



Московский Государственный Университет им. М.В. Ломоносова

Факультет Вычислительной Математики и Кибернетики

Кафедра суперкомпьютеров и квантовой информатики

Худолеева Анна Александровна

Исследование влияния системы мониторинга производительности на выполнение memory-bound инструкций и MPI операций

Научный руководитель:

к.ф.-м.н., с.н.с. НИВЦ МГУ
К.С. Стефанов

Москва 2021

Введение	3
Цель работы	3
Коллективные MPI операции	3
Критерий обнаружения шума системы мониторинга	4
Влияние шума на детектор, число ядер = числу логических ядер	5
Влияние шума на детектор, стандартная постановка задачи	5
Влияние шума на детектор, привязка к ядрам	6
Новый критерий	7
All-to-All, 14 ядер, стандартная частота мониторинга	7
Barrier, 14 ядер, стандартная частота мониторинга	9
Сравнение с другими режимами работы агента системы мониторинга	10
Результаты	11
Memory-bound операции и операции MPI типа точка-точка	11
Выбранный инструмент	11
Описание конфигурации бенчмарка	12
Результаты запусков	13
Короткие неблокирующие сообщения	13
Запуски на 4-х узлах	13
Масштабируемость	14
Влияние системы мониторинга стандартной частоты	15
Пересылка длинных сообщений	16
Запуски на 4-х узлах	16
Выводы	16
Ссылки	17

Введение

Параллельное приложение, которое запускается на СК, может быть исследовано с помощью системы мониторинга производительности. Агент системы мониторинга собирает информацию о состоянии программно-аппаратной среды во время запуска параллельного приложения. Эта информация включает в себя количество кэш-промахов, уровень загрузки процессора, число выполненных операций с плавающей точкой и другие метрики. При этом 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. Влияние системы мониторинга производительности на пользовательские задачи изучено мало.

В дипломной работе в бакалавриате был предложен инструмент — детектор шума, с помощью которого можно обнаружить малый шум и его влияние на пользовательские задачи. Детектор шума основан на выполнении коллективных MPI операций. Было показано, что детектор является чувствительным к небольшому искусственно внедренному шуму, и его можно пробовать использовать для обнаружения реальной системы мониторинга производительности, уровень шума от которой мал.

В представленной работе изучается влияние агента системы мониторинга производительности СК Ломоносов-2 на детектор шума, который основан на выполнении коллективных MPI операций All-to-All и Barrier. Также в работе рассматриваются memory-bound пользовательские задачи, использующие MPI операции типа точка-точка, и влияние шума системы мониторинга производительности на такие задачи.

Цель работы

Целью работы является исследование влияния системы мониторинга производительности суперкомпьютера Ломоносов-2 на сильно синхронизированные программы, содержащие коллективные MPI операций All-to-All и Barrier, и на memory-bound программы, содержащие MPI операций типа точка-точка.

Коллективные MPI операции

В этой части работы рассматривается влияние шума системы мониторинга производительности суперкомпьютера Ломоносов-2 на производительность сильно синхронизированных параллельных программы. Для исследования выбираются программы, содержащие большое количество коллективных MPI операций All-to-All и Barrier. Такие программы моделируются с помощью детектора шума — инструмента,

который был разработан и исследован в рамках проведения ВКР. В основе детектора шума лежит цикл, содержащий MPI операции All-to-All или Barrier. В ВКР было показано, что такой детектор является чувствительным к небольшому искусственному шуму.

Будем исследовать чувствительность детектора к шуму реальной системы мониторинга производительности СК. На СК Ломоносов-2 установлена система мониторинга производительности dimmon. Стандартная частота сбора данных для агента этой системы — 1 раз в 1 секунду, 1 Гц. При стандартной частоте сбора данных агент системы мониторинга dimmon занимает до 0,3% процессорного времени. То есть при стандартной частоте работы шум, создаваемый системой мониторинга, можно считать малым.

Критерий обнаружения шума системы мониторинга

На длинных запусках детектора (продолжительность работы от 10 до 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:

$$CI_{99\%} t_{no-mon} < CI_{99\%} t_{1Hz},$$

t_{no-mon} - выборка значений времени запуска детектора без мониторинга,

t_{1Hz} - выборка значений времени запуска детектора с мониторингом 1 Гц.

Расстояние между доверительными интервалами будем называть *boundary gap*. Влияние детектора может быть обнаружено статистически, если *boundary gap* имеет положительное значение.

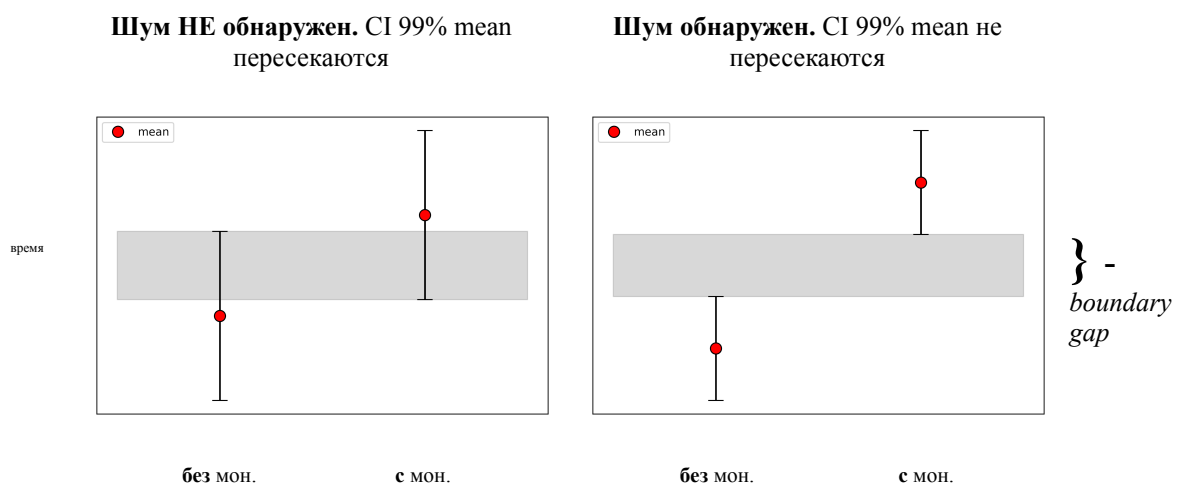


Рис. 1: Варианты расположения доверительных интервалов

Влияние шума на детектор, число ядер = числу логических ядер

Здесь и далее значения времени запусков детектора нормированы по среднему значению времени запусков детектора без системы мониторинга.

■ - разница между средними значениями времени работы детектора с и без мониторинга. Если значение положительное, значит шум замедляет детектор.

■ - boundary gap. Если значение положительное, считаем, что детектор обнаруживает шум 1 Гц.

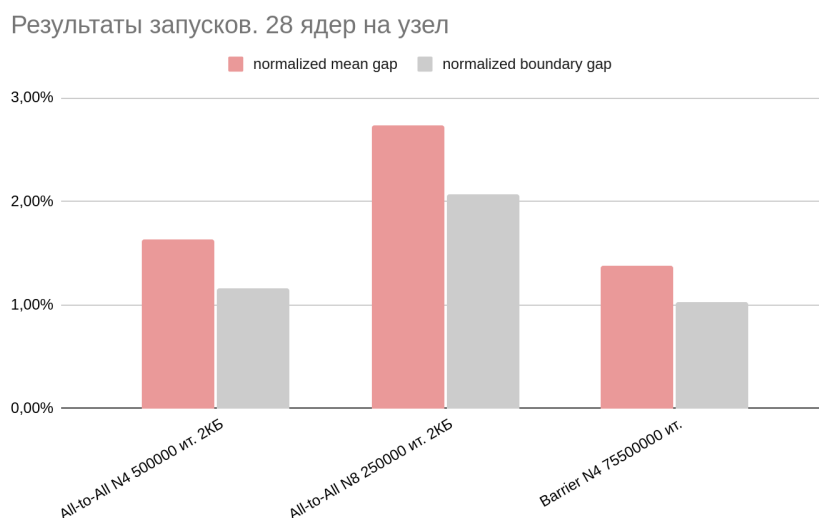


Рис. 2: Результаты запусков детектора. 28 ядер на узел

На рис. 2 приведены результаты запусков детекторов с операцией All-to-All и Barrier на 4-х и 8-ми узлах СК. Детектор запущен на 28 ядрах каждого узла (на всех логических ядрах), мониторинг на одном ядре узла. Значение, отмеченное серым цветом - *boundary gap* — во всех трех случаях положительное. Это означает, что детектор обнаруживает агент системы мониторинга, работающий в стандартном режиме с частотой 1 Гц, так как доверительные интервалы не пересекаются.

Влияние шума на детектор, стандартная постановка задачи

После того, как было показано, что время работы детектора, использующего все логические ядра узла, увеличивается от шума агента системы dimmon, было решено проверить, можно ли обнаружить систему мониторинга, если детектор запущен стандартно, когда число процессов равно числу физических ядер, каждый процесс работает на отдельном ядре. Так запускается большинство приложений на СК, это стандартный способ постановки задачи.

На рис. 3 представлены результаты запусков. На графиках усы — это 99% доверительный интервалы среднего значения времени запуска детектора. 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.

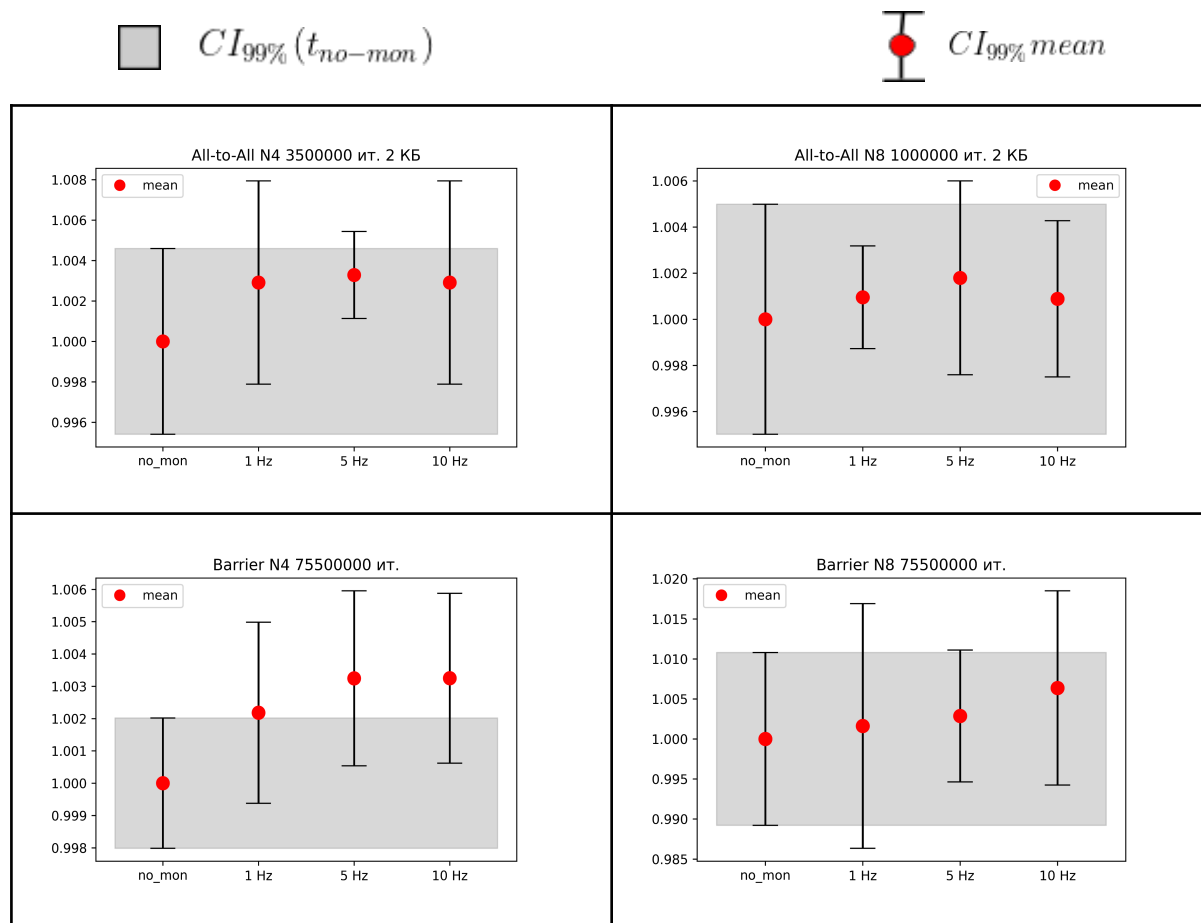


Рис. 3: Результаты запусков детектора. Детектор использует 14 ядер на узел

Пересечение доверительных интервалов, связано с разбросанностью значений времени работы детектора, которая в этих запусках появилась из-за недетерминированности вычислений — ОС при наличии свободных логических ядер по-разному планирует вычисления в разных запусках детектора. Для уточнения результатов экспериментов, попробуем установить привязку детектора и агента системы мониторинга к ядрам.

Влияние шума на детектор, привязка к ядрам

Далее 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 целях накопления статистики, разделим большой цикл детектора, и будем рассматривать время работы пакетов из коллективных операций. Выберем размер пакета так, чтобы время его работы было порядка 10-20-ти секунд.

Новый критерий

Объект исследования изменился — для накопления статистики рассматривается время меньшего числа операций. Длина доверительного интервала в таком случае составляет сотые доли секунды, и не покрывает большинство полученных значений времени работы пакета из коллективных операций. Поэтому здесь и далее 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.

Значение является выбросом, если оно лежит вне интервала

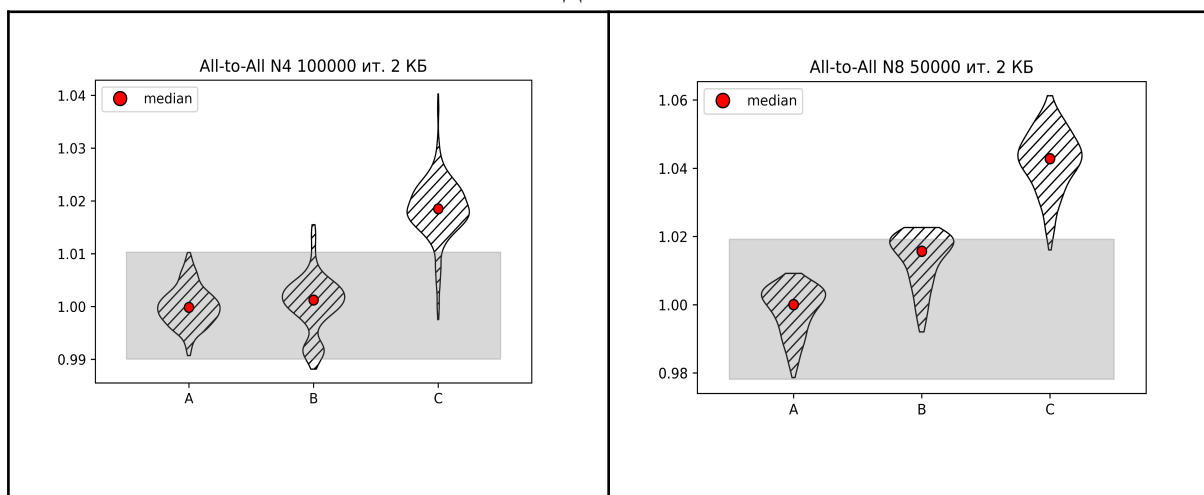
$[Q_{25} - 1.5 IQR, Q_{75} + 1.5 IQR]$, построенного для времени запусков детектора без системы мониторинга.

All-to-All, 14 ядер, стандартная частота мониторинга

Здесь и далее на рисунках значения времени запусков детектора нормированы по медиане времени запусков детектора без системы мониторинга.

На рис. 4 представлены результаты экспериментов, в которых были рассмотрены случаи А, В и С. Графики на рисунке соответствуют конфигурациям детектора с разным числом узлов.

■ $[Q_{25} - 1.5 IQR, Q_{75} + 1.5 IQR]$ для А



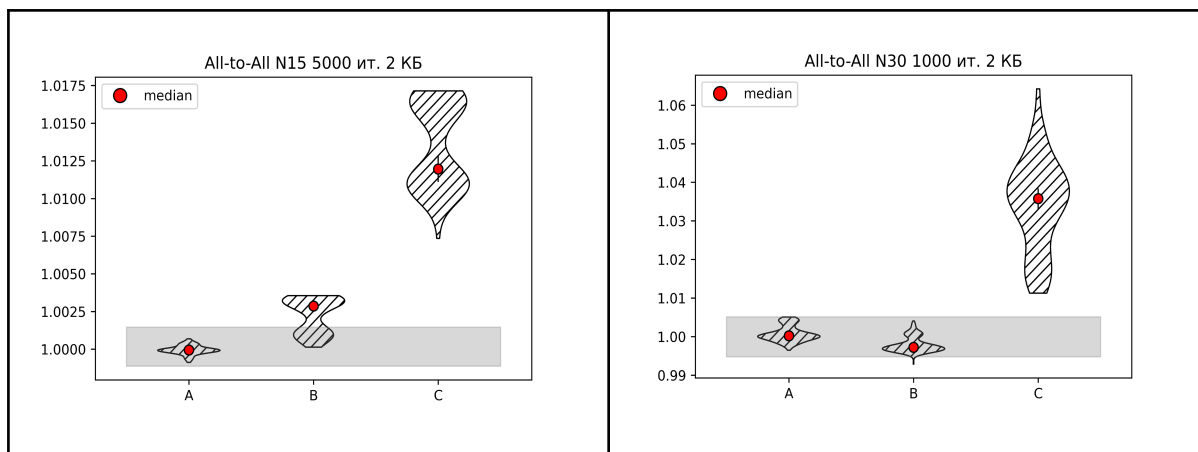


Рис. 4: All-to-All. 14 ядер

Видно, что в случае привязки детектора и мониторинга к одному ядру (случай С), время работы All-to-All увеличивается, и значения времени для варианта С являются выбросами по отношению к времени работы детектора без мониторинга. В табл. 1 приведено значение, на которое увеличивается время работы цикла для всех конфигураций детектора в случае С.

Таблица 1: Увеличение времени работы детектора All-to-All в случае С относительно времени работы без системы мониторинга

число узлов	$median\ C - median\ A$, секунды	$median\ A$, секунды
N4	0.4	22.6
N8	1.1	25.8
N15	0.2	19.1
N30	0.4	11.7

Для случая В, когда детектор и мониторинг не пересекаются, детектор не обнаруживает шум системы мониторинга на 4-х, 8-ми, 30-ти узлах СК. Для 8-ми узлов наблюдается незначительное увеличение времени работы детектора. А на 15 узлах, где шум обнаружен, замедление составляет всего 0,05 секунды.


Таблица 2: Наиболее чувствительная конфигурация All-to-All. Увеличение времени работы детектора All-to-All в случае В, 15 узлов, относительно времени работы без системы мониторинга.

число узлов	$median\ B - median\ A$, секунды	$median\ A$, секунды
N15	0.05	19.1

Barrier, 14 ядер, стандартная частота мониторинга

На рис. 5 видно, что детектор с операцией Barrier обнаруживает запуски в случае С, когда мониторинг работает на том же логическом ядре, что и детектор. В табл. 3 приведено значение, на которое увеличивается время работы цикла из операций Barrier для всех конфигураций детектора в случае С.

Для случая В шум не изменяет времени работы детектора. Наибольшее увеличение времени работы детектора, когда мониторинг привязан к другому ядру, составляет 0,04 секунды на 8 узлах — это изменение времени можно считать незначительным.

 $[Q_{25} - 1.5 IQR, Q_{75} + 1.5 IQR]$ для А

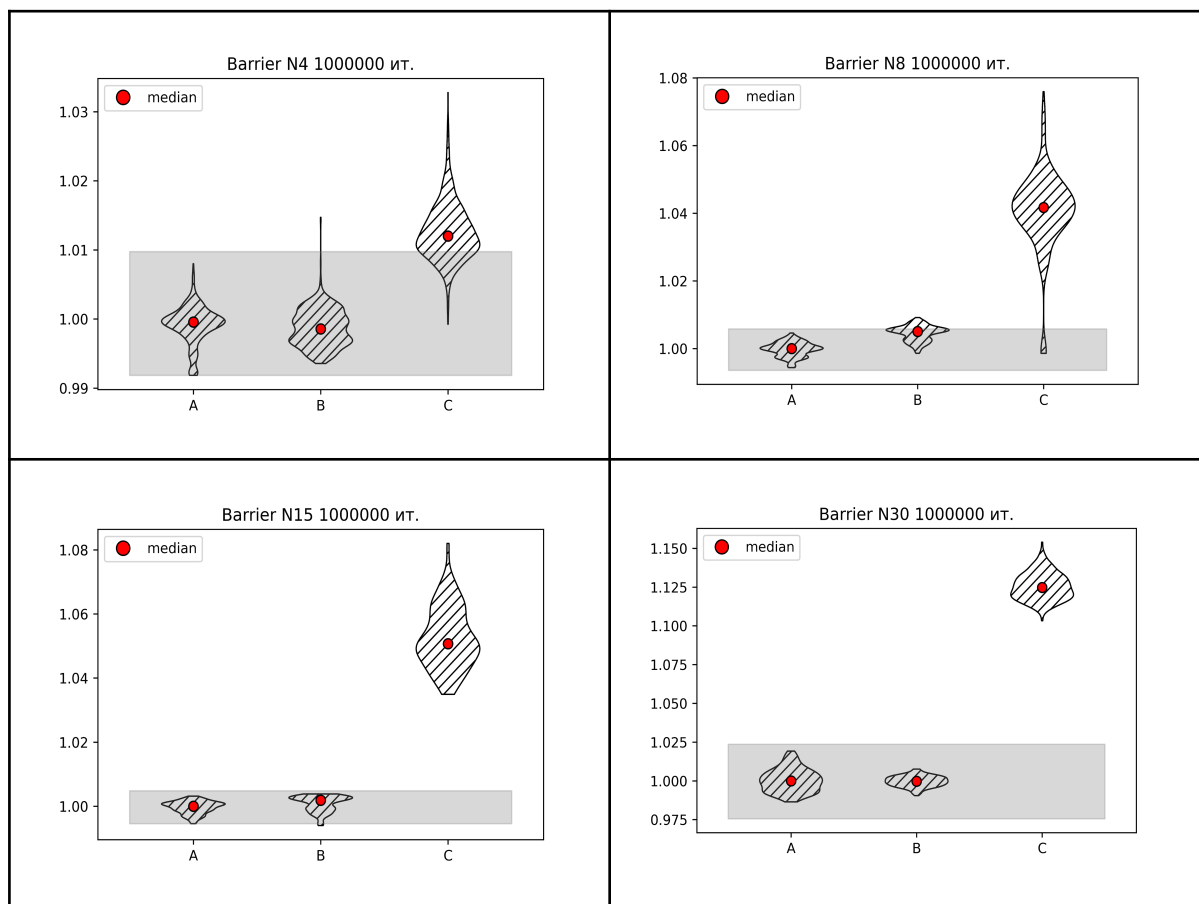


Рис. 5: Barrier 14 ядер

Таблица 3: Увеличение времени работы детектора Barrier в случае С относительно времени работы без системы мониторинга

	<i>median C - median A, секунды</i>	<i>median A, секунды</i>
N4	0.08	6.7
N8	0.3	8.2
N15	0.6	12

N30	1.1	25.8
-----	-----	------

Таблица 4: Наиболее чувствительная конфигурация Barrier. Увеличение времени работы детектора Barrier в случае В, 8 узлов, относительно времени работы без системы мониторинга.

	<i>median B - median A, секунды</i>	<i>median A, секунды</i>
N8	0.04	8.2

Сравнение с другими режимами работы агента системы мониторинга

Здесь рассматривается конфигурация запуска, когда логические номера ядер, на которых работает детектор и агент системы мониторинга, не совпадают (В).

Было показано, что влияние работы агента системы мониторинга с частотой 1 Гц не обнаруживается детектором, то есть влияние агента, работающего в стандартном режиме на отдельном ядре, на коллективные операции является незначительным. Было решено проверить, на сколько можно повысить частоту сбора данных агентом мониторинга, так чтобы его влияние осталось незначительным? Какая частота не ухудшит производительность коллективных MPI операций?

Для конфигураций, при которых наблюдалось увеличение времени работы детектора с мониторингом, расположенном на другом ядре, по сравнению с отсутствием мониторинга, было проведено сравнение режимов работы системы мониторинга. Сравниваются режимы с частотой сбора данных 1 Гц (В), 10 Гц (D) и 20 Гц (E). Рассматривается работа детектора на 8-ми узлах. Результаты запусков приведены на рис. 6.

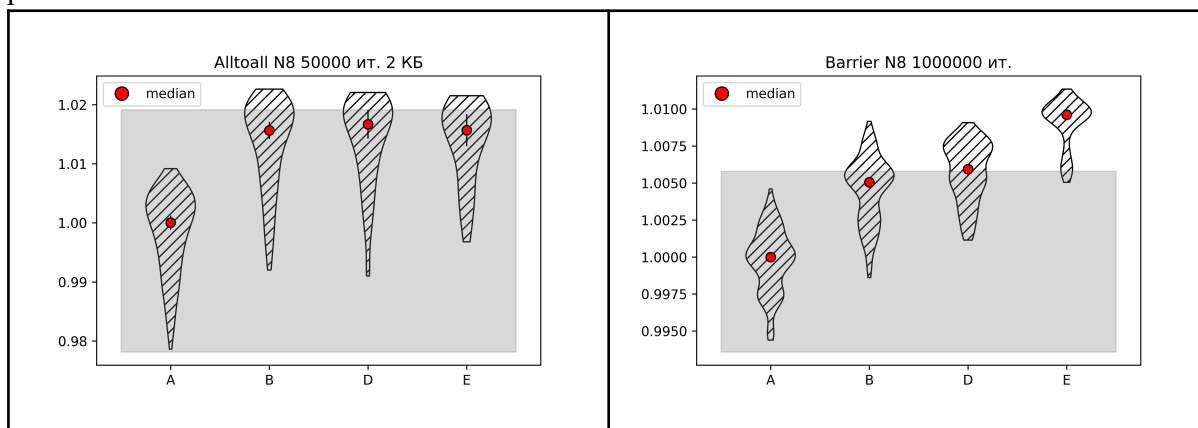


Рисунок 6: Сравнение влияния разных режимов работы агента системы мониторинга

Оказалось, что влияние на операцию All-to-All всех трех режимов работы агента системы мониторинга является неразличимым. Но для операции Barrier это верно только для режимов 1 Гц и 10 Гц. Детектором обнаруживается влияние мониторинга при 20 Гц. Но, стоит заметить, что разница между медианами времени работы

детектора без мониторинга и с мониторингом 20 Гц составляет 0,08 секунды, что в 3,75 раза меньше, чем влияние агента мониторинга, запущенного на том же ядре, что и детектор, с частотой 1 Гц.

Результаты

Были рассмотрены два варианта запуска агента системы мониторинга — его выполнение происходит на том же логическом ядре, что и выполнение процесса детектора, или на другом логическом ядре.

Было установлено, что в первом случае присутствие агента системы мониторинга заметно, система мониторинга может внести заметное замедление в выполнение коллективных операций, при их очень большом количестве.

Если детектор работает на другом логическом ядре, то влияние на коллективные операции является незначительным. В такой конфигурации, режимы работы системы мониторинга 1 Гц и 10 Гц одинаково “не влияют” на обе рассмотренные коллективные операции All-to-All и Barrier.

Memory-bound операции и операции MPI типа точка-точка

В этом разделе приводятся результаты исследования влияния системы мониторинга производительности на MPI операции точка-точка и на memory-bound инструкции. Далее приведено описание бенчмарка, с помощью которого проводится исследование, а также результаты запусков бенчмарка на СК Ломоносов-2.

Выбранный инструмент

Было решено рассматривать программы, в которых memory-bound операции и MPI операции типа точка-точка используются совместно. В качестве memory-bound нагрузки была взята распределенная MPI версия бенчмарка STREAM triad [1]. В ядре бенчмарка суммируются элементы двух массивов типа double, результат суммирования записывается в третий массив: $a(i) = b(i) + q * c(i)$. После выполнения нескольких итераций бенчмарка STREAM triad процессы обмениваются сообщениями по кольцу в одном направлении: процесс i отправляет сообщение процессу $i + 1$. На рисунке 7 представлена схема реализации программного инструмента, с помощью которого исследуется влияние системы мониторинга производительности.

```

do over measurements_count {
    start timer;

    // mssg_cnt - число сообщений
    send to myrank+1 mssg_cnt messages;
    do over stream_count
        a[i] = b[i] + q * c[i];
    recv from myrank-1 mssg_cnt messages;

    finish timer;
}

```

Рисунок 7: схема используемого инструмента

Описание конфигурации бенчмарка

На каждом узле СК Ломоносов-2 установлен Intel Haswell-EP E5-2697v3 процессор с 14 ядрами. Размер кэша — 35 MB [2].

Размер одного массива должен в 4 раза превышать размер кэш памяти процессора [1]. При запуске STREAM на нескольких узлах, на нескольких процессах одного узла для вычисления минимального значения величины массива размеры кэшей узлов суммируются. Для запуска STREAM на СК Ломоносов-2 на n узлах суммарный размер массива должен быть не меньше, чем $(4 * n * 35 MB) / (8 B)$, 8 B - размер типа double.

Таблица 5: Минимальный размер массива бенчмарка STREAM triad при разном числе узлов.

1 узел	4 узла	8 узлов	16 узлов
18×10^6 эл.	70×10^6 эл.	140×10^6 эл.	280×10^6 эл.

Было показано, что коллективные операции All-to-All и Barrier являются чувствительными к системе мониторинга производительности, если агент системы мониторинга запущен на тех же логических ядрах, что и программа с коллективными операциями. В частности, если коллективные операции запущены на всех логических ядрах узлах (на 28 ядрах), а агент системы мониторинга на одном из логических ядер того же узла, его влияние будет значительным. Поэтому предлагается начать измерение влияния системы мониторинга производительности на программный инструмент с запуска инструмента на всех логических ядрах процессора.

Для того, чтобы определить оказывает ли влияние система мониторинга, увеличим частоту сбора данных агента этой системы до 10 Гц. В случае, если влияние агента системы мониторинга с частотой 10 Гц статистически обнаружено, можно уменьшить частоту системы мониторинга до стандартной и изучить влияние системы мониторинга в стандартной конфигурации запуска.

В выбранной конфигурации агент системы мониторинга собирает данные каждые 10^{-1} с. В табл. 6 приведена медиана времени запуска 1 операции STREAM triad на 1 и 4 узлах СК, на 28 ядрах каждого узла. При значении числа элементов массива $n \times 250 \times 10^6$, где n — число узлов, время работы 1 ядра STREAM превышает 10^{-1} с, значит, выполнение ядра STREAM пересекается с работой агента системы мониторинга. Зафиксируем подобное значение числа элементов массива.

Таблица 6: соответствие времени работы STREAM triad и числа элементов 1-го массива.

кол-во узлов	1 узел	4 узла
время STREAM triad	0,136 с	0,137 с
кол-во эл. 1-го массива	250×10^6	1000×10^6

Для анализа полученных результатов запуска бенчмарка на СК будем использовать статистический критерий, основанный на пересечении времени запуска бенчмарка, запущенного в присутствии шума системы мониторинга производительности, с интервалом $[Q_{25} - 1.5 IQR, Q_{75} + 1.5 IQR]$ времени работы бенчмарка без системы мониторинга производительности. (См. раздел *Новый критерий*)

Результаты запусков

Короткие неблокирующие сообщения

Запуски на 4-х узлах

В экспериментах было решено измерять время выполнения пакетов из 100 операций STREAM triad. Выбран малый размер сообщения — 32 В. На рисунке 8 и в таблице 7 приведены результаты экспериментов. Наиболее заметное замедление в рассмотренных конфигурациях наблюдается, когда к циклу из 100 операций STREAM triad добавляется 20 пересылок MPI типа точка-точка. Эта конфигурация является лучшей из выбранных и единственной, для которой выполняется критерий обнаружения шума. Чувствительность бенчмарка к шуму при переходе от 20 пересылок к 1000000 пересылок ухудшается. Время 1000000 неблокирующих пересылок сообщения длины 32 В в отсутствие ядра STREAM triad превышает 10^{-1} с. Выполнение такого количества пересылок пересекается выполнением агента системы мониторинга. Однако это не влияет на чувствительность рассмотренной конфигурации. Можно сказать, что время работы рассмотренной конфигурации детектора не зависит от неблокирующих сообщений. При этом время работы ядра STREAM triad замедляется на 0,32% под влиянием системы мониторинга производительности с частотой сбора данных 10 Гц.

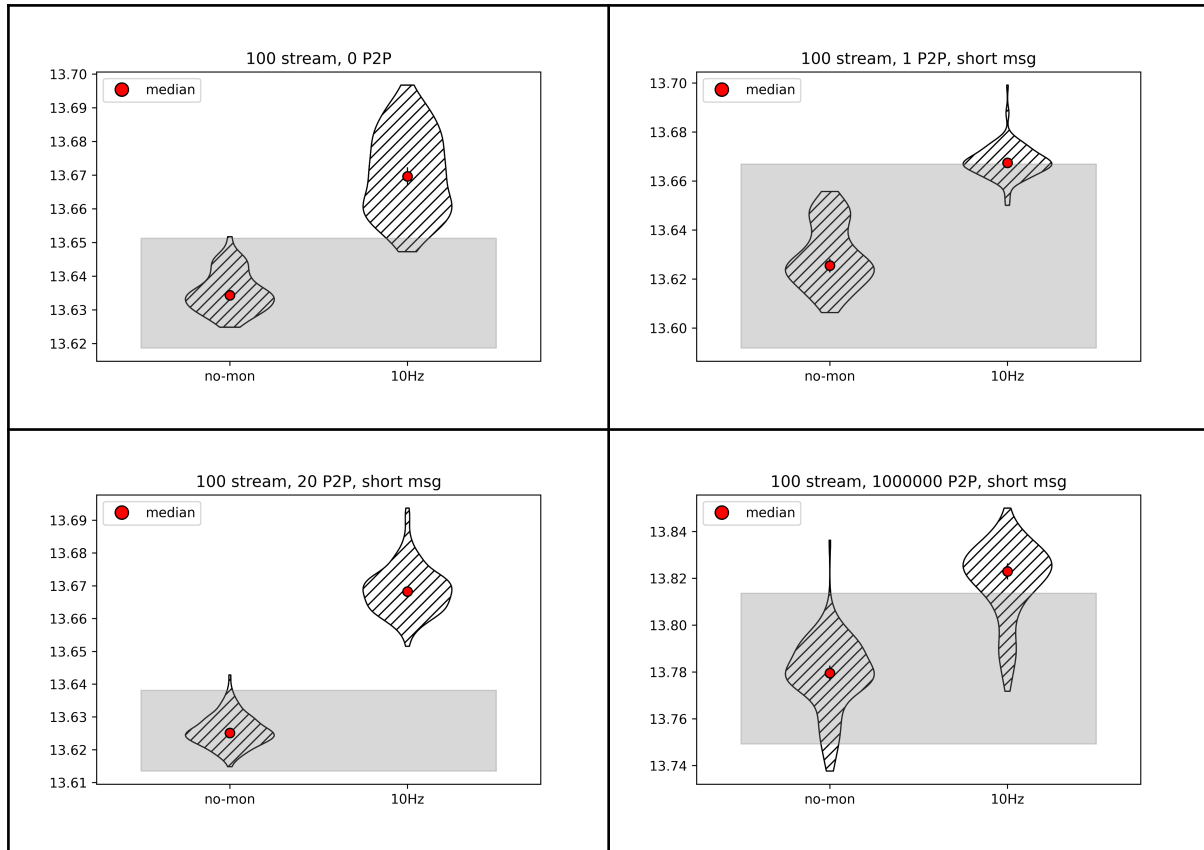


Рисунок 8: Сравнение времени работы бенчмарка без системы мониторинга и в присутствии агента системы мониторинга с частотой 10 ГГц на 4 узлах СК. 100 stream - число измеряемых stream операций, x P2P - количество пересылаемых сообщений.

Таблица 7: Замедление в % - $[median(T_{10Hz}) - median(T_{no-mon})] / median(T_{no-mon})$

0 P2P - 0,25%	1 P2P - 0,31%
20 P2P - 0,32%	1000000 P2P - 0,32%

Масштабируемость

На рисунке 9 приведено сравнение результатов запуска бенчмарка в конфигурации 100 операций STREAM triad + 20 сообщений на 4-х, 8-ми и 15 узлах СК. При переходе от 4-х узлов к 8-ми замедление бенчмарка под влиянием системы мониторинга 10 ГГц сохраняется, критерий обнаружения влияния системы мониторинга выполняется. На 15-ти узлах СК время запусков с и без системы мониторинга статистически неразличимы, формы двух распределений схожи между собой.

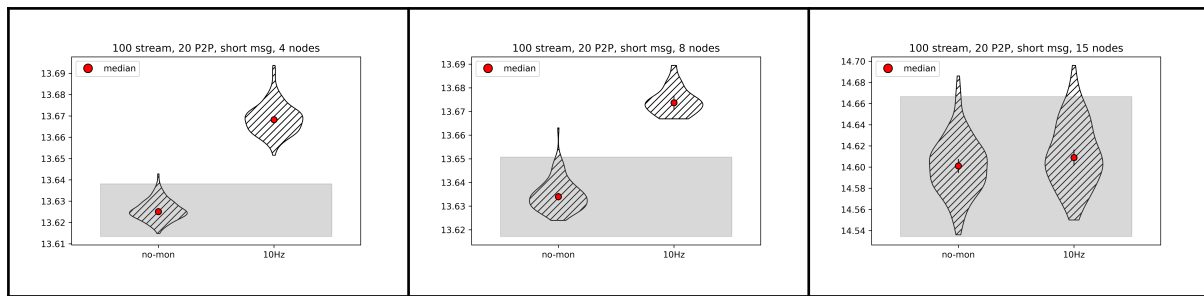


Рисунок 9: Сравнение времени работы бенчмарка без системы мониторинга и в присутствии агента системы мониторинга с частотой 10 Гц на 4, 8, 15 узлах СК при длине сообщения 32 В. 100 stream - число измеряемых stream операций, 20 P2P - количество пересылаемых сообщений.

Влияние системы мониторинга стандартной частоты

На рисунке 10 приведено сравнение влияния агента системы мониторинга с частотой 10 Гц и 1 Гц. Как видно, влияние агента системы мониторинга с частотой 1 Гц не фиксируется с помощью критерия обнаружения шума. Однако, часть выборки времени работы бенчмарка с мониторингом частотой 1 Гц расположена за серой областью. Замедление для случая 1 Гц составляет 0,09%, что является малым замедлением.

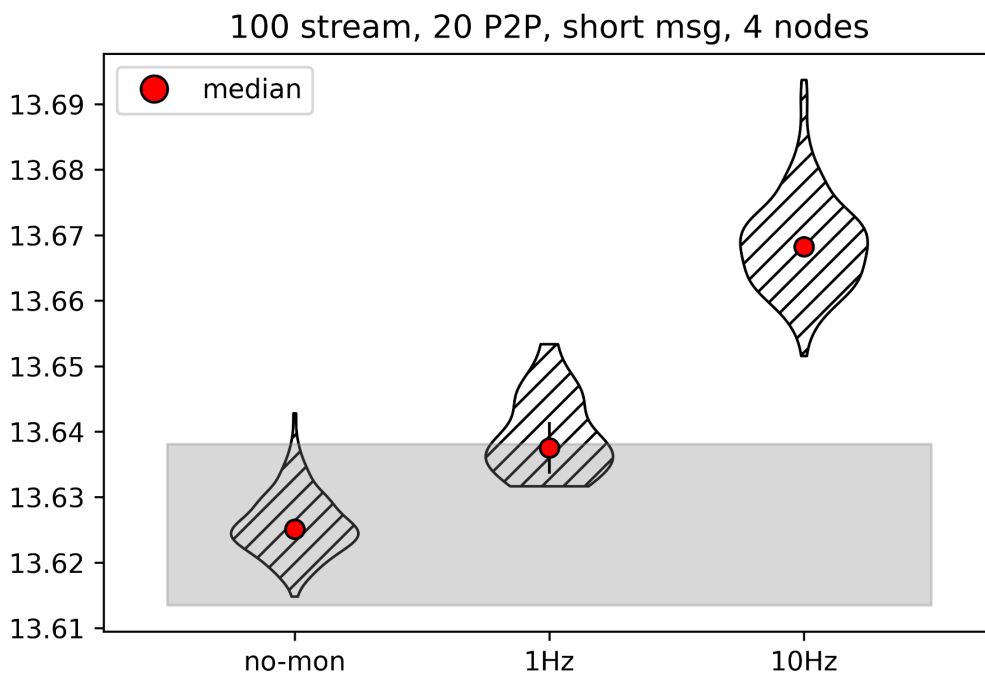


Рисунок 10: Сравнение влияния двух режимов работы системы мониторинга производительности на лучшую конфигурацию с коротким сообщением.

Пересылка длинных сообщений

Запуски на 4-х узлах

Было рассмотрено поведение бенчмарка при пересылки длинных сообщений объемом 5 MB. Число операций STREAM triad оставлено тем же. На рисунке 11 представлены результаты запуска новой конфигурации бенчмарка при разном числе пересылок типа точка-точка. Картинка “100 stream, 0 P2P” перенесено с рисунка 8. Видно, что наибольшее изменение во времени выполнения бенчмарка заметно либо при отсутствии пересылок, либо при наличии 1 пересылки. При большом числе сообщений влияние системы мониторинга производительности незаметно, время выполнение программ одинаковое. Замедление для случая “100 stream, 1 P2P” - 0,84%. Это значение больше, чем для лучшей конфигурации бенчмарка с коротким сообщением.

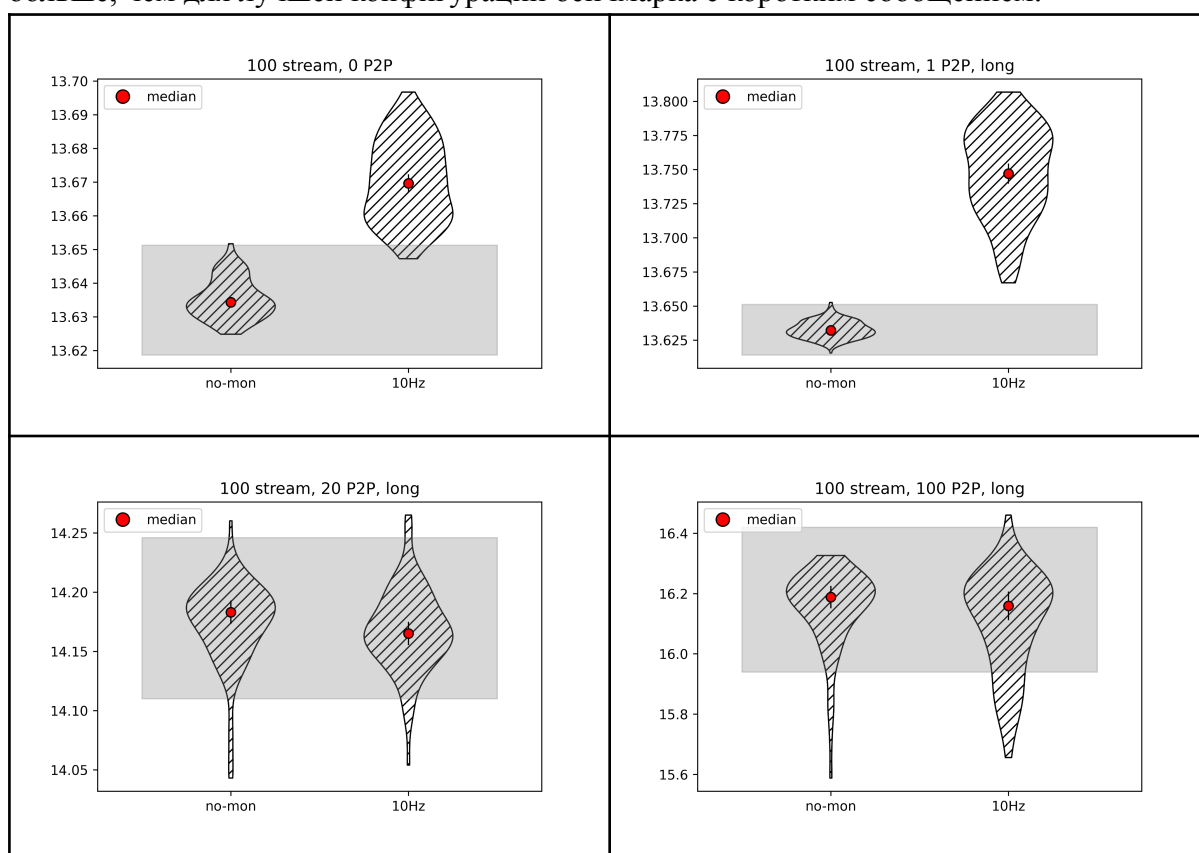


Рисунок 11: Сравнение времени работы бенчмарка без системы мониторинга и в присутствии агента системы мониторинга с частотой 10 ГГц на 4 узлах СК при длине сообщения 5 MB. 100 stream - число измеряемых stream операций, x P2P - количество пересылаемых сообщений.

Выводы

В работе рассмотрено влияние шума системы мониторинга dimmon на выполнение коллективных MPI операции All-to-All и Barrier. Выделены два разных случая взаимного расположения пользовательской программы и агента системы мониторинга производительности: случай В, когда выполнение пользовательской программы и

агента системы мониторинга производительности происходит на логических ядрах с разными номерами, и случай С, когда выполнение процессов происходит на логических ядрах с одинаковыми номерами. Было показано, что в случае В замедление коллективных MPI операций All-to-All и Barrier в присутствии агента системы мониторинга производительности на том же физическом ядре (но на другом логическом ядре), является статистически незначительным. А в случае С шум системы мониторинга производительности может оказывать заметное влияние на коллективные MPI операции и замедлять их выполнение до 3%. Таким образом, можно сказать, что на пользовательские приложения, в которых используются коллективные MPI операции All-to-All и Barrier, и которые поставлены на счет на СК стандартным способом (когда число процессоров равно числу физических ядер) шум системы мониторинга производительности со стандартной частотой сбора данных 1 Гц не оказывает влияния.

Также в работе рассмотрено влияние агента системы мониторинга производительности с частотой сбора данных 10 Гц на выполнение программы, содержащей ядро бенчмарка STREAM triad и неблокирующие пересылки MPI типа точка-точка. Рассматривается запуск бенчмарка и агента системы мониторинга на всех логических ядрах узла. Обнаружено, что добавление неблокирующих пересылок малых сообщений незначительно влияет на чувствительность программы к шуму. Влияние системы мониторинга на memory-bound операции еще предстоит установить. На данный момент известно, что шум системы мониторинга может быть статистически обнаружен одной конфигурацией бенчмарка с коротким сообщением. Фиксируемое замедление составляет всего 0,32% от времени работы этой конфигурации без системы мониторинга производительности. Для сравнения, замедление коллективных операций составляет 1-3% при их запуске на всех логических ядрах с системой мониторинга производительности, частота которой равна 1 Гц. Для конфигурации бенчмарка с длинным сообщением замедление 0,8% наблюдается при внедрении 1 пересылки. При увеличении числа пересылок этот эффект пропадает, и время работы программы с и без системы мониторинга становится неразличимым.

Ссылки

1. <http://www.cs.virginia.edu/stream/>
2. <https://ark.intel.com/content/www/ru/ru/ark/products/81059/intel-xeon-processor-e5-2697-v3-35m-cache-2-60-ghz.html>