Исследование производительности, энергоэффективности и оптимизация кодов для решения задач физики плазмы

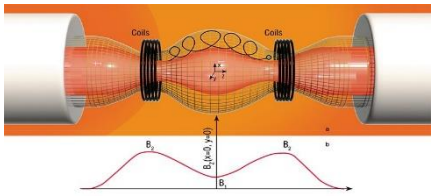
к.ф.-м.н., рук ЦКП ССКЦ СО РАН, зав лаб суперкомпьютерного моделирования ИВМиМГ СО РАН

Черных Игорь Геннадьевич

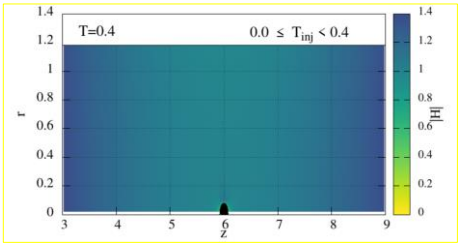
Введение

К задачам физики плазмы относится исследование процессов, определяющих поведение и свойства ионизованной среды, включая коллективное взаимодействие заряженных частиц, электромагнитные поля, перенос энергии и вещества, а также развитие неустойчивостей.

Физика плазмы охватывает как фундаментальные исследования, так и прикладные задачи, связанные с термоядерным синтезом, космической плазмой, плазменными технологиями и газовыми разрядами.



A. A. Shoshin, A. V. Anikeev, 23 Apr 2018 , Budker's Universe , volume Special Issue, N1





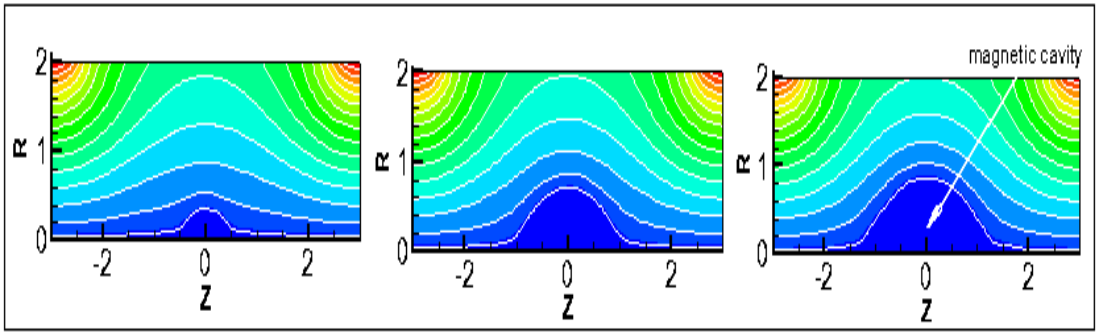
Введение

В начальный момент магнитное поле в камере ловушки $B=(0,0,B_0)$. Камера заполнена водородной фоновой плазмой с плотностью n_0. Источник нейтральных частиц расположен внутри области. Быстрая ионизация обеспечивает инжекцию ионов и электронов, которые взаимодействуют с фоновой плазмой.

Теоретически, эволюция системы может привести к образованию насыщенного частицами стационарного диамагнитного «пузыря» и соответствующего удерживающего плазму магнитного поля [А. Беклемишев, 2016]. Цель – исследование параметров образования диамагнитного пузыря.

На рисунке представлены расчеты по двумерной модели с аксиальной симметрией в цилиндре.

Введение



Показано, что самосогласованное взаимодействие пучка с плазмой приводит к формированию магнитной каверны, размеры которой зависят от характеристик ионного пучка и определяют объём и плотность накопленной плазмы..

G.I. Dudnikova, M.A. Boronina, E.A. Genrikh, V.A. Vshivkov. Computer simulation of ion beam-plasma interaction // Journal of Physics: Conference Series -2019.-V. 1393. – 012041.

Уравнения

Это одна из наиболее полных классических моделей бесстолкновительной плазмы.

Кинетическое уравнение Власова

$$\frac{\partial f_i}{\partial t}+\vec{v}\frac{\partial f_i}{\partial \vec{r}}+\frac{\vec{F}_i}{m_i}\frac{\partial f_i}{\partial \vec{v}}=0$$

$$\vec{F}_i=e\left(\vec{E}+\frac{1}{c}\left[\vec{v},\vec{B}\right]\right)+R_i$$

$$n_i\left(\vec{r}\right)=\int f_i\left(t,\vec{r},\vec{v}\right)d\vec{v}$$

$$\vec{V}_i\left(\vec{r}\right)=\frac{1}{n_i\left(\vec{r}\right)}\int \vec{v} f_i\left(t,\vec{r},\vec{v}\right)d\vec{v}$$

МГД уравнения

$$-e\vec{E}-\frac{e}{c}\left[\vec{V}_e,\vec{B}\right]-\frac{\nabla p_e}{n_e}+\vec{R}_e=m_e\frac{d\vec{V}}{dt}$$

$$\vec{R}_e=-m_e\frac{\vec{V}_e-\vec{V}_i}{\tau_{ei}}$$

Уравнения Максвелла

$$\frac{1}{c}\frac{\partial \vec{E}}{\partial t}=rot\vec{B}-\frac{4\pi}{c}\vec{j}$$

$$\frac{1}{c}\frac{\partial \vec{B}}{\partial t}=-rot\vec{E}$$

$$div\vec{E}=4\pi\rho$$

$$div\vec{B}=0$$

$$n_i=n_e$$

$$R_i=-R_e$$

$$\vec{j}=e\left(n_i\vec{V}_i-n_e\vec{V}_e\right)$$

$$\rho=e\left(n_i-n_e\right)$$

Уравнения переноса тепла

$$n_e\left(\frac{\partial T_e}{\partial t}+\left(\vec{V}_e\nabla\right)T_e\right)=\left(\gamma-1\right)\left(Q_e-div\vec{q}_e-p_e div\vec{V}_e\right)$$

$$Q_e=\frac{j^2}{\sigma}$$

$$\vec{q}_e=-k\nabla Te$$

Уравнение состояния

$$p_e=n_eT_e$$

Уравнения

Для каждого сорта частиц s решается кинетическое уравнение Власова:

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_s + \frac{q_s}{m_s}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f_s = 0.$$

Здесь $f_s(\mathbf{x}, \mathbf{v}, t)$ — функция распределения частиц по координатам и скоростям.

Самосогласованные электромагнитные поля определяются уравнениями Максвелла:

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E},$$

$$\frac{\partial \mathbf{E}}{\partial t} = c^2 \nabla \times \mathbf{B} - \frac{\mathbf{J}}{\varepsilon_0}.$$

Плотность заряда и ток вычисляются из функций распределения:

$$\rho = \sum_s q_s \int f_s \, d^3v, \qquad \mathbf{J} = \sum_s q_s \int \mathbf{v} f_s \, d^3v.$$

Дополнительно должны выполняться ограничения:

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}, \qquad \nabla \cdot \mathbf{B} = 0.$$

Какие процессы описывает:

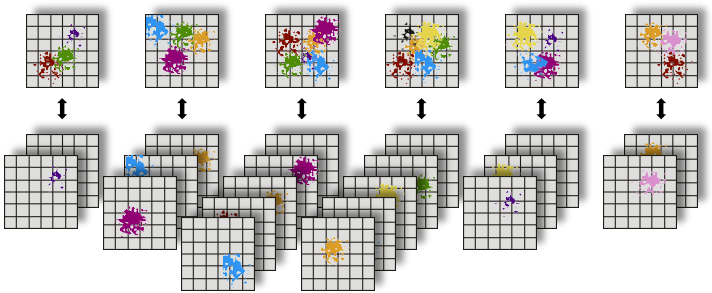
- Двухпотоковую, пучковую и циклотронную неустойчивости.
- Бесстолкновительное затухание Ландау.
- Электромагнитные и электростатические колебания.
- Нелинейное взаимодействие волн и частиц.
- Формирование токовых слоёв и кинетическое магнитное пересоединение.
- Аномальный перенос, вызванный кинетическими неустойчивостями.

Уравнения

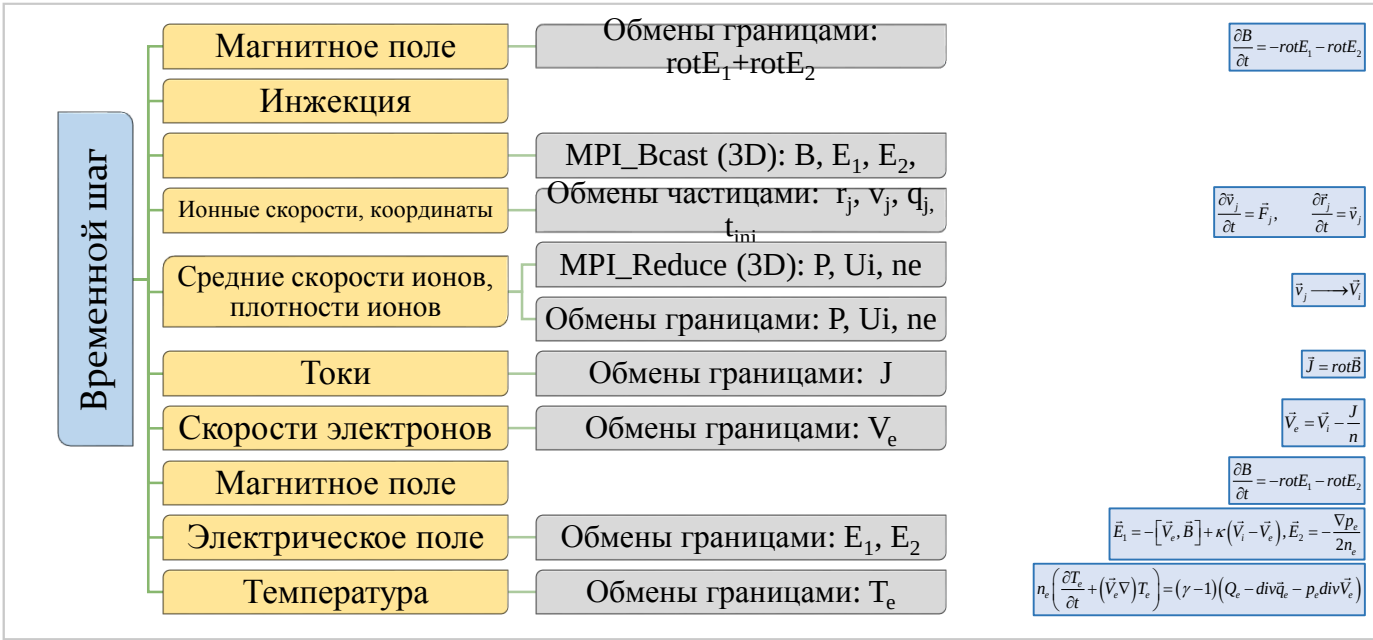
Физический процесс	Основные уравнения	Подход
Движение электронов и ионов в электромагнитных полях	Уравнения Лоренца	Частицы, PIC
Коллективные бесстолкновительные процессы	Власова–Максвелла	Кинетический
Столкновения, релаксация, термализация	Больцмана, Фоккера–Планка, Ландау	Столкновительно-кинетический
Ленгмюровские и ионно-звуковые волны	Власова–Пуассона или Власова–Максвелла	Кинетический
Двухпотоковая и пучковая неустойчивости	Власова–Пуассона, Власова–Максвелла	Кинетический
Альфвеновские и магнитозвуковые волны	Уравнения МГД	Жидкостный
Магнитное пересоединение	МГД, Холл-МГД, двухжидкостные, Власова–Максвелла	Многомасштабный
Диффузия, теплопроводность, вязкость	МГД с транспортными членами, кинетические уравнения	Жидкостный / кинетический
Аномальный перенос и турбулентность	Дрейфово-кинетические, гирокинетические, Власова–Максвелла	Кинетический
Магнитное удержание и потери частиц	Уравнения движения, дрейфово-кинетические, Власова–Максвелла	Кинетический
Газовый разряд, ионизация, рекомбинация	Больцмана, уравнения переноса, Пуассона, кинетические уравнения реакций	Столкновительный
Ионосферные и магнитосферные возмущения	МГД, гибридные, Власова–Максвелла	Многомасштабный

Подходы

- ✓ Декартовы координаты
- ✓ Гибридная модель: ионы описываются кинетически, электроны – МГД
- ✓ Метод частиц-в-ячейках
- ✓ Смешанная декомпозиция (область + частицы)
- ✓ Векторизуемые циклы для движения частиц и расчета плотностей частиц



Один временной шаг



Исследование свойств программы. Компиляция

module load oneapi/2022.3.1/ALL

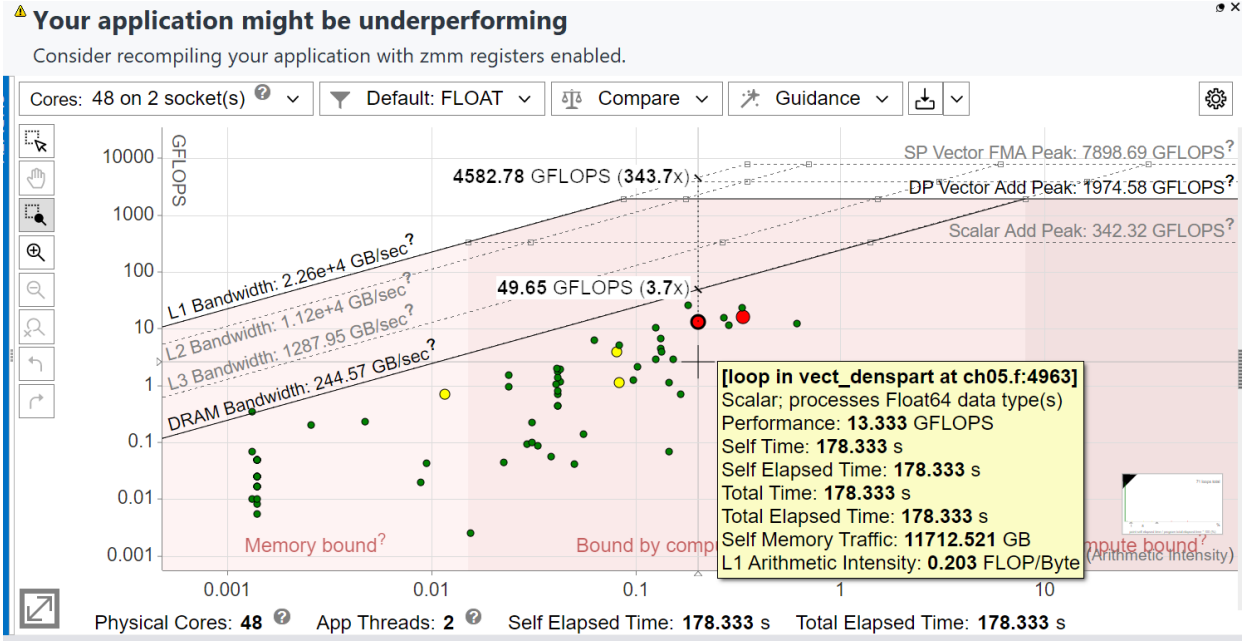
ifort -xCASCADELAKE -g -debug inline-debug-info -O3 -align array64byte -qopt-report=5 -qopt-report-phase=vec -qopenmp-simd -qopt-zmm-usage=high –o <имя файла>

Исследование свойств программы. Intel Advisor

```
module load  oneapi/2022.3.1/ALL

mpirun -n 96 advisor --collect=survey --project-dir=./advi_results
./a.out > run_survey.out
mpirun -n 96 advisor --collect=tripcounts --flop --project-
dir=./advi_results ./a.out > run_flop.out
```

Исследование свойств программы. Intel Advisor



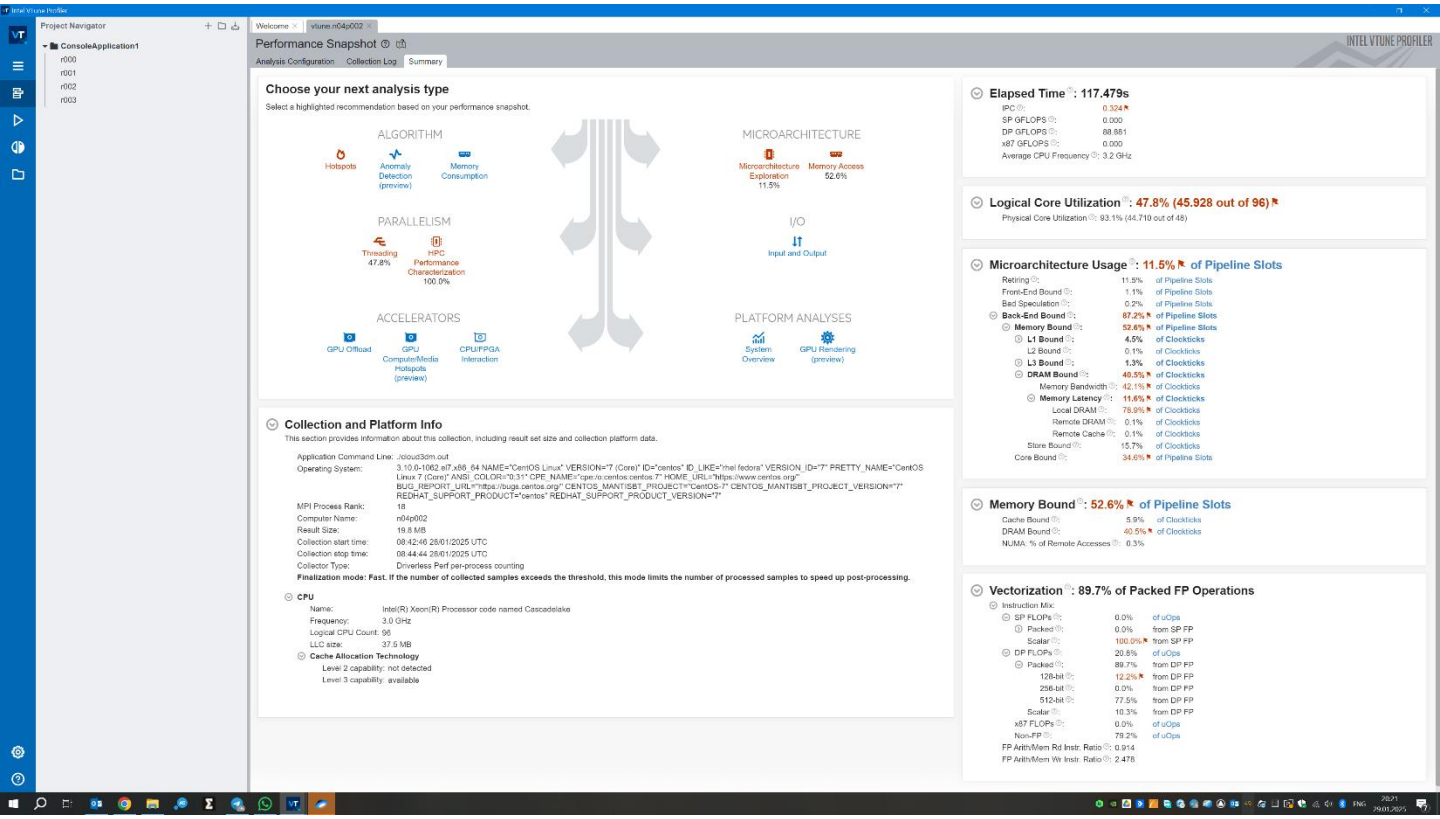
Исследование свойств программы. Intel VTune

```
module load  oneapi/2023.1/ALL_waste
export OMP_NUM_THREADS=32
export OMP_PROC_BIND=close
Export OMP_PLACES=cores
vtune -collect threading -duration=600 -result-
dir=vtune_threading ./dbs2d_tests > run_threading.out
vtune -collect hpc-performance -duration=600 -knob collect-
memory-bandwidth=true -result-dir=vtune_hpc ./dbs2d_tests >
run_hpc.out
vtune -collect uarch-exploration -duration=600 -knob collect-
bad-speculation=false -knob collect-core-bound=false -result-
dir=vtune_uarch  ./dbs2d_tests > run_uarch.out
```

Исследование свойств программы. Intel VTune

vtune -collect performance-snapshot -duration=600 -result-dir=vtune_performance ./dbs2d_tests > run_performance.out
vtune -collect memory-access -duration=600 -result-dir=vtune_memoryacc ./dbs2d_tests > run_memory.out

Производительность



Производительность

Instruction Mix:		
SP FLOPs:	0.0%	of uOps
DP FLOPs:	0.1%	of uOps
Packed:	100.0%	from DP FP
128-bit:	16.1%	from DP FP
256-bit:	0.0%	from DP FP
512-bit:	83.9%	from DP FP
Scalar:	0.0%	from DP FP

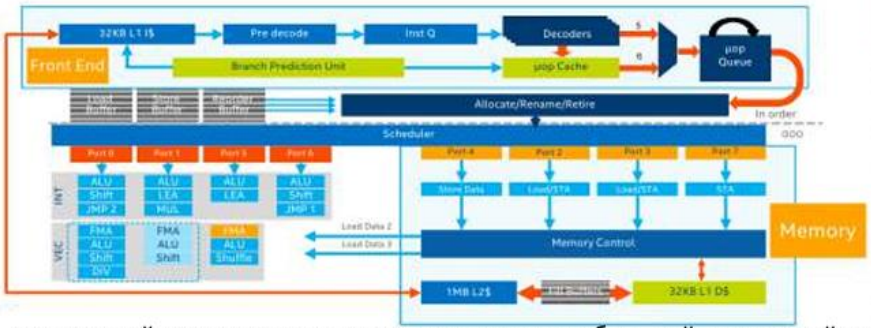
Производительность

Function	CPU Time	% of FP Ops	FP Ops: Packed	FP Ops: Scalar	Vector Instruction Set	Loop Type
[Loop at line 28 in MPIR_SUM]	78.464s	18.8%	100.0%	0.0%	SSE(128); SSE2(128)	Body
system_call	16.158s	0.2%	100.0%	0.0%		
[Loop at line 2300 in temperature]	6.656s	75.4%	100.0%	0.0%	AVX512DQ_512(512); AVX512F_256(256); AVX512F_512(512)	Body
[Loop at line 356 in parmove]	4.049s	4.8%	100.0%	0.0%	AVX512F_512(512)	Body
[Loop at line 617 in parmove]	3.849s	16.3%	100.0%	0.0%	AVX(256); AVX512DQ_512(512); AVX512F_256(256); AVX512F_512(512)	Body
[Others]	23.575s	30.0%	100.0%	0.0%		

*N/A is applied to non-summable metrics.

Векторизация

Улучшения микроархитектуры



	Broadwell uArch	Skylake uArch
Out-of-order Window	192	**224**
In-flight Loads + Stores	72 + 42	72 + **56**
Scheduler Entries	60	**97**
Registers – Integer + FP	168 + 168	**180** + 168
Allocation Queue	56	**64/thread**
L1D BW (B/Cyc) – Load + Store	64 + 32	**128 + 64**
L2 Unified TLB	4K+2M: 1024	4K+2M: **1536** 1G: **16**

- улучшенный предсказатель переходов, декодер с большей пропускной способностью, больше окно извлечения команд (ILP);
- улучшенный планировщик и исполнительные устройства, улучшенная пропускная способность и задержки для операций divide/sqrt;
- увеличена пропускная способность load/store, увеличена глубина load/store Буферов, улучшен механизм предварительной подкачки;
- наличие двух исполнительных устройств на ядро, поддерживающих AVX-512 FMA

Около 10% увеличения производительности на целочисленных приложениях при той же частоте

Векторизация

Эйлеров этап векторизуется легко, но не трудоемок:

$$\frac{B^m_{x\,i-\frac{1}{2},l,k} - B^{m-\frac{1}{2}}_{x\,i-\frac{1}{2},l,k}}{\tau/2} = -(curl\vec{E}^m_1)_x - (curl\vec{E}^m_2)_x$$

$$(curl\vec{E}^m_2)_x = -\frac{1}{h_y h_z \overline{n_i}_{lk}} G^m_{x\,ilk}$$

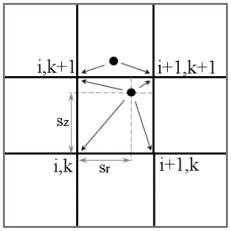
$$G_{x\,ilk} = \Big[\Big(T_{i-\frac{1}{2},l+\frac{1}{2},k+\frac{1}{2}} - T_{i-\frac{1}{2},l-\frac{1}{2},k-\frac{1}{2}}\Big)\Big(n_{i-\frac{1}{2},l-\frac{1}{2},k+\frac{1}{2}} - n_{i-\frac{1}{2},l+\frac{1}{2},k-\frac{1}{2}}\Big) - \Big(n_{i-\frac{1}{2},l+\frac{1}{2},k+\frac{1}{2}} - n_{i-\frac{1}{2},l-\frac{1}{2},k-\frac{1}{2}}\Big)\Big(T_{i-\frac{1}{2},l-\frac{1}{2},k+\frac{1}{2}} - T_{i-\frac{1}{2},l+\frac{1}{2},k-\frac{1}{2}}\Big)\Big]$$

```
do i=1, Imax
do l=1, Lmax-1
do k=1, Kmax-1
  Gn% x(i,l,k) = ((T(i,l+1,k+1) - T(i,l,k)) * ( n(i,l,k+1) - n(i,l+1,k))-
                 ( n(i,l+1,k+1) - n(i,l,k)) * ((T(i,l,k+1) - T(i,l+1,k))))/
                 ( n(i,l,k) + n(i,l+1,k) + n(i,l,k+1) + n(i,l+1,k+1) )
```

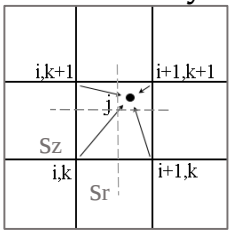
Время счета, сек

Динамика частиц	215	46%
Плотности	237	51%
Полное время	463	

Плотность



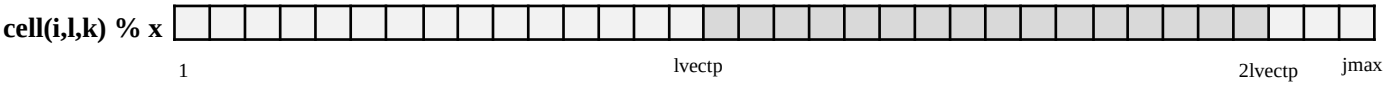
Сила, действующая на частицу



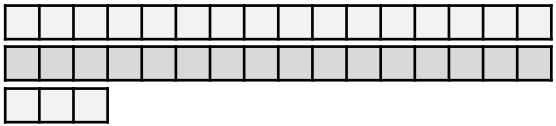
Сетка - частицы

теперь частицы привязаны к ячейке cell(i,l,k)
cell(i,l,k) % jmax – сколько в ячейке частиц
hole $_{i,l,k}$ % j – номера вылетевших из ячейки
in $_{i,l,k}$ % x(j), y(j)… – их данные

```
do j=1,  jmax
   dx(j)=…
   dy(j)=…
   dz(j)=…
```



НО: Cache L1
Cascade Lake
data 32kb
Поэтому:



do k=1, Kmax
do l=1, Lmax
do i=1, Imax
do j=1, cell(i,l,k)%jmax

- вычисление пачками скоростей и координат
- граничные условия
- влетевшие / вылетевшие: Nin, Nout, hole, in
- обмены частицами
- сортировка: удаление вылетевших и вставка влетевших частиц с учетом инжекции на следующем шаге.

```
DO jpack=1, jmax, lvectp
 DO jloc=1, MIN(lvectp, jmax - jpack +
1)
     j = jpack+jloc-1
        dx(jloc)=…
        dy(jloc)=…
        dz(jloc)=…
```

Объявление переменных

Structure of arrays and data alignment:

```
type coo
real*8, allocatable :: r(:)
real*8, allocatable :: z(:)
real*8, allocatable :: u(:)
real*8, allocatable :: v(:)
real*8, allocatable :: w(:)
real*8, allocatable :: a(:)
real*8, allocatable :: it(:)
dir$ attributes align:64 :: r
dir$ attributes align:64 :: z
dir$ attributes align:64 :: u
dir$ attributes align:64 :: v
dir$ attributes align:64 :: w
dir$ attributes align:64 :: a
end type coo
```

Векторизация AVX512

```
257           jEx(jloc) = sx(jloc)*( sy(jloc)*( sz(jloc)*cEx(1,1,1) + dz(jloc)*cEx(1,1,2) )
258     *               +            dy(jloc)*( sz(jloc)*cEx(1,2,1) + dz(jloc)*cEx(1,2,2) ))
259     *               + dx(jloc)*( sy(jloc)*( sz(jloc)*cEx(2,1,1) + dz(jloc)*cEx(2,1,2) )
260     *               +            dy(jloc)*( sz(jloc)*cEx(2,2,1) + dz(jloc)*cEx(2,2,2) ))
```

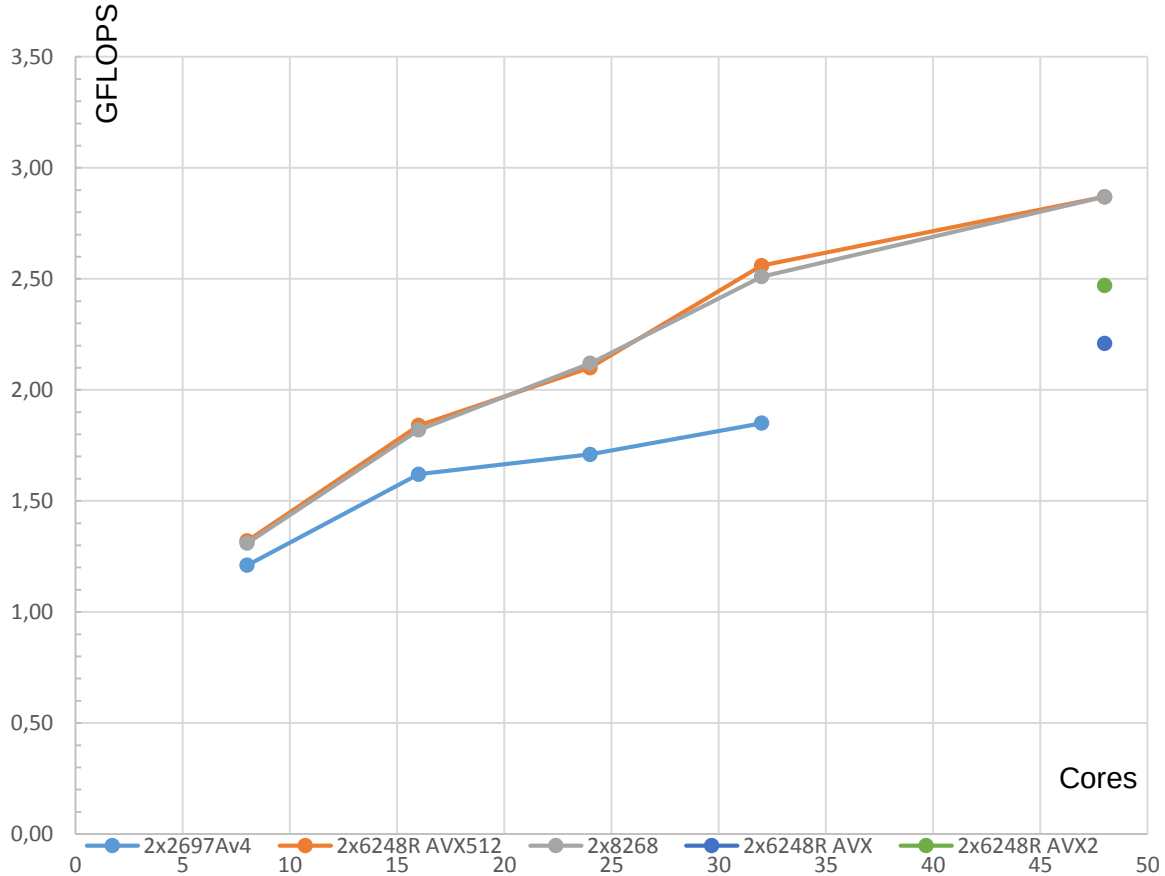
Module: cloud3dm.out!0x462598

Address	Lin.	
0x462755	254	vfmadd213pd %zmm7, %zmm8, %k0, %zmm31
0x46275b	260	vmovupsz -0x7f0(%rbp), %k0, %zmm7
0x462765	257	vfmadd231pd %zmm10, %zmm19, %k0, %zmm9
0x46276b	252	vfmadd213pd %zmm3, %zmm8, %k0, %zmm30

[loop in move_packj at push.f:221]
Vectorized (Body) AVX512; processes Float32; Float64 data type(s)
Performance: **10.951** GFLOPS
Self Time: **12.296** s
Self Elapsed Time: **12.296** s
Total Time: **12.296** s
Total Elapsed Time: **12.296** s
Self Memory Traffic: **396.151** GB
L1 Arithmetic Intensity: **0.34** FLOP/Byte

[loop in move_packj at push.f:221]
Vectorized (Body) AVX512; processes Float32; Float64 data type(s)
Performance: **12.19** GFLOPS
Self Time: **11.046** s
Self Elapsed Time: **11.046** s
Total Time: **11.046** s
Total Elapsed Time: **11.046** s
Self Memory Traffic: **396.151** GB
L1 Arithmetic Intensity: **0.34** FLOP/Byte

Производительность



Энергоэффективность

nature astronomy

Comment | Published: 10 September 2020

The ecological impact of high-performance computing in astrophysics

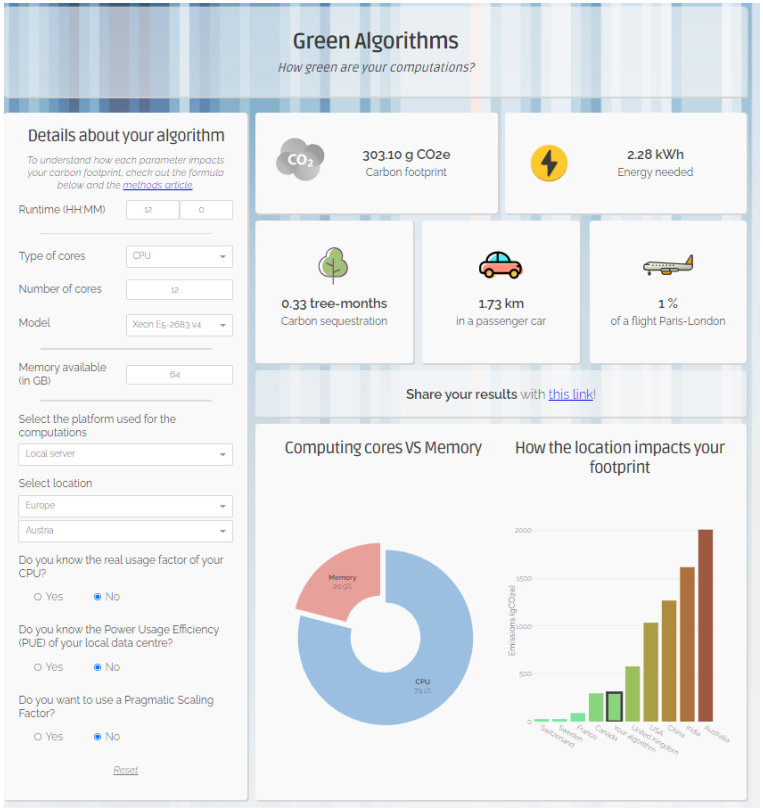
Simon Portegies Zwart

Nature Astronomy **4**, 819–822 (2020) | Cite this article

1908 Accesses | **10** Citations | **552** Altmetric | Metrics

Computer use in astronomy continues to increase, and so also its impact on the environment. To minimize the effects, astronomers should avoid interpreted scripting languages such as Python, and favour the optimal use of energy-efficient workstations.

Энергоэффективность



https://www.advancedsciencenews.com/measuring-computers-carbon-footprint-with-green-algorithms/

The formula

The carbon footprint is calculated by estimating the energy draw of the algorithm and the carbon intensity of producing this energy at a given location:

```
carbon footprint = energy needed * carbon intensity
```

Where the energy needed is:

```
runtime * (power draw for cores * usage + power draw for memory) * PUE * PSF
```

The power draw for the computing cores depends on the model and number of cores, while the memory power draw only depends on the size of memory available. The usage factor corrects for the real core usage (default is 1, i.e. full usage). The PUE (Power Usage Effectiveness) measures how much extra energy is needed to operate the data centre (cooling, lighting etc.). The PSF (Pragmatic Scaling Factor) is used to take into account multiple identical runs (e.g. for testing or optimisation).

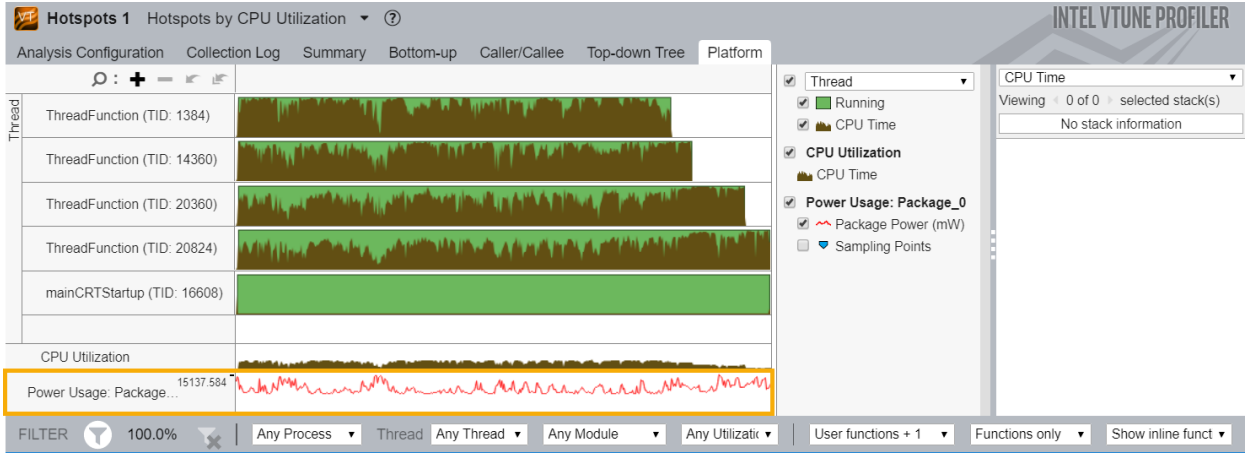
The Carbon Intensity depends on the location and the technologies used to produce electricity. But note that **the "energy needed" indicated at the top of this page is independent of the location.**

Энергоэффективность

1. Run Intel SoC Watch for collecting data

```
socwatch -t 30 -f sys -r vtune -m -o run1
```

2. Read with Intel Vtune:



Спасибо за внимание