

Возможности использования процессоров Иртыш для научных вычислений

Александр Зубанков
a.zubankov@tramplin.tech

29.09.2026

План доклада

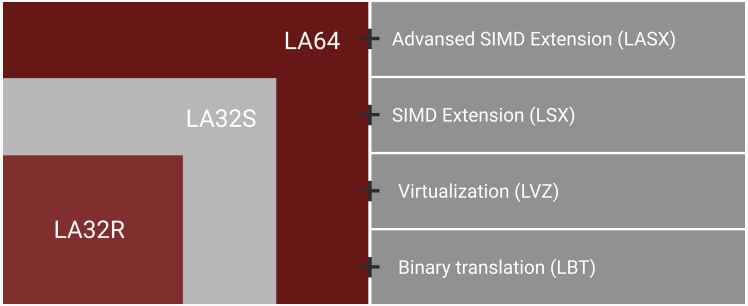
1. Серверные процессоры Иртыш
2. Поддержка архитектуры loongarch64 сообществом
3. Результаты тестирования в сравнении с Эльбрус и Intel Xeon
4. Результаты тестирования серверных процессоров Иртыш в SPEC CPU
5. Прототип HPC кластера с планировщиком slurm на базе процессоров Иртыш
6. Пример запуска научного ПО в области физики высоких энергий
7. Выводы

Серверные процессоры Иртыш

Линейка серверных процессоров Иртыш

	Иртыш С616	Иртыш С632	Иртыш С664
			
Архитектура ядер	LA664		
Ядра / Потоки	16/32	32/64	64/128
Частота, ГГЦ	2.2	2.1	2.0
L1 I/D, KB	64 / 64		
L2 (total) / L3 (shared)	16×256KB / 32 MB	2×16×256KB / 2×32MB	4×16×256KB / 4×32MB
Каналы памяти DDR4 (72-Bit)	4	8	
PCIe 3.0/4.0	64 линии	128 линий	
ISA Extensions	LBT, LVZ, LSX, LASX		
Coherent Link LCL	до 128 потоков	до 256 потоков	

Архитектура LoongArch



- LoongArch —RISC архитектура похожая на MIPS и RISC-V.
- Архитектура LoongArch включает в себя 32 разрядную версию (LA32) и 64 разрядную (LA64).
- 32 разрядная версия делится на упрощенную (LA32R) и стандартную (LA32S).
- Все инструкции имеют фиксированный размер 32 бита и требуют выравнивание по границе в 4 бита.
- Текущая версия архитектуры – 1.1.

Поддержка архитектуры loongarch64 сообществом

Дистрибутивы

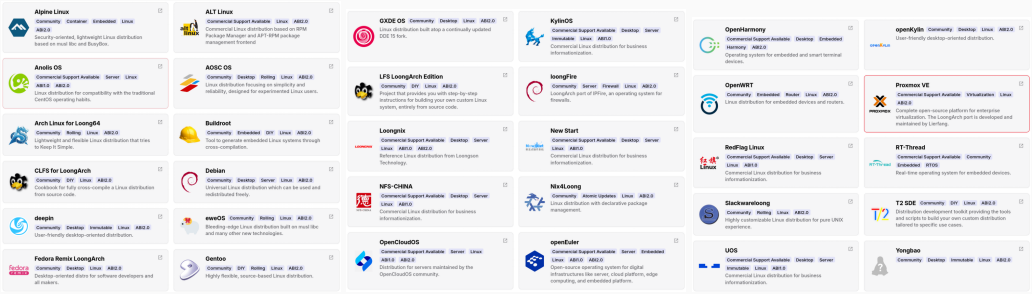
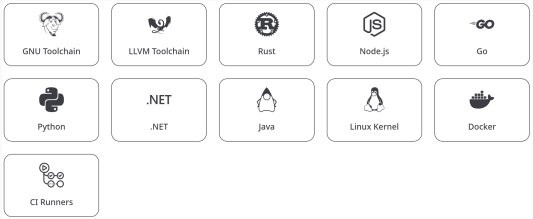


Рисунок 1 – Дистрибутивы, поддерживающие LoongArch

При разработке для не-x86 систем многие могут задаются вопросом:
– Где находится SDK этого устройства?

Благодаря усилиям участников сообщества и инженеров Loongson, начиная с 2021 года большинство наборов инструментов имеют поддержку в исходном коде. Процесс установки различных наборов инструментов программирования теперь аналогичен, если не идентичен, платформе x86 – установка так же проста, как установка из репозиториев дистрибутива и осуществляется с помощью командной строки!



Научное ПО с нативной поддержкой loongarch64

Стек покрывает полный цикл: генерация событий, моделирование, анализ, статистика, визуализация.

Физика высоких энергий

- **CERN ROOT** — анализ больших данных, статистика, визуализация; стандарт на LHC, FAIR, NICA.
- **Geant4** — Монте-Карло симуляция частиц и излучения в веществе; детекторы, радиационная защита, медицина.
- **LHAPDF** — партонные функции распределения для расчётов сечений на адронных коллайдерах.
- **Pythia8** — генератор событий: ливни, адронизация, множественные взаимодействия.

Низкоуровневые библиотеки

- OpenBLAS, BLAS, LAPACK, FFTW, GSL, HDF5, h5py, MPICH, OpenMPI.

Статистика и численные расчёты

- **R** — статистический анализ и графика; стандарт в биоинформатике и анализе данных.
- **GNU Octave** — численные расчёты, совместим с MATLAB; линейная алгебра, ОДУ.

Визуализация

- **Gnuplot** — 2D/3D-графики для научных публикаций.

Экосистема Python

- **NumPy** — многомерные массивы, быстрая математика.
- **SciPy** — оптимизация, интегрирование, интерполяция, обработка сигналов.
- **Matplotlib** — публикационные графики.

Результаты тестирования в сравнении с Эльбрус и Intel Xeon

Характеристики CPU

CPU	Threads	GHz	L1 I/D, KB	L2, KB	L3, MB	RAM, GB
Эльбрус 8СВ (× 4)	8(× 4)	1.5	128/64	512	16	128 (16ch)
Иртыш С616	32	2.2	64/64	256	32	64 (4ch)
Иртыш С632 (× 2)	64 (× 2)	2.1	64/64	256	32	96 (4ch)
Intel Xeon Silver 4314	32	2.4	32/48	1280	24	32 (2ch)
Intel Xeon Silver 4310	24	2.1	32/48	1280	18	128 (2ch)
Intel Xeon E5 2697v2 (× 2)	24 (× 2)	2.7	32/32	256	30	128 (8ch)

Таблица 1 – Основные характеристики CPU.

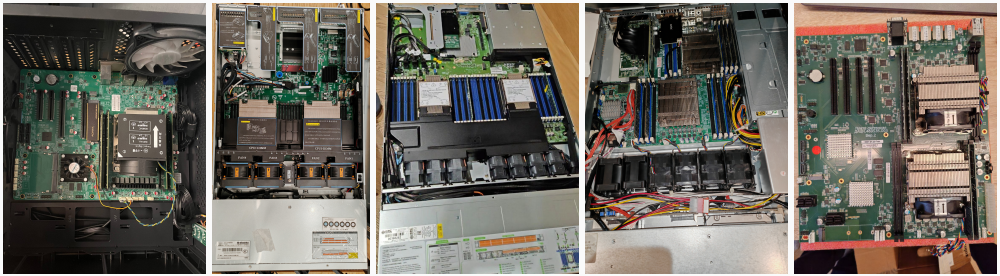


Рисунок 2 – Иртыш С616, С632, Intel Xeon Silver 4310, Intel Xeon E5 2697v2, Эльбрус 8СВ.

Результаты однопоточного тестирования

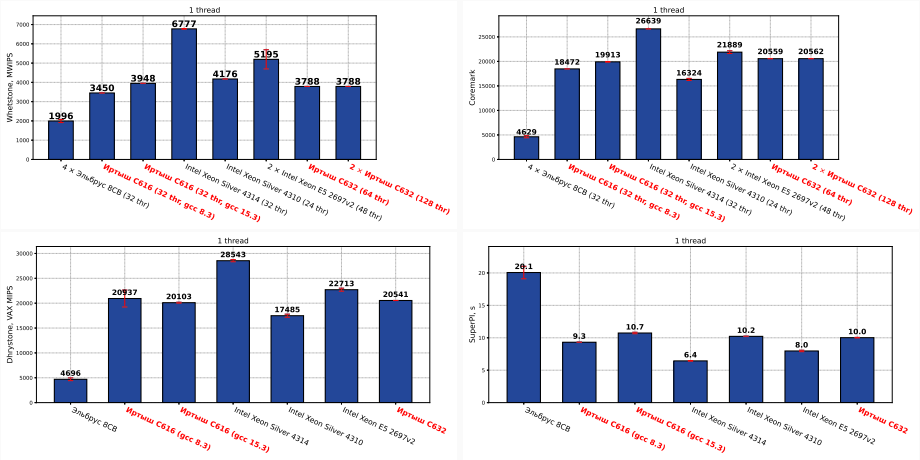


Рисунок 3 – Результат однопоточного тестирования в Whetstone, Dhrystone, Coremark, SuperPi. Большее значение – лучшее (кроме SuperPi).

Результаты многопоточного тестирования

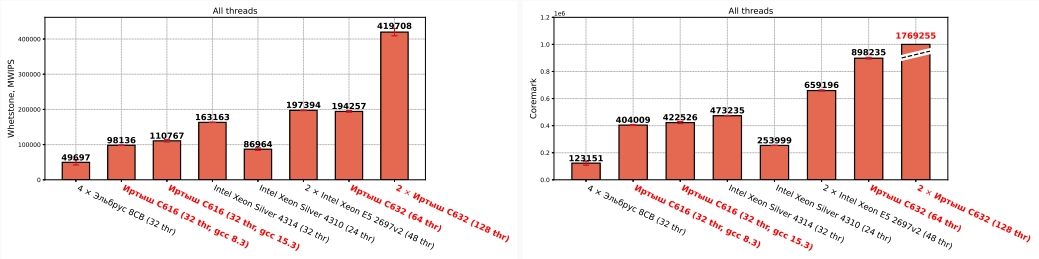


Рисунок 4 – Результат многопоточного тестирования в WhetstoneMP, Coremark. Большее значение – лучшее.

Результаты в приближенных к реальным задачам

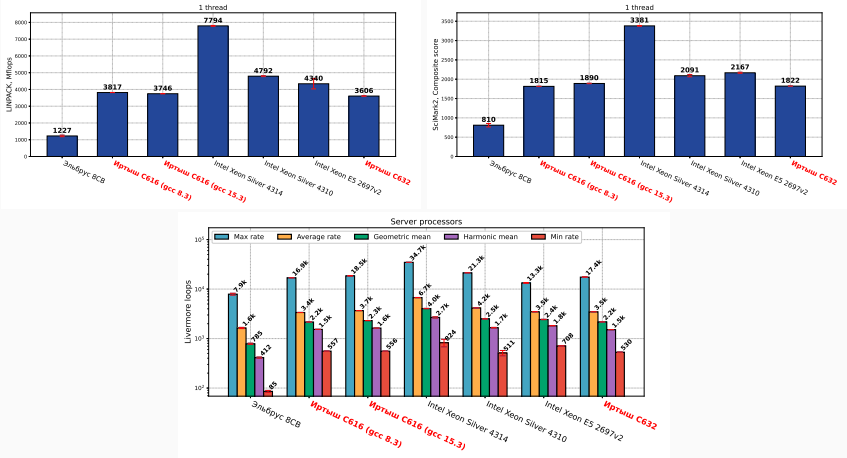


Рисунок 5 – Результат однопоточного тестирования в Linpack, SciMark 2, Livermore loops. Большее значение – лучшее.

Результаты тестирования серверных процессоров Иртыш в SPEC CPU

SPEC CPU® содержит разработанные SPEC набор тестов нового поколения, соответствующие отраслевым стандартам и ориентированные на интенсивную нагрузку на ЦП, предназначенные для измерения и сравнения производительности при выполнении вычислительно-интенсивных задач, при этом подвергая интенсивному нагрузке процессор системы, подсистему памяти и компилятор.

Набор тестов SPEC CPU содержит тесты, сгруппированные в четыре набора:

- Наборы SPECspeed® Integer и SPECspeed® Floating Point используются для сравнения времени, затрачиваемого компьютером на выполнение отдельных задач.
- Наборы SPECrate® Integer и SPECrate® Floating Point измеряют пропускную способность или объем работы в единицу времени.

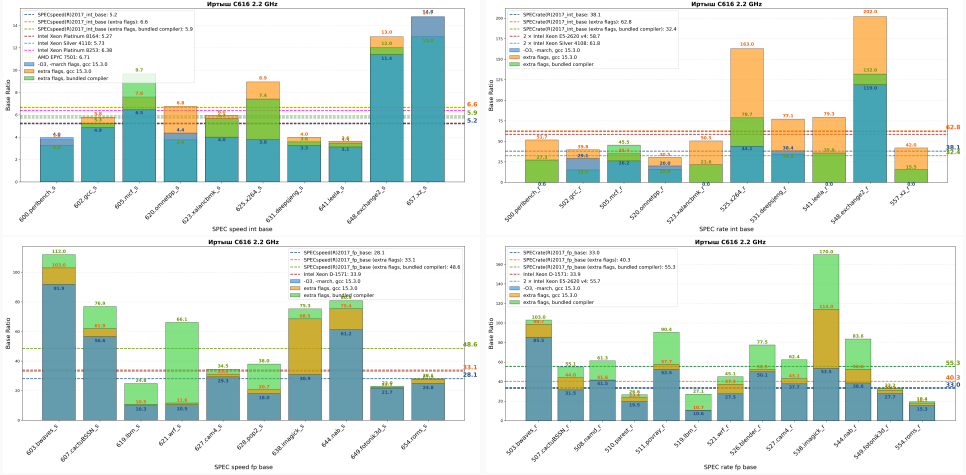


Рисунок 6 – Результаты тестирования C616 в SPEC CPU 2017

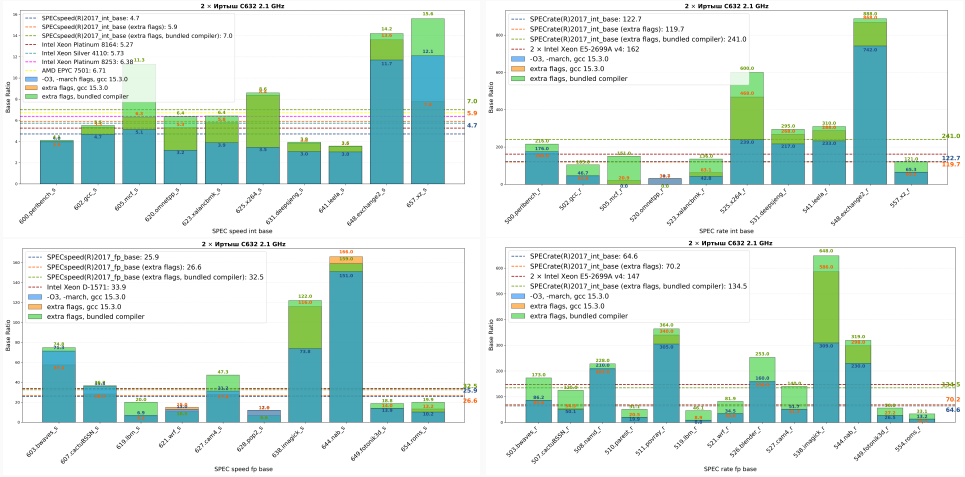


Рисунок 7 – Результаты тестирования 2 × C632 в SPEC CPU 2017

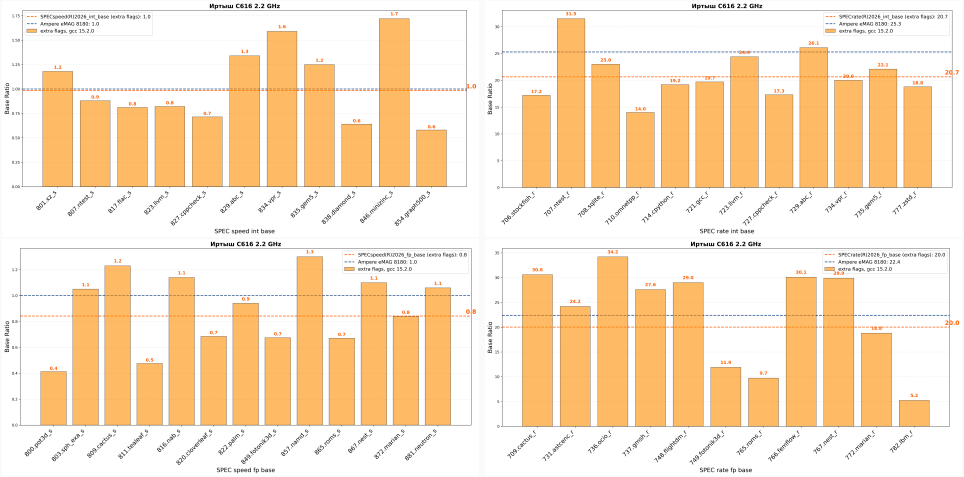


Рисунок 8 – Результаты тестирования C616 в SPEC CPU 2026

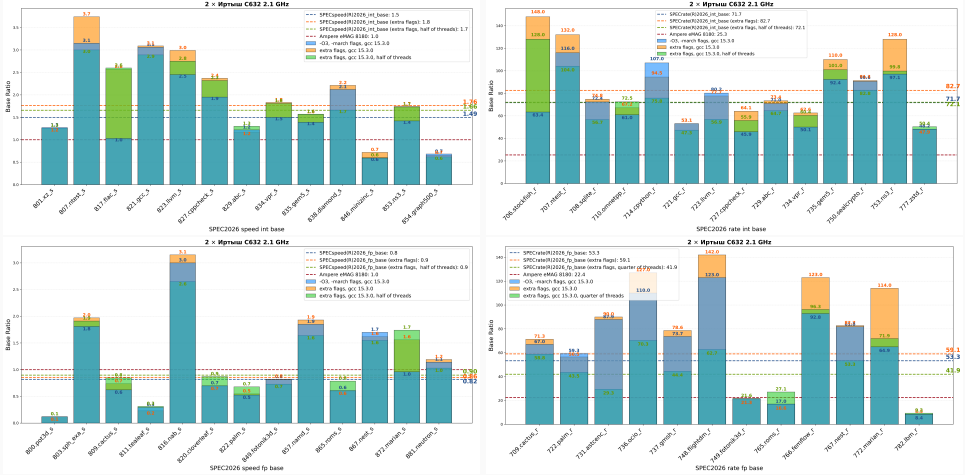
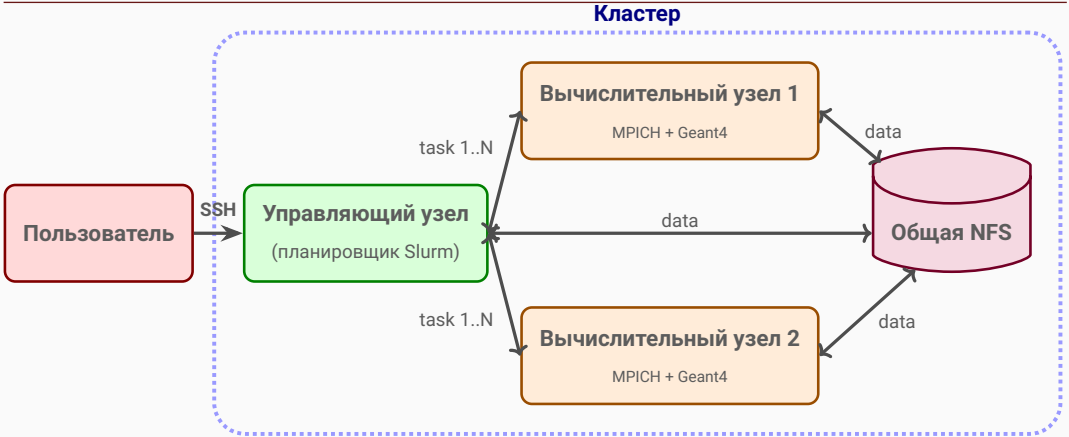


Рисунок 9 – Результаты тестирования 2 × С632 в SPEC CPU 2026

Прототип HPC кластера с планировщиком slurm на базе процессоров Иртыш

Схема вычислительного кластера

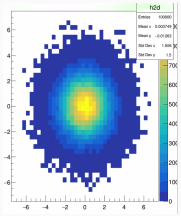


Рабочий процесс: • постановка задач в очередь Slurm, • гибкое управление ресурсами • моделирование в Geant4 через MPICH, • просмотр статуса и истории задач

Демонстрация работы HPC кластера

Запуск рачетов в Geant4, просмотр статуса задач и результат рачета:

```
[clusteradmin@head geant4-examples]$ sbatch -p main -n 16 --wrap "mpiexec ./HGCal_testbeam run.mac"
Submitted batch job 44
[clusteradmin@head geant4-examples]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                44      main     wrap clustera  R       0:07      2 node[1-2]
[clusteradmin@head geant4-examples]$ sbatch -p main -n 16 --wrap "mpiexec ./HGCal_testbeam run.mac"
Submitted batch job 45
[clusteradmin@head geant4-examples]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                45      main     wrap clustera PD       0:00      2 (Resources)
                44      main     wrap clustera  R       0:13      2 node[1-2]
[clusteradmin@head geant4-examples]$
```

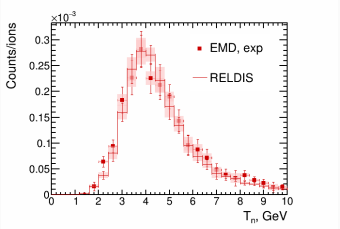


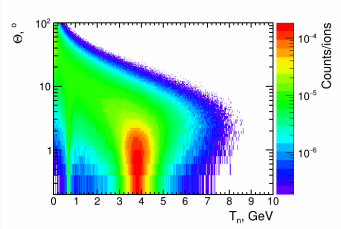
Пример запуска научного ПО в области физики высоких энергий

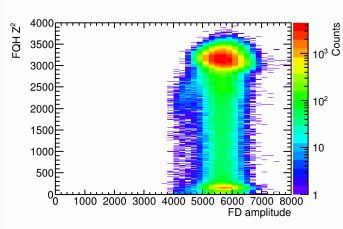
Тестирование ROOT

Построим гистограммы из моей статьи[1] (Q1, IF 3.8). Просмотр статьи доступен по ссылке: https://rdcu.be/eHp1A

Данные получены как из экспериментальных данных, так и из моделирования в Geant4.







[1] Zubankov, A., Afanasiev, S., Golubeva, M. et al. Performance study of the Highly Granular Neutron Detector prototype in the BM@N experiment. *NUCL SCI TECH* **36**, 226 (2025). DOI: 10.1007/s41365-025-01817-x

Портирование BmnRoot для эксперимента BM@N

Подготовлена нативная сборка фреймворка BmnRoot, используемого для набора и анализа измерений на установке BM@N ускорительного комплекса NICA в ОИЯИ, г. Дубна. Поскольку проект создавался под x86-64 архитектуру с активным использованием SSE и AVX инструкций, то для удобства портирование выполнялось с использованием библиотеки SIMDe (SIMD Everywhere).

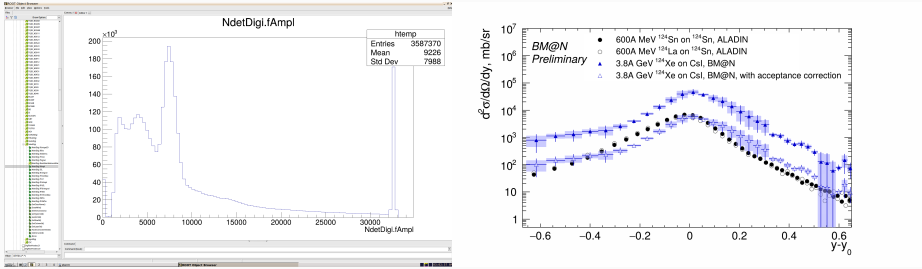


Рисунок 10 – Портирование позволило провести оцифровку из сырых данных, набранных в 2022 году, а также провести обработку и получить сечения нейтронов.

Geant4 – инструмент моделирования прохождения частиц через вещество в ядерной, ускорительной и физике высоких энергий, а также в областях медицины и космонавтики.

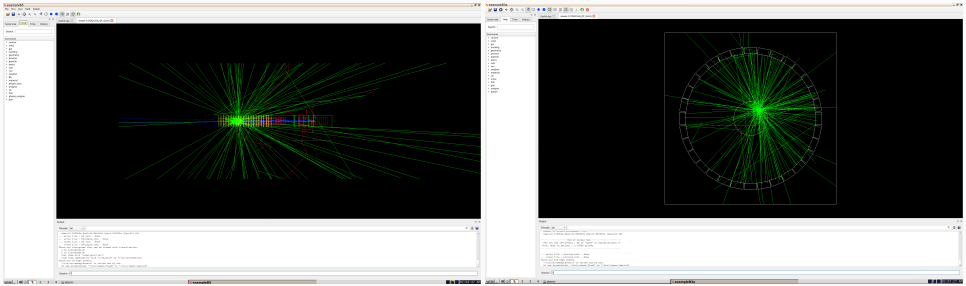


Рисунок 11 – Визуализация базовых примеров моделирования взаимодействия излучения с веществом (калориметр и ПЭТ) в Geant4.

Выводы

Выводы по архитектуре и экосистеме

- Архитектура LoongArch – современная развивающаяся RISC архитектура, цель которой – стать третьей по популярности архитектурой после x86-64 и aarch64.
- Благодаря усилиям сообщества и разработчиков по поддержке архитектуры в исходных кодах, LoongArch очень удобна в переходе с других архитектур и использовании – не требуются SDK и специальные сборки, все работает в любом дистрибутиве.
- Процессоры Иртыш ”из коробки” без дополнительных оптимизаций позволяют достичь достаточного уровня производительности для замещения Intel, а доступность на рынке и большой объем ПО с открытым исходным кодом позволяют обеспечить плавный переход на новую архитектуру.

Выводы

- В однопоточных тестах производительности Иртыш С616 и С632 на уровне Intel Xeon Silver 4310 и Xeon 2697v2, в $\sim$1.5 раза хуже Xeon Silver 4314, но в $\sim$2-3 раза лучше Эльбрус-8СВ.
- В многопоточных тестах производительности Иртыш С616 в $\sim$1.5 раза опережает Intel Xeon Silver 4310, приближаясь к Xeon Silver 4314, а Иртыш С632 опережает 2$\times$Xeon 2697v2 и Xeon Silver 4314 благодаря большему числу потоков.
- В SPECrate и SPECspeed CPU 2026 ближайшим аналогом для С616 ялвляется Ampere eMAG 8180. Для 2 $\times$ С632 аналогичными в SPECrate можно считать Intel Xeon 6503P-B и NVIDIA GB10.
- В SPECrate CPU 2017 ближайшими аналогами для С616 являются 2 $\times$ Intel Xeon E5-2620 v4 и Intel Xeon D-1571. Для 2 $\times$ С632 аналогичным в SPECrate можно считать 2 $\times$ Intel Xeon E5-2699A v4.

Спасибо за внимание!

a.zubankov@tramplin.tech

Презентация полностью выплнена на процессоре Иртыш С616

Backup

Подход к сравнению производительности

Производительность CPU оценивалась набором бенчмарков **anybench** от EntityFX.

Для каждого CPU использовался компилятор gcc с оптимизацией -O3 и -march=native:

- *Иртыш С616* - gcc 8.3.0 & gcc 15.3.0
- *Иртыш С632* - gcc 15.2.0
- *Эльбрус 8СВ* - lcc 1.29.12 (gcc 11.3.0 compatible)
- *Intel Xeon Silver* - gcc 14.3.1
- *Intel Xeon E5* - gcc 16.1.1

Большинство бенчмарков предназначены для однопоточных вычислений. Многопоточные бенчмарки представлены для максимального числа потоков. Бенчмарки запускались по несколько раз для оценки среднего и СКО.

Стабильность результата от компилятора (gcc/clang)

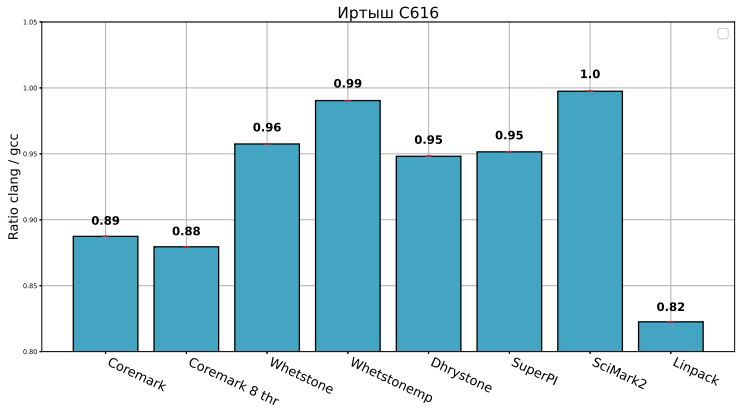


Рисунок 12 – Отношение результатов бенчмарков с компилятором clang 20.1.6 к gcc 14.2.0. Результаты в среднем хуже на 10%.

Стабильность от ОС

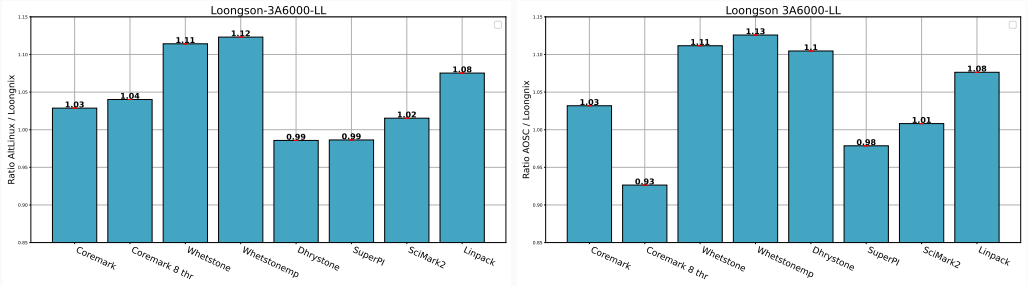


Рисунок 13 – Отношение результатов бенчмарков в AltLinux (gcc 14.2.1) (слева) и AOSC (gcc 14.2.0) (справа) к Loongnix (gcc 8.3.0). Результаты в пределах 15%, что говорит о правильности набора тестов для оценки производительности ЦПУ.

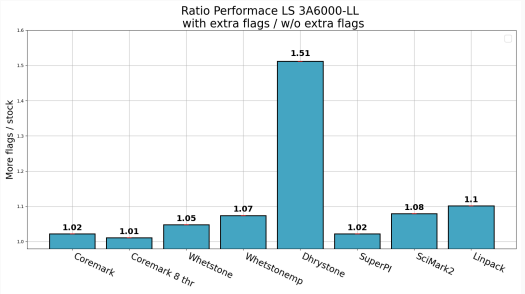
Additional flags / Дополнительные флаги

In addition to the flags -march=native -mtune=native, the following were used in this testing:
Помимо флагов -march=native -mtune=native в данном тестировании использовались следующие:

loongarch64: -mabi=lp64d -mlasx

loongarch64 & x86_64: -ljemalloc -lpthread -flto=auto -fno-prefetch-loop-arrays -funroll-loops -lstdc++ -mrecip -mfrecipe -flto-partition=one

Additional flags / Дополнительные флаги



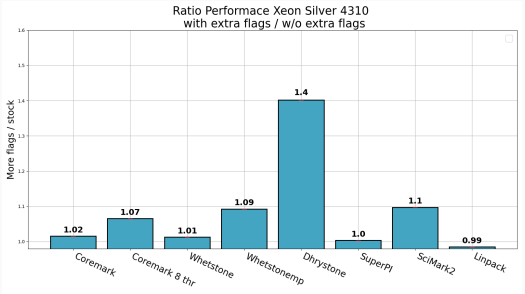


Рисунок 14 – Ratio of benchmark results with additional flags to the original.
Отношение результатов бенчмарков с дополнительными флагами к исходным.

Валидация результатов

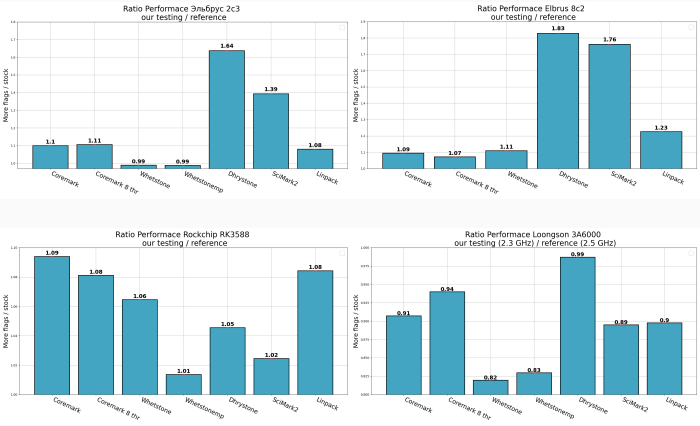
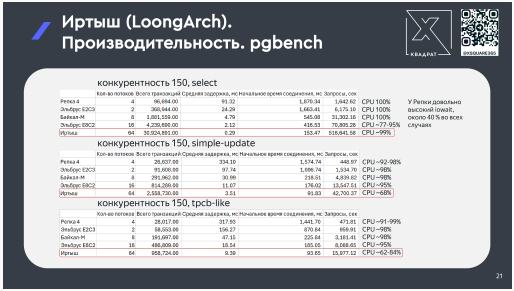


Рисунок 15 – В сравнении с оригинальными результатами Anybench.

Тестирование ЦПУ Иртыш от XSQUARE



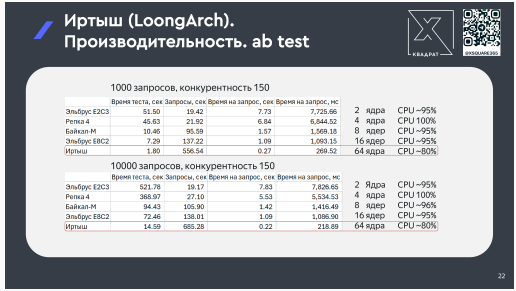


Рисунок 16 – https://pgbootcamp.ru/talks/7oaey4csp1-opit-ekspluatatsii-problemi-i-proizvodit

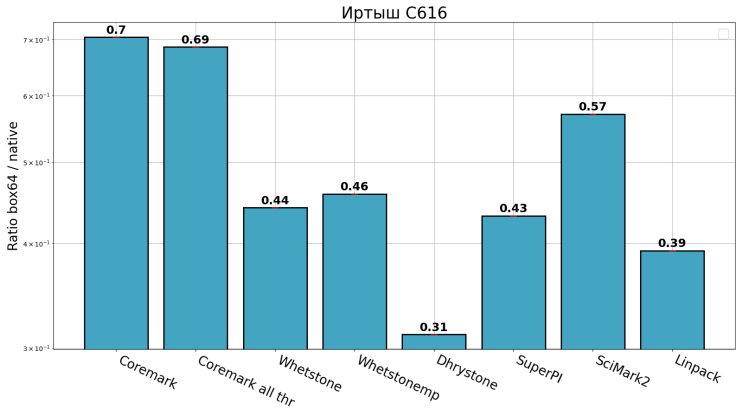


Рисунок 17 – Отношение результатов бенчмарков скомпилированных на x86-64 и запущенных через box64 к скомпилированным нативно.

SPECrate Integer	SPECspeed Integer	Language	Application Area
500.perlbench_r	600.perlbench_s	C	Perl interpreter
502.gcc_r	602.gcc_s	C	GNU C compiler
505.mcf_r	605.mcf_s	C	Route planning
520.omnetpp_r	620.omnetpp_s	C++	Discrete Event simulation - computer network
523.xalancbmk_r	623.xalancbmk_s	C++	XML to HTML conversion via XSLT
525.x264_r	625.x264_s	C	Video compression
531.deepsjeng_r	631.deepsjeng_s	C++	AI: alpha-beta tree search (Chess)
541.leela_r	641.leela_s	C++	AI: Monte Carlo tree search (Go)
548.exchange2_r	648.exchange2_s	Fortran	AI: recursive solution generator (Sudoku)
557.xz_r	657.xz_s	C	General data compression

SPEC CPU 2017 fp

SPECrate FP	SPECspeed FP	Language	Application Area
503.bwaves_r	603.bwaves_s	Fortran	Explosion modeling
507.cactuBSSN_r	607.cactuBSSN_s	C++, C, Fortran	Physics: relativity
508.namd_r		C++	Molecular dynamics
510.parest_r		C++	Biomedical imaging: optical tomography with finite elements
511.povray_r		C++, C	Ray tracing
519.lbm_r	619.lbm_s	C	Fluid dynamics
521.wrf_r	621.wrf_s	Fortran, C	Weather forecasting
526.blender_r		C++, C	3D rendering and animation
527.cam4_r	627.cam4_s	Fortran, C	Atmosphere modeling
	628.pop2_s	Fortran, C	Wide-scale ocean modeling (climate level)
538.imagick_r	638.imagick_s	C	Image manipulation
544.nab_r	644.nab_s	C	Molecular dynamics
549.fotonik3d_r	649.fotonik3d_s	Fortran	Computational Electromagnetics
554.roms_r	654.roms_s	Fortran	Regional ocean modeling

SPEC CPU 2026 int

SPECrate Integer	SPECspeed Integer	Language	Application Area
	801.xz_s	CXX, C	Data compression
706.stockfish_r		CXX	Game / AI (chess) - alpha-beta tree search, neural network
707.ntest_r	807.ntest_s	CXX	Game / AI (othello)
708.sqlite_r		C	SQL compiler/interpreter and database
710.omnetpp_r		CXX, C	Discrete event modeling - network and queuing simulations
714.cpython_r		C	Python interpreter
	817.flac_s	CXX, C	Lossless audio compression
721.gcc_r	821.gcc_s	CXX, C	C language optimizing compiler
723.llvm_r	823.llvm_s	CXX, C	C/C++ language optimizing compiler
727.cppcheck_r	827.cppcheck_s	CXX	Static analysis of C/C++ code
729.abc_r	829.abc_s	CXX, C	Sequential logic synthesis and formal verification
734.vpr_r	834.vpr_s	CXX, C	FPGA place and route
735.gem5_r	835.gem5_s	CXX, C	Computer architecture simulation
	838.diamond_s	CXX, C	Bioinformatics - metagenomics and protein sequencing
	846.minizinc_s	CXX, C	Constraint programming
750.sealcrypto_r		CXX, C	Security and privacy - Homomorphically Encrypted (HE) query
753.ns3_r	853.ns3_s	CXX	Discrete event network simulator for internet systems
	854.graph500_s	C	Graph analytics
777.zstd_r		C	Data compression/decompression

SPEC CPU 2026 fp

SPECrate FP	SPECspeed FP	Language	Application Area
	800.pot3d_s	F	Solar physics: finite difference method, conjugate gradient solver
	803.sph_exa_s	CXX	Astrophysics - Smoothed Particle Hydrodynamics (SPH)
709.cactus_r	809.cactus_s	CXX, C	Astrophysics - relativity, finite difference method, time integration
	811.tealeaf_s	C	High energy physics
	816.nab_s	C	Molecular modeling
	820.cloverleaf_s	F	Explicit hydrodynamics
722.palm_r	822.palm_s	F	Atmospheric science
731.astcenc_r		CXX	Image compression - Adaptive Scalable Texture Compression (ASTC)
736.ocio_r		CXX	Color management for visual effects and animation
737.gmsh_r		CXX, C	Finite element mesh generation
748.flightdm_r		CXX	Flight dynamics models for aeronautics
749.fotonik3d_r	849.fotonik3d_s	F	Computational Electromagnetics (CEM)
	857.namd_s	CXX	Classical molecular dynamics simulation
765.roms_r	865.roms_s	F	Regional ocean modeling
766.femflow_r		CXX	Fluid dynamics: high-order finite element method
767.nest_r	867.nest_s	CXX	Neuroscience simulator for spiking neural network models
772.marian_r	872.marian_s	CXX	Neural machine translation for written language
782.lbm_r		C	Computational fluid dynamics, Lattice Boltzmann Method
	881.neutron_s	C	Physics simulation of neutron transport in nuclear reactors