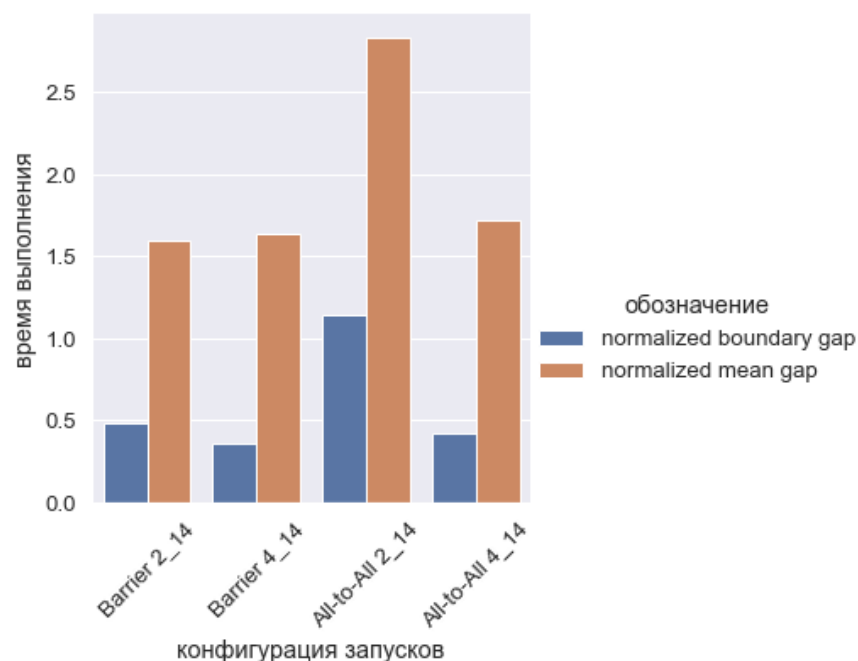


Рис. 4: Результаты запусков детектора

Обозначения на графиках:

- normalized mean gap – разница между средними значениями времени работы детектора с и без мониторинга.
- normalized boundary gap – boundary gap. Если значение положительное, значит детектор обнаруживает шум 1 Гц.

После того, как было показано, что время работы детектора, работающего на тех же узлах, что и задача, увеличивается от шума агента системы dimmon, было решено проверить, можно ли обнаружить систему мониторинга, если детектор запущен стандартно, когда, каждый процесс работает на отдельном ядре. Так запускается большинство приложений на СК, это стандартный способ постановки задачи.

На рисунках 5 (представленный в исследуемой работе), 6 и 7 представлены результаты запусков. Видно, что для каждого режима работы мониторинга - с частотой 1, 5 и 10 Гц, доверительные интервалы времени запусков детектора, работающего без системы мониторинга и с ней пересекаются для всех рассмотренных конфигураций детектора. В каждом случае влияние агента системы мониторинга статистически не обнаруживается. В данных опытах используется конфигурация с 14 ядрами на узел.

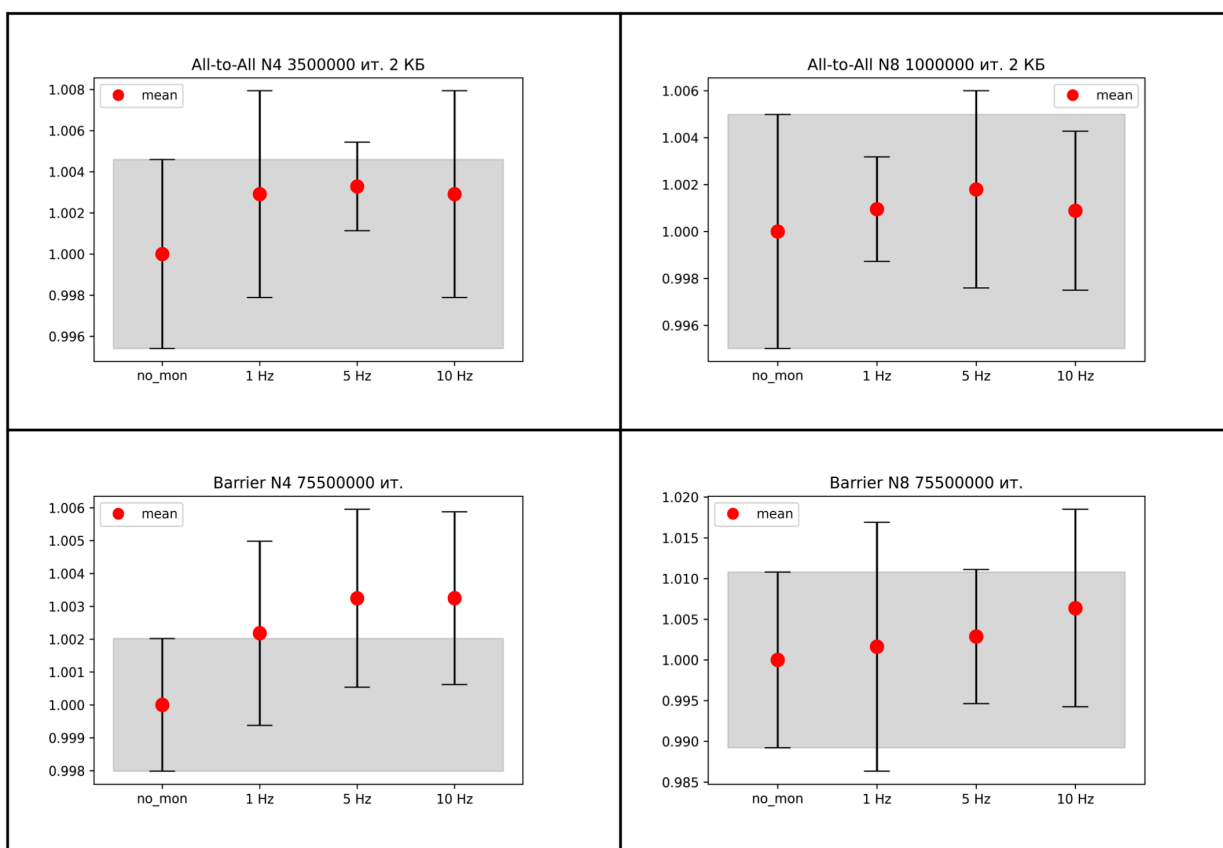


Рис. 5: Результаты запусков детектора, представленные в исследуемой работе. Детектор использует 14 ядер на узел.

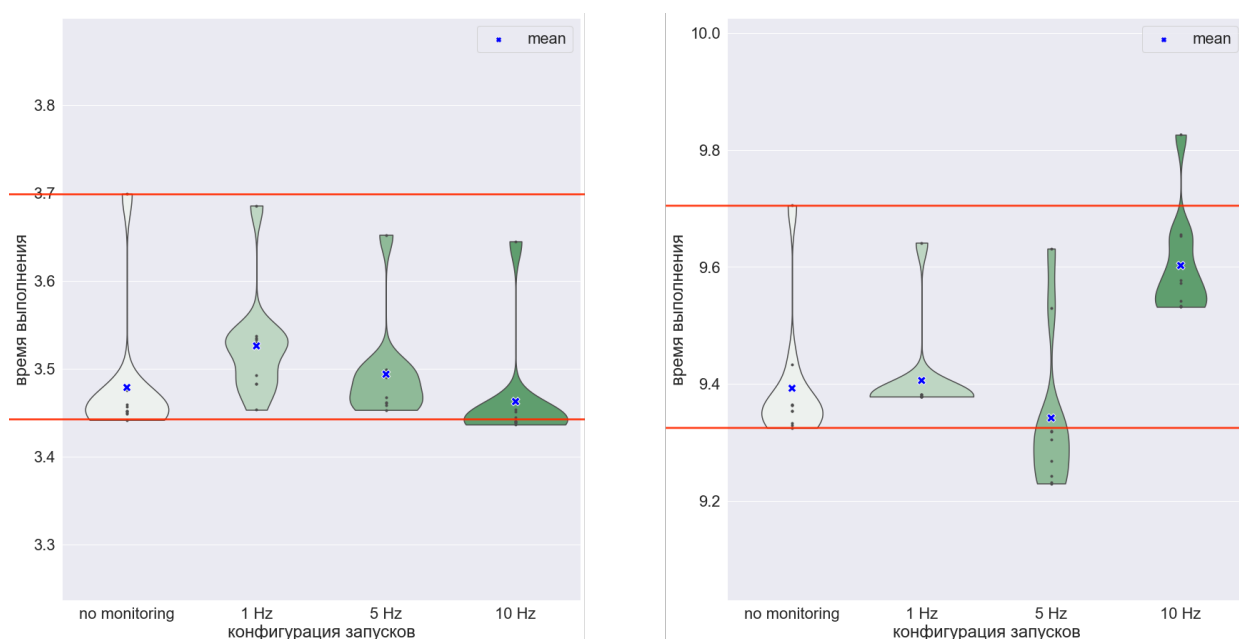


Рис. 6: Результаты запусков детектора. All-to-All, 4 и 8 узлов соответственно. Детектор использует 14 ядер на узел.

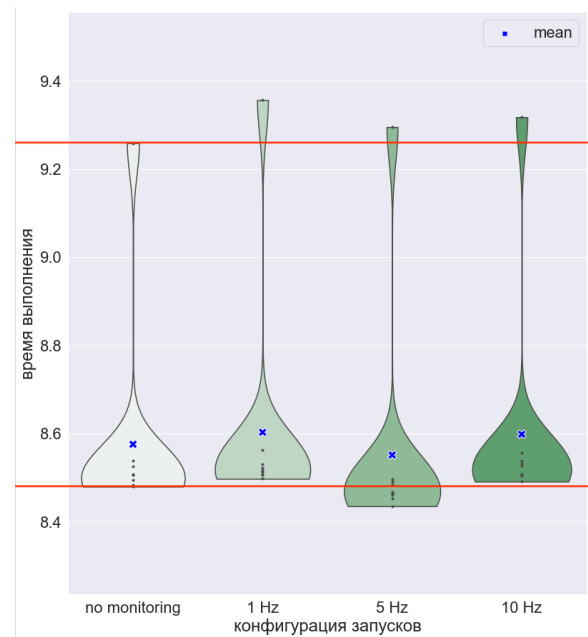
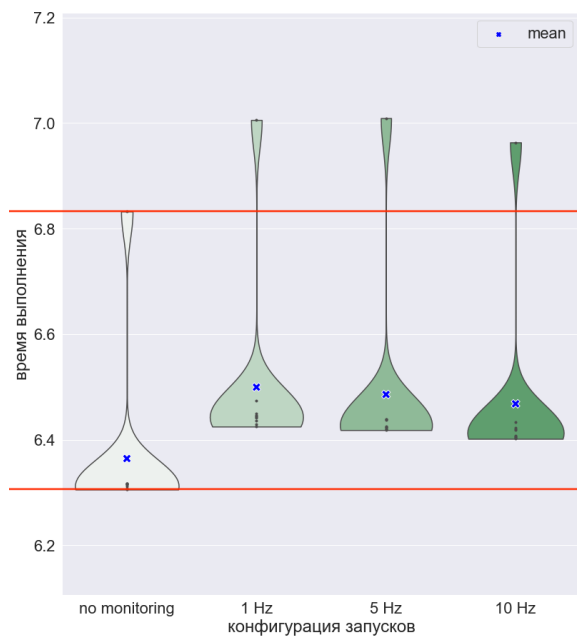


Рис. 7: Результаты запусков детектора. Bargier, 4 и 8 узлов соответственно. Детектор использует 14 ядер на узел.

6. Влияние шума на детектор, привязка к ядрам

Для продолжений исследований, и для получения более точных и понятных результатов, попробуем установить привязку детектора и агента системы мониторинга к ядрам. Будем рассматривать три варианта работы детектора:

- А - детектор работает без системы мониторинга, число процессов совпадает с числом физических ядер, каждый процесс работает на отдельном ядре;
- В - детектор и мониторинг расположены на логических ядрах с разными номерами. Число процессов детектора совпадает с числом физических ядер;
- С - агент привязан к логическому ядру, на котором выполняется детектор шума. Число процессов детектора совпадает с числом физических ядер.

В этом случае рассматриваются интервалы времени работы меньшего размера и тогда будем считать, что детектор обнаруживает систему мониторинга, если время работы детектора с мониторингом можно отнести к выбросу, относительно значений выборки по времени запуска детектора без мониторинга.

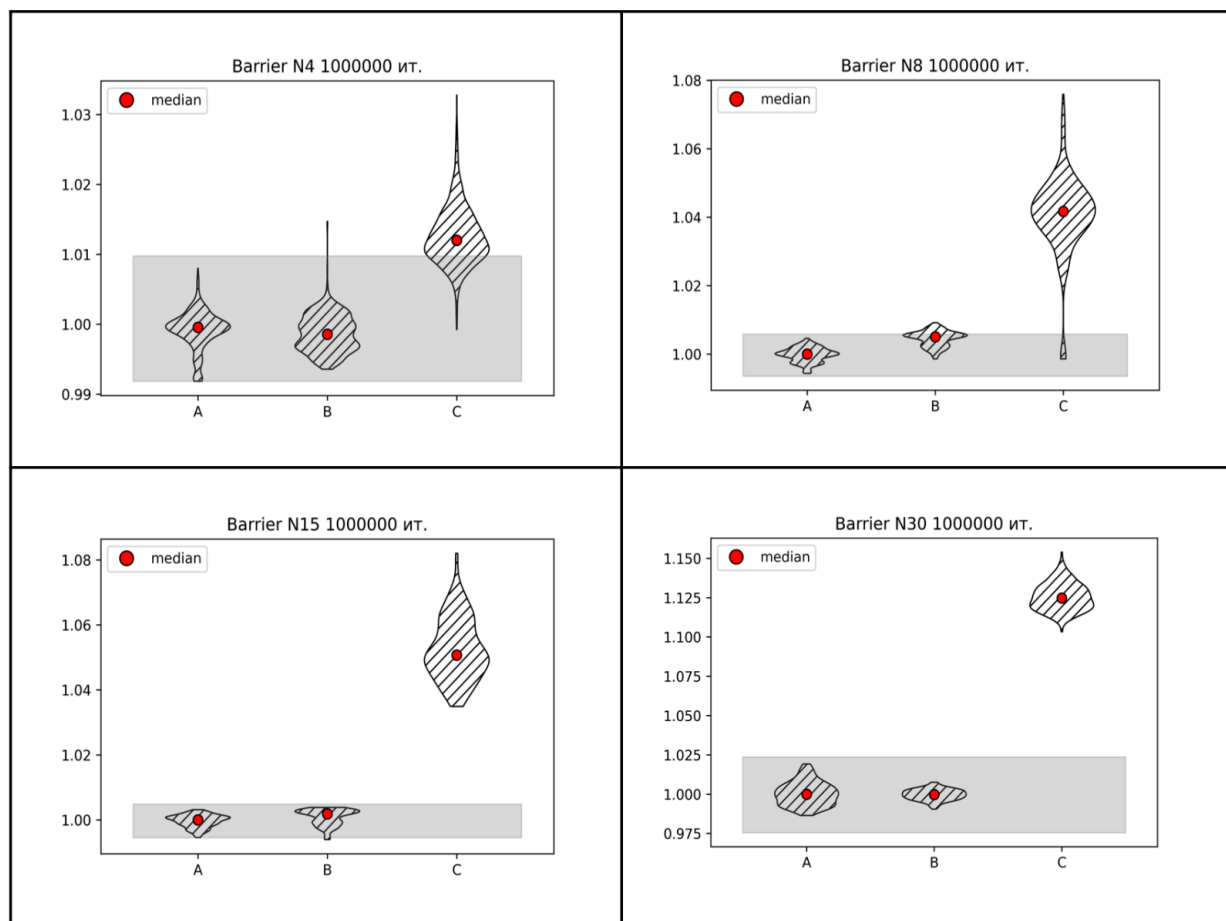


Рис. 8: Barrier 14 ядер на узел, представлены в исследуемой работе.

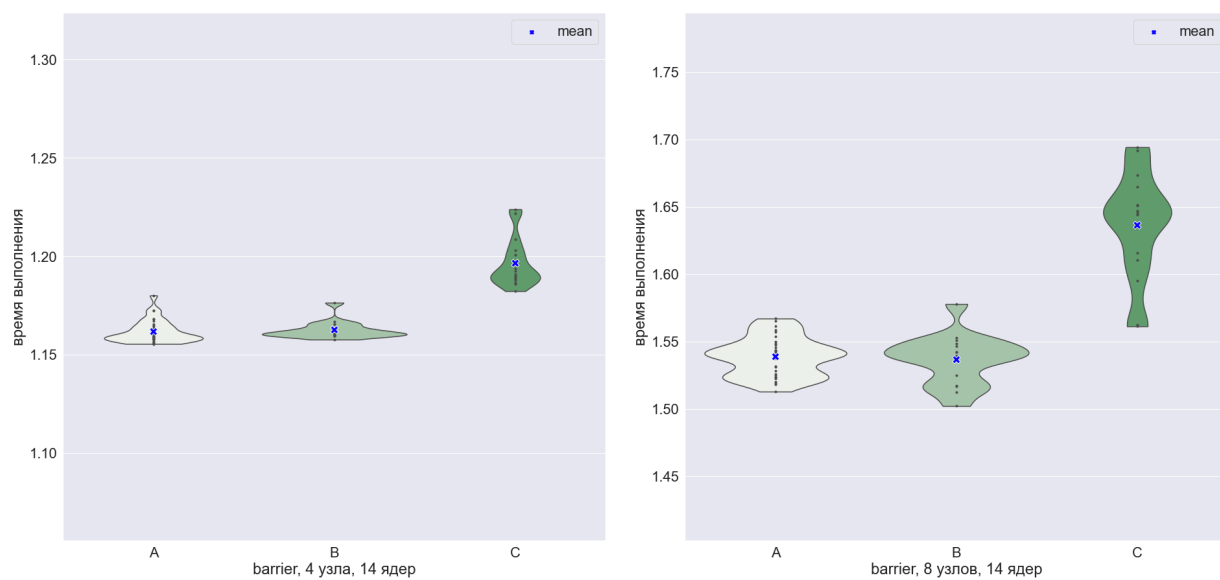


Рис. 9: Barrier 14 ядер на узел, 4 и 8 узлов соответственно.

На рисунках 8 и 9 видно, что детектор с операцией Barrier обнаруживает запуски в случае С и в изначальных и в повторно исследуемых опытах, когда мониторинг работает на том же логическом ядре, что и детектор.

Те же самые исследования с детектором с операцией All-to-All провести не удалось.

7. Ссылки

1. <https://users.parallel.ru>
2. Исходная ВКР (как оформить ссылку?)