



НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР
«КУРЧАТОВСКИЙ ИНСТИТУТ»

Отделение
суперкомпьютерных
систем и параллельных
вычислений

Влияние векторизации плоских циклов на энергоэффективность параллельной программы

Яровой А.В., Киселёв Е.А., Баранов А.В., Шабанов Б.М.





Гипотезы исследования

E_C – энергопотребление скалярного случая

E_V – энергопотребление векторного случая

Гипотеза 1. $E_C > E_V$

векторизация уменьшает время вычислений,
следовательно, уменьшается суммарное энергопотребление

Гипотеза 2. $E_C \approx E_V$

для вычислений используется одинаковое количество операций,
а значит энергопотребление не должно меняться

Гипотеза 3. $E_C < E_V$

векторизация является сложным механизмом,
который интенсивнее нагружает ядра процессора



Результаты НИЦ КИ

- Savin G., Shabanov B., Rybakov A., Shumilin S. Vectorization of flat loops of arbitrary structure using instructions AVX-512. // Lobachevskii Journal of Mathematics, 2020, Vol. 41, No. 12, P. 2566-2574. DOI: 10.1134/S1995080220120331
- Е.А. Киселёв, А.В. Баранов, О.С. Аладышев, П.Н. Телегин, А.В. Яровой. Программные инструменты энергоэффективного планирования суперкомпьютерных заданий // Труды НИИСИ РАН, 2021. Вып. 11, № 4(48). С. 48-55. DOI: 10.25682/NIISI.2021.4.0005
- Киселёв Е.А., Чубаров Д.А., Баранов А.В. Оценка влияния различных участков программного кода на энергопотребление вычислительной системы // Труды НИИСИ РАН, 2024. Т. 14, № 4. С. 67-74
- Kiselev, E.A., Yarovoy, A.V., Telegin, P.N. et al. Scheduling Supercomputer Jobs under Existing Power Consumption Constraints. Lobachevskii J Math 45, 5082–5091 (2024). <https://doi.org/10.1134/S199508022460612X>

Предложены методы автоматической трансформации графа потока управления программы для векторизации **плоских циклов** – циклов без вложенности, содержащих сложную структуру ветвлений, за счёт чего удаётся достигнуть сокращения непроизводительных вычислительных операций (*методы Рыбакова*)



Методика проведения эксперимента

- 2 варианта запуска задания¹:
 1. распараллеливание вычислений с использованием OpenMP (*скалярный вариант*)
 2. дополнительная векторизация программного кода с помощью инструкций AVX-512 (*векторный вариант*)
- контроль с помощью MesStat²:
 - среднего, медианного и пикового энергопотребления узла
 - совокупного энергопотребления узла за время выполнения задания

¹ https://github.com/rumatha/riemann_vec

² https://gitlab.com/mrpravda/messtat_parser

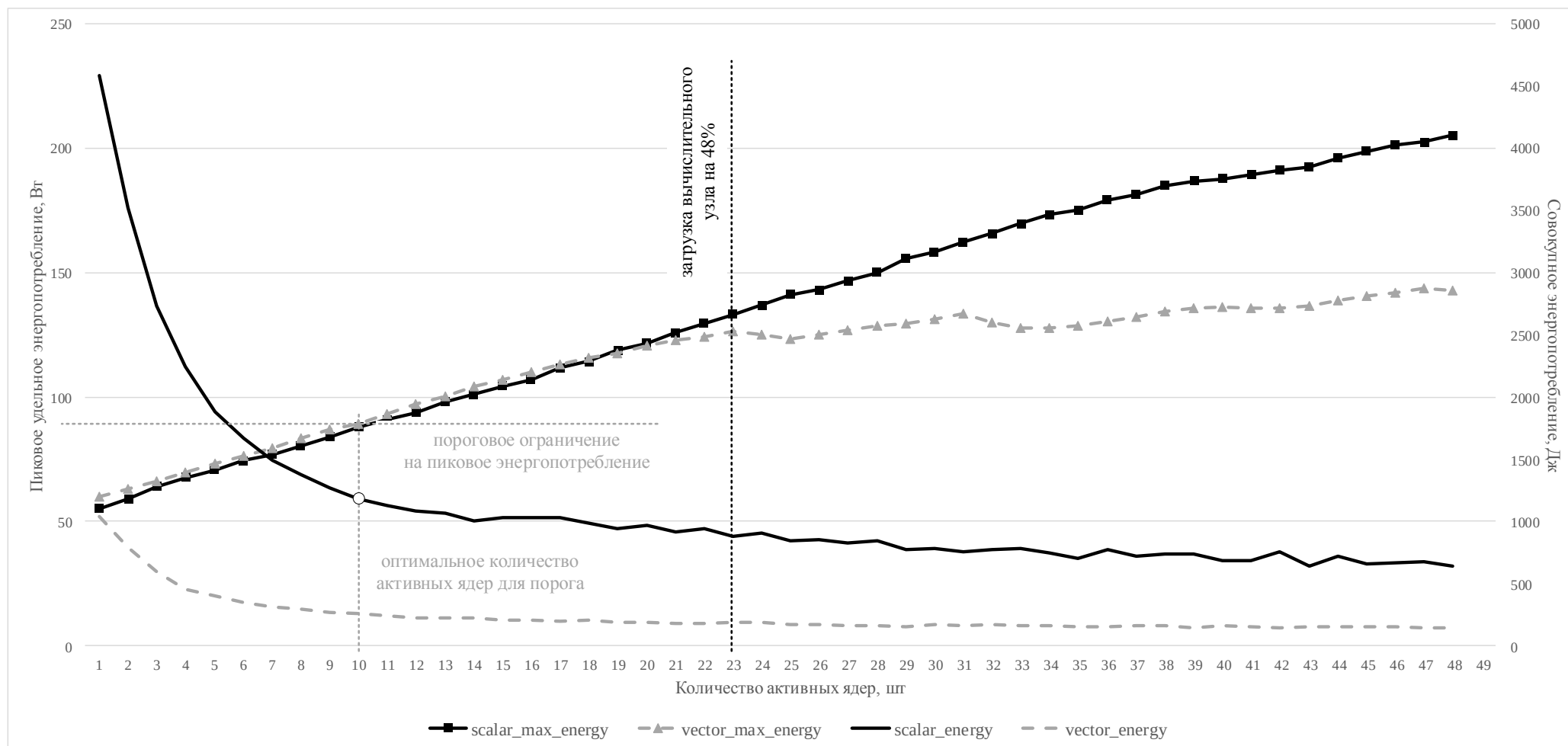


Использованные суперкомпьютерные разделы

Параметр	Узел CSL	Узел KNL	Узел ICELAKE
Микропроцессор	Intel Xeon Gold 6248R, 24 ядра	Intel Xeon Phi 7290, 72 ядра	Intel Xeon Platinum, 32 ядра
Количество микропроцессоров на узле, шт.	2	1	2
Коммуникационная архитектура	Intel Omni-Path		
Архитектура микропроцессора	Cascade Lake	Knight Landing	Ice Lake

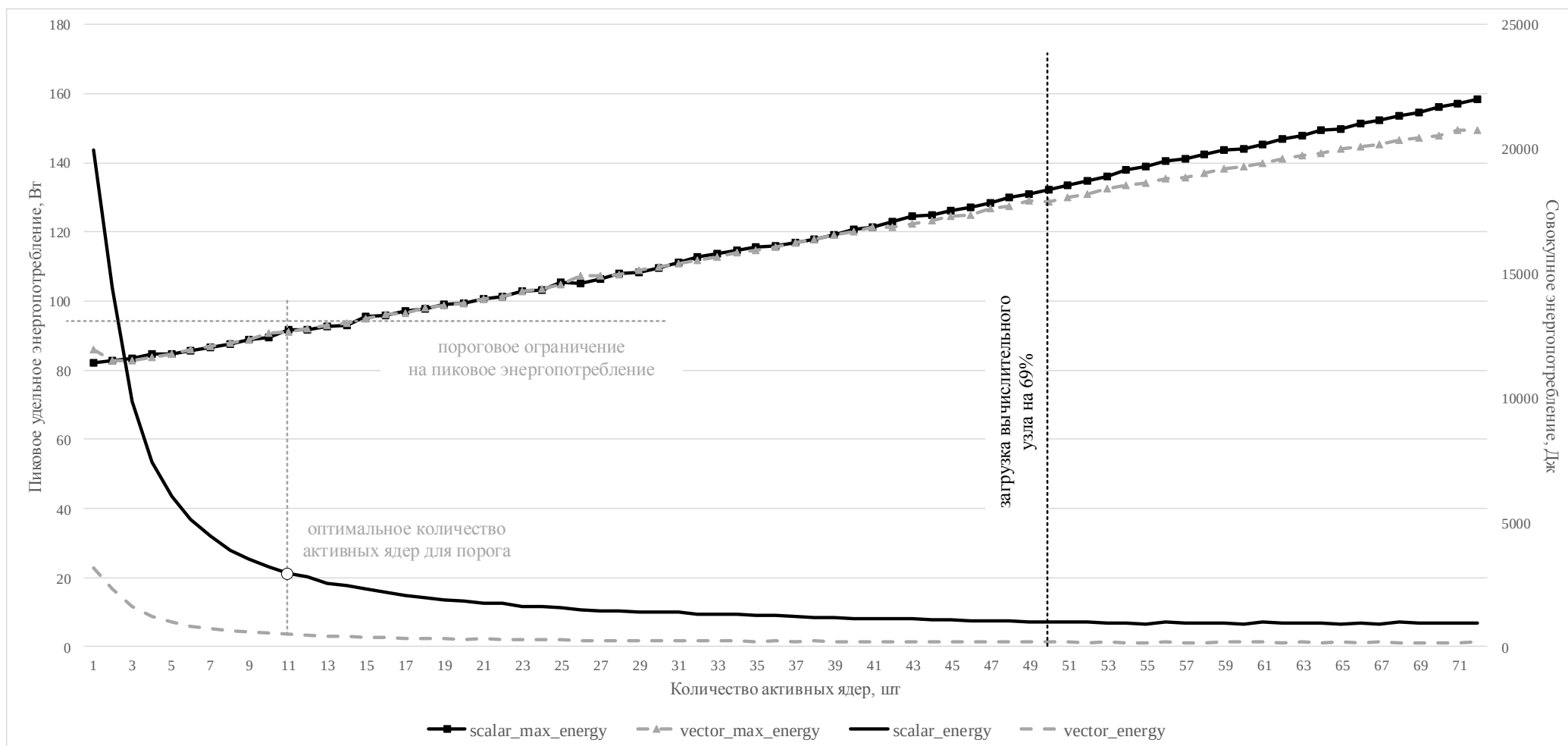


Результаты для CSL



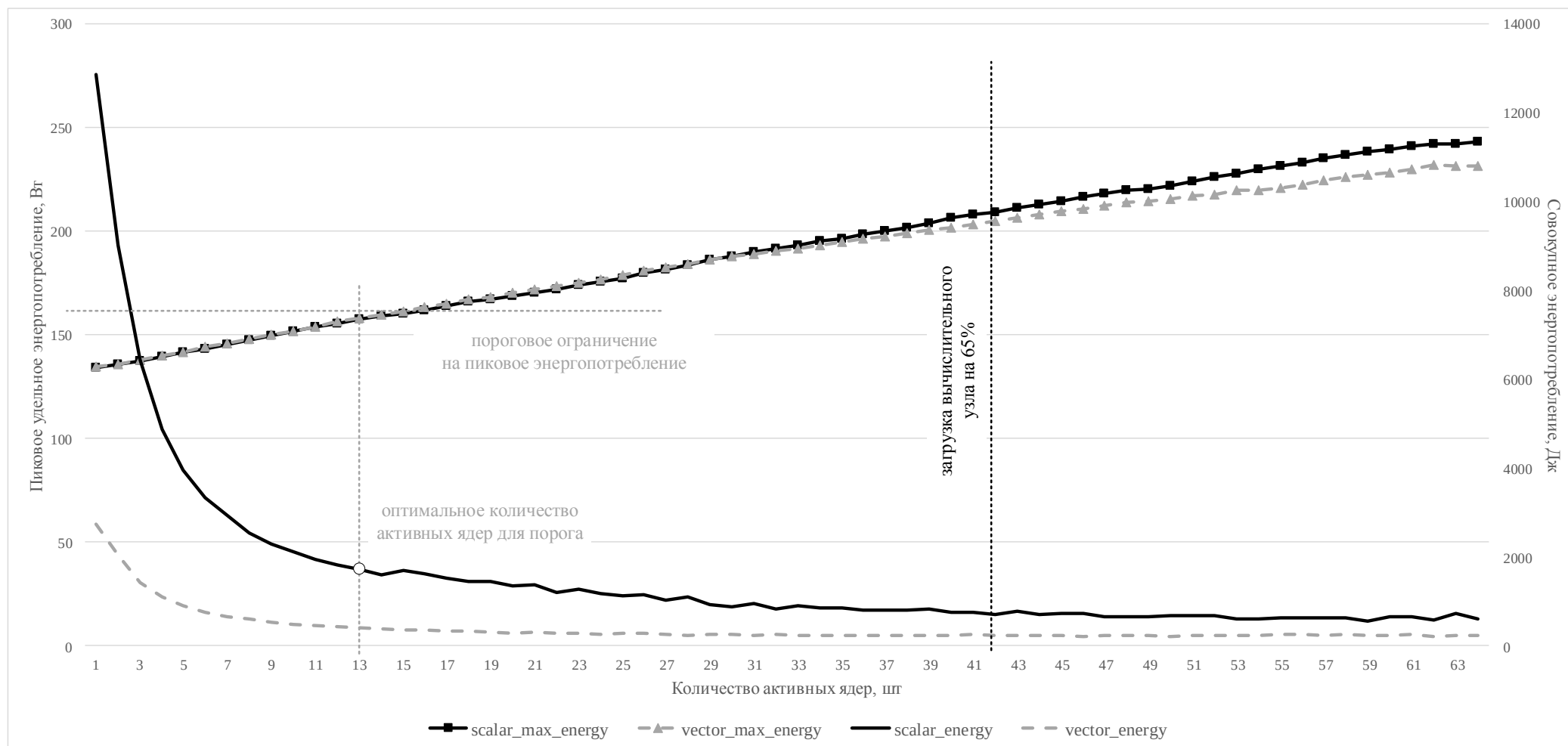


Результаты для KNL





Результаты для ICELAKE





Результаты экспериментов

- при увеличении количества работающих потоков совокупное энергопотребление *уменьшается*
- с некоторого момента значения выходят на энергетическое «плато», после достижения которого значения *уменьшаются незначительно*
- увеличение количества активных ядер *не приводит к статистическому разбросу* средних, медианных и пиковых значений энергопотребления
- с некоторого числа активных ядер (при загрузке вычислительного узла более чем на 65%) наблюдается *меньший рост* пикового энергопотребления



Условия работы метода оптимизации программного кода с учётом его энергоэффективности

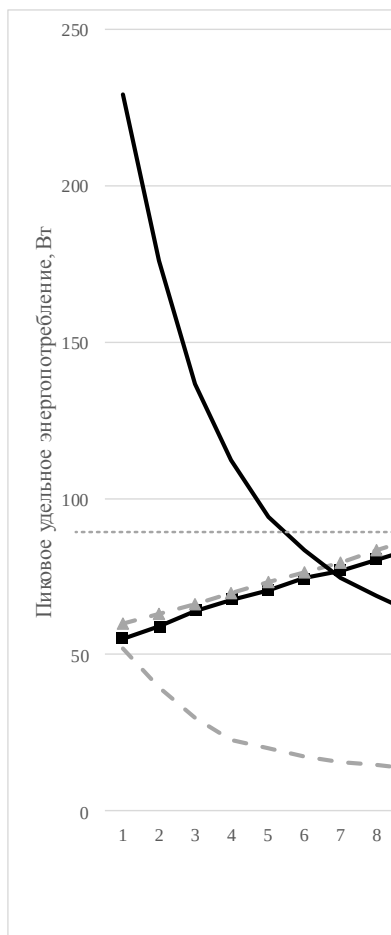
1. Возможна векторизация анализируемого программного кода методами Рыбакова
2. Возможно распараллеливание анализируемого программного кода по общей памяти
3. Возможно применение и векторизации, и распараллеливания по общей памяти (т.е. выполняются одновременно условия 1 и 2)



Метод при выполнении условия 1 и не выполнении условия 2

(возможна векторизация и невозможно распараллеливание)

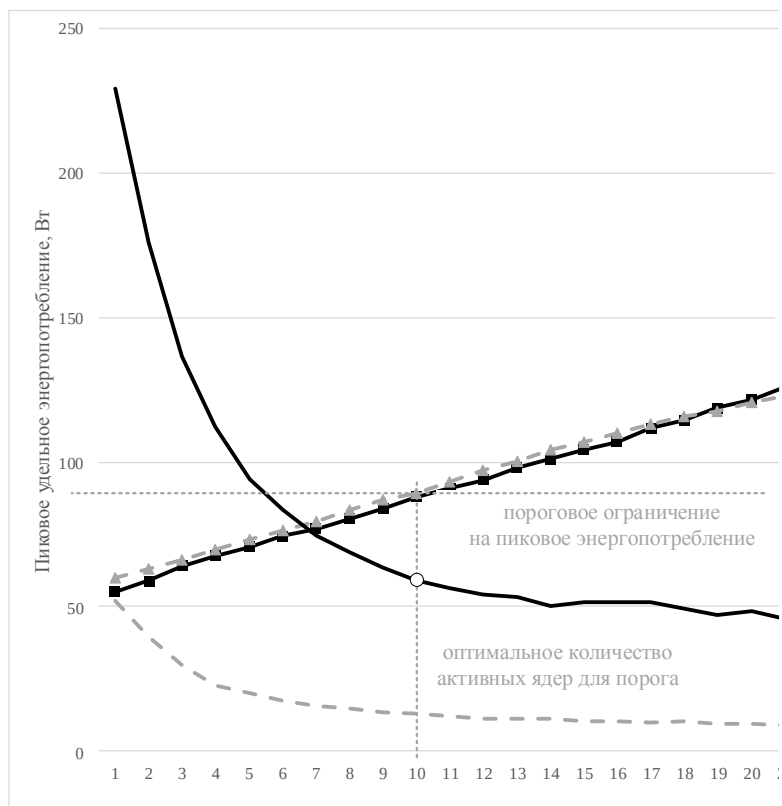
Применение векторизации
на 1 вычислительном узле
для снижения совокупного
энергопотребления в сравнении
с не векторизованным кодом





Метод при выполнении условия 2 и не выполнении условия 1

(возможно распараллеливание и невозможна векторизация)

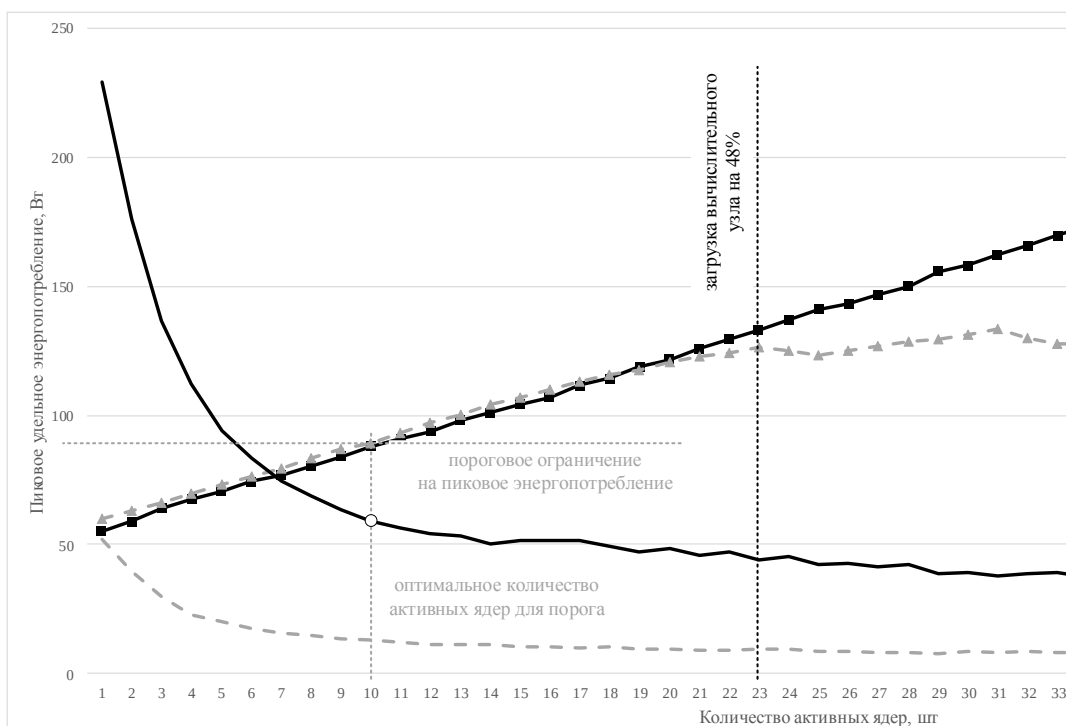


Распараллеливание выполнения программного кода на как можно большем числе ядер вычислительного узла с учётом имеющихся ограничений по пиковому энергопотреблению



Метод при выполнении условий 1 и 2

(возможны распараллеливание и векторизация)



- при наличии пиковых ограничений требуется использовать сценарий 2
- если ограничения менее жёсткие, то рекомендуется распараллеливать вычисления далее до достижения, как минимум, 60% загрузки вычислительных узлов системы



Основные выводы

- применение векторизации существенно улучшает совокупное и пиковое энергопотребление вычислительной системы
- предложенный метод оптимизации программного кода позволяет снизить совокупное энергопотребление вычислительной системы при наличии ограничений на пиковое энергопотребление (в 4,9-14,9 раз) и без таковых ограничений (дополнительно в 4 раза) с гарантией снижения времени выполнения кода
- при наличии возможности загрузить вычислительные узлы, как минимум, на 60% за счёт использования метода удаётся снизить пиковое энергопотребление, как минимум, на 4-5%



НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР
«КУРЧАТОВСКИЙ ИНСТИТУТ»

Отделение
суперкомпьютерных
систем и параллельных
вычислений

Влияние векторизации плоских циклов на энергоэффективность параллельной программы

Яровой А.В., Киселёв Е.А., Баранов А.В., Шабанов Б.М.

