Институт прикладной математики им. М.В.Келдыша РАН

Отказоустойчивые расчеты на экзафлопсных вычислительных системах

М.В. Якобовский
lira@imamod.ru

Перспективы использования экзафлопсных вычислительных систем

Текущее состояние

- Относительно малое число использования вычислительных мощностей превышающих 100 TFLOPs
  причины: острый дефицит математических моделей, численных алгоритмов и программных средств для высокопроизводительных вычислительных систем
- Необходимы логически простые и эффективные алгоритмы для современных и для будущих архитектур высокопроизводительных вычислительных систем
- Основные проблемы инвариантны относительно типа используемых вычислительных систем (CPU, GPU)
- Решение на основе фундаментальной науки

Ближайшие перспективы

- Реальная необходимость высокопроизводительных вычислительных систем следующего поколения для решения задач:
  - нефтегазовые проблемы разведки и оптимизации добычи
  - экологические двигатели
  - ядерная энергетика и термоядерный синтез
  - фундаментальные проблемы астрофизики
  - …

- 2015 - Достаточно широкое использование PetaFLOPs (10^{15} операций в секунду) вычислительных систем

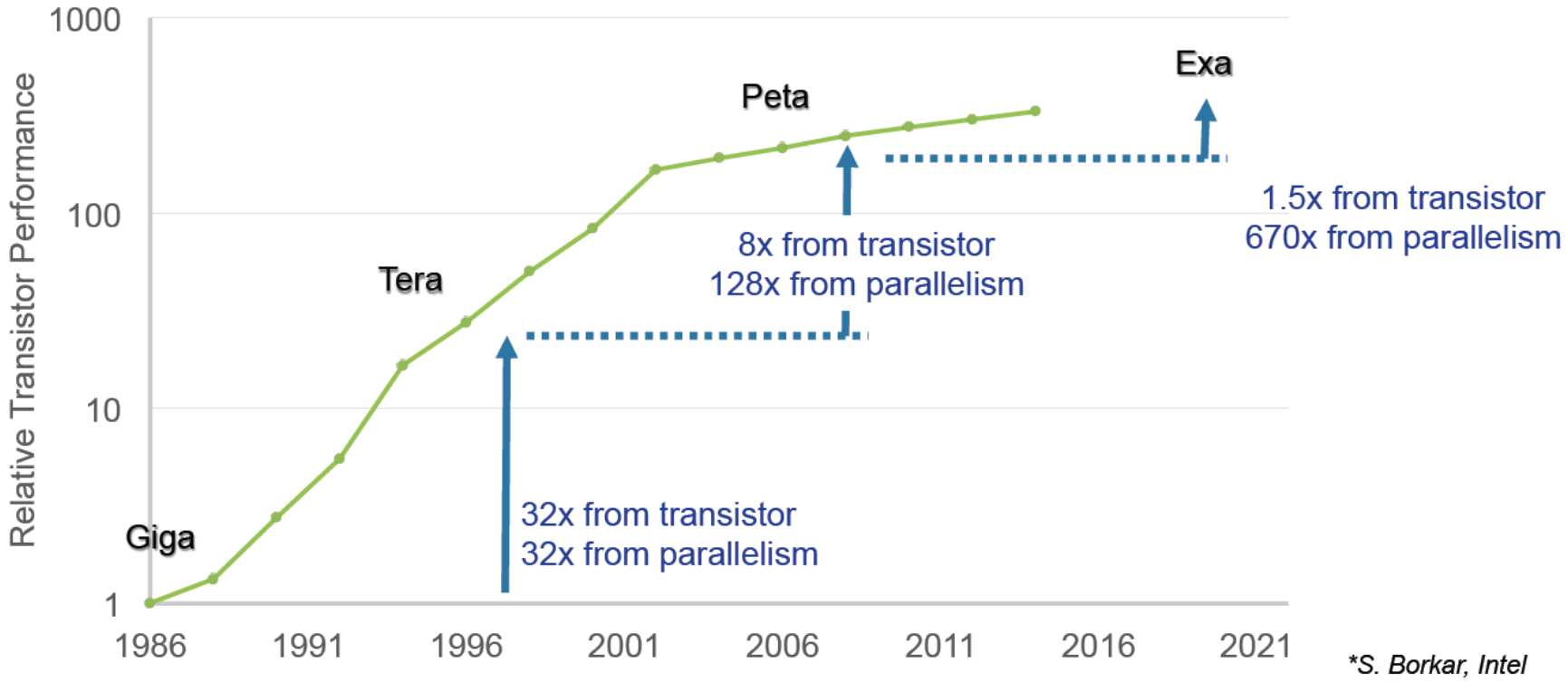
- 2018…2021… - Производительность суперкомпьютеров 1 ExaFLOPs (10^{18} операций в секунду)

July 31, 2016

Paul Messina, Argonne National Laboratory, ECP Director

A Path to Capable Exascale Computing

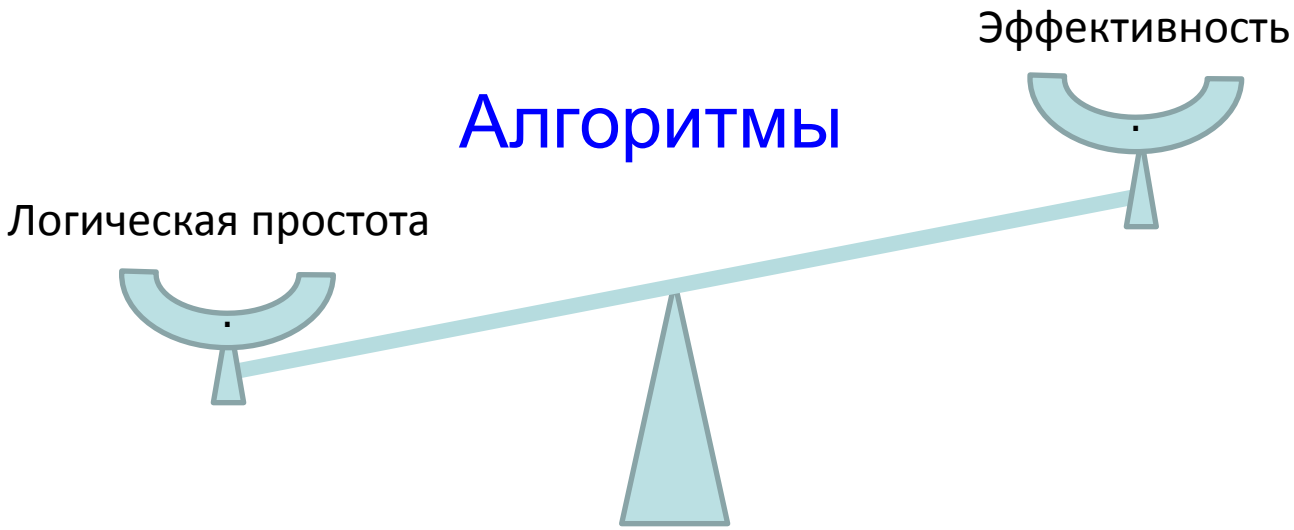
From Giga to Exa, via Tera & Peta*



*S. Borkar, Intel

Performance from parallelism

Алгоритмы



- Явные схемы позволяют создавать логически простые алгоритмы, но имеют строгие ограничения на дискретизацию по времени из условий устойчивости:
  - для параболического типа уравнений условие устойчивости

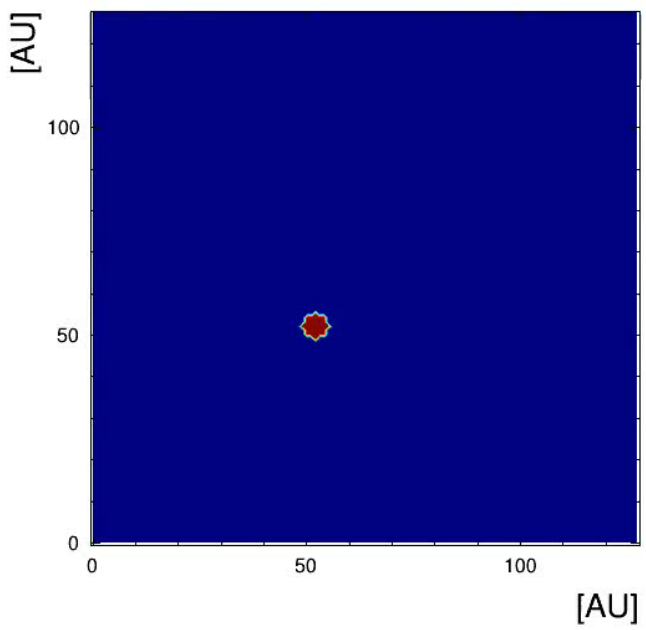
    $\Delta t \le h^2$ – *шаг по времени мал,* практически нельзя использовать высокое разрешение по пространству
  - для уравнений гиперболического типа условие устойчивости

    $\Delta t \le h$ – *можно использовать большой шаг по времени, вычислять в десятки раз быстрее*

    Δt - *шаг дискретизации по времени,* h - *шаг дискретизации по пространству*

Аккреция облака межзвездного газа на компактном астрономическом объекте

низкое разрешение



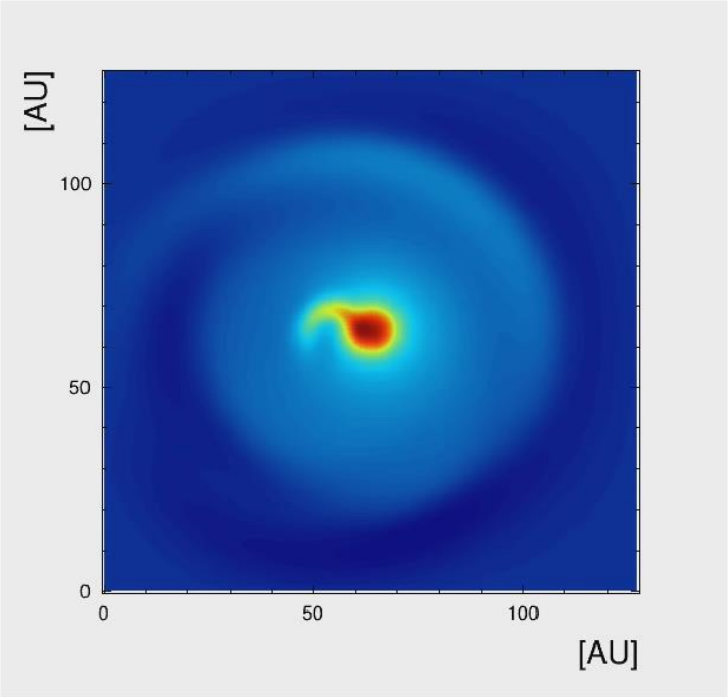
высокое разрешение

- Межзвездное облако 5 AU
- Плотность 0.8 x 10^{-11} кг/m^3
- Скорость облака 300 m/s
- Импакт параметер 4-10 AU

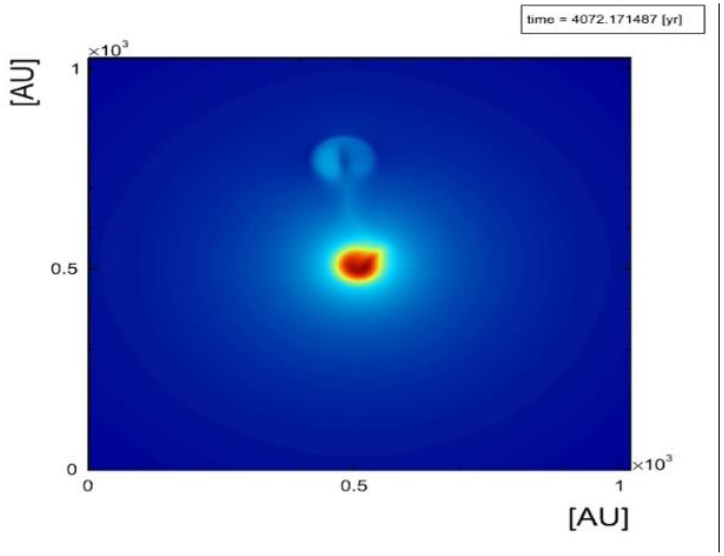
- Компактный обьект:
  - масса 10^{30} Kg
  - радиус 0.5 AU
- Температура пространства T = 20 K

Аккреция облака межзвездного газа на компактном астрономическом объекте

Low resolution



High resolution



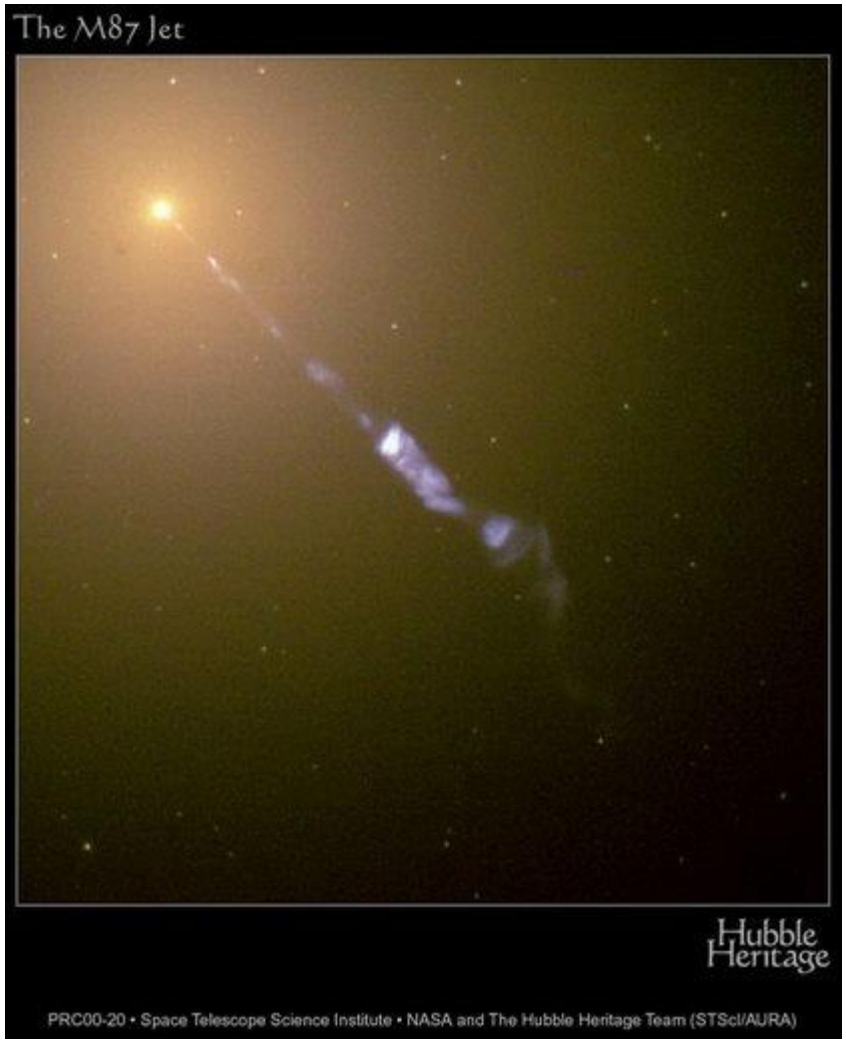
- Межзвездное облако 5 AU
- Плотность 0.8 x 10^{-11} кг/m^3
- Скорость облака 300 m/s
- Импакт параметер 4-10 AU

- Компактный обьект:
  - масса 10^{30} Kg
  - радиус 0.5 AU
- Температура пространства T =~ 20 K

Релятивистская струя галактики M87

Снимок космического телескопа им. Хаббла

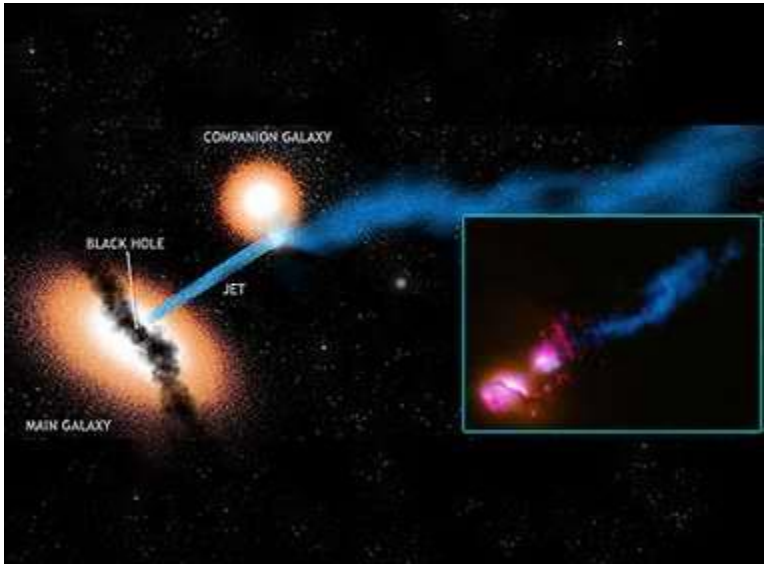
Astronomy Picture of the Day, July 6, 2000
https://apod.nasa.gov/apod/ap000706.html



http://planetarium-kharkov.org/?q=galaxy-CGCG049-033

Струя из одной галактики направлена на соседнюю галактику и оказывает воздействие на ее межзвездную среду

Система двух галактик C321



http://planetarium-kharkov.org/?q=galaxy-CGCG049-033

Cray HLRS – Germany, Stuttgart

- Каждые 4.2 часа фиксируется отказ, требующий восстановления части системы
- Полный отказ системы каждые 160 часов

Di Martino, Catello, Zbigniew Kalbarczyk, Ravishankar K. Iyer, Fabio Baccanico, Joshi Fullop, and William Kramer. "Lessons learned from the analysis of system failures at petascale: The case of blue waters." In*Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*, pp. 610-621. IEEE, 2014.

Время между отказами на экзафлопсных системах ~ 30 минут

Marc Snir, et al. Addressing failures in exascale computing. International Journal of High Performance Computing Applications, 28(2):129–173, May 2014

Частота аппаратных отказов будет возрастать

- Уменьшение размера транзистора делает его менее устойчивым к космической радиации
- Ёмкости меньшего размера содержат меньший заряд, - его проще изменить
- Программное обеспечение становится сложнее и содержит больше ошибок
- Оборудование становится сложнее (неоднородные ядра, многоуровневая иерархия памяти, сложная топология объединения узлов), что существенно усложняет программное обеспечение
- *Мультифизичность и многомасштабность* решаемых задач приводит к объединению большого числа программных модулей
- Сокращение обменов, использование асинхронных взаимодействий, обеспечение защищённости от отказов оборудования приводит к созданию сложных прикладных кодов

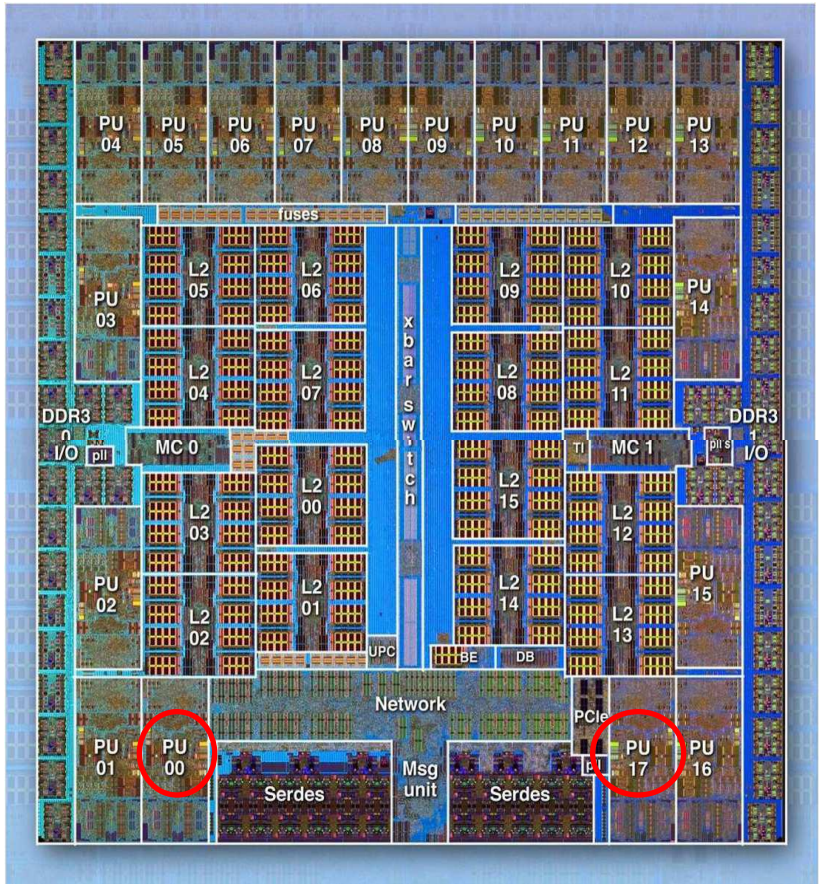
Cappello, F., Geist, A., Gropp, W., Kale, S., Kramer, B., & Snir, M. (2014). Toward exascale resilience: 2014 update. *Supercomputing frontiers and innovations*, *1*(1).

Время создания контрольной точки ~ 30 минут

System from TOP 500	Max performance	Checkpoint time (minutes)
LLNL Zeus Lawrence Livermore National Laboratory	11 TeraFLOPS	26
LLNL BlueGene/L	500 TeraFLOPS	20
Argonne BlueGene/P	500 TeraFLOPS	30
LANL RoadRunner **Los Alamos Natlional Labs**	1 PetaFLOPS	~ 20

Cappello F. 2009. Fault Tolerance in Petascale/ Exascale Systems: Current Knowledge, Challenges and Research Opportunities. International Journal of High Performance Computing Applications 23, 3, 212–226.

IBM PowerPC® A2 1.6 GHz, 16 cores per node



Robert W. Wisniewski.

BlueGene/Q: Architecture,

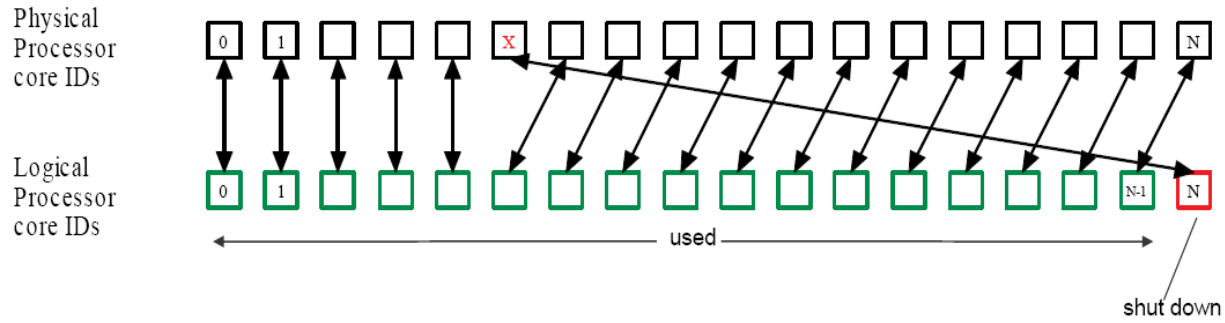
CoDesign; Path to Exascale / Blue

Gene Supercomputer Research,

January 25, 2012

There are two spare cores here.
One core performs service functions.
One core is idle.

Cores are renumbered.
The idle one includes into work.



Уровни управления контрольными точками

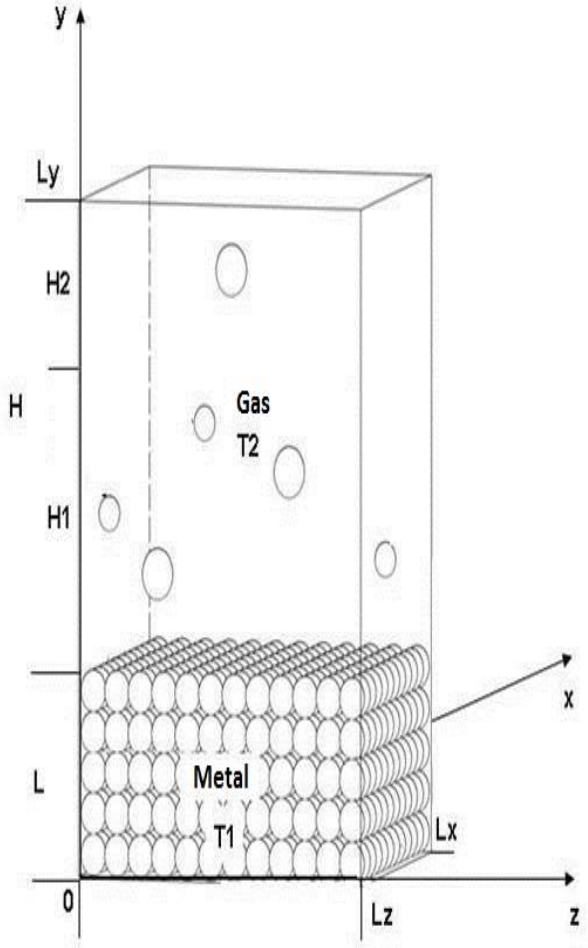
- Системный уровень
  - Простота использования
- Уровень пользователя
  - Радикальное сокращение объёмов контрольных точек
  - Вместо рестарта всей системы - замена вычислительного узла
  - Хранение данных не только на локальных дисках HDDs но и в оперативной памяти

Численное моделирование

- Масштабное МД моделировании:
  - - 5 вариантов расчетов
  - - 3 разных вычислительных ресурса:
    - MVS10-P (MSC RAS)
    - K1 (NICEVT)
    - IMM6 (KIAM RAS)

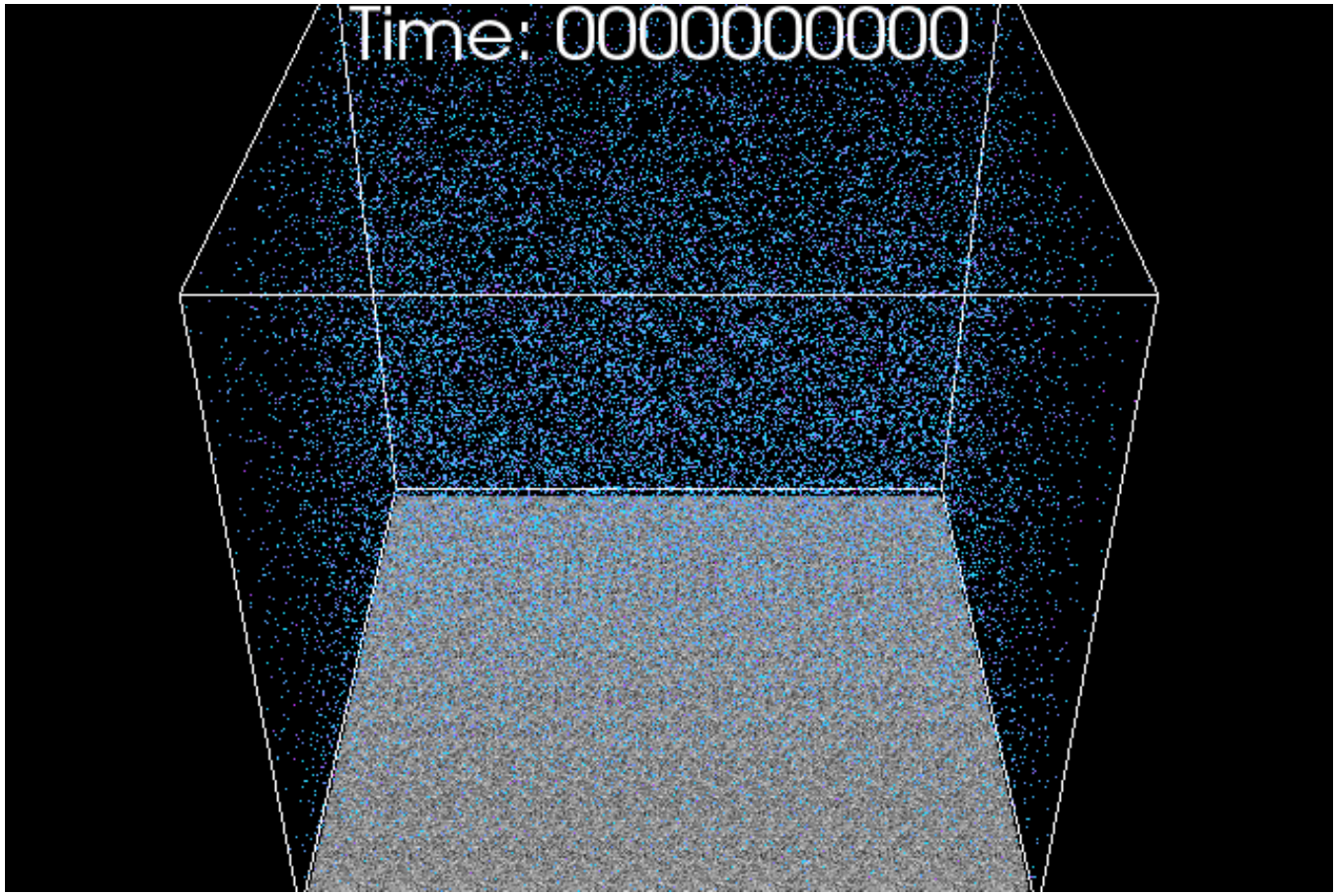
Проблемы относительно управления задачами:

- Ручной запуск и мониторинг задач
- Ручная переброска данных с одного ресурса на другой
- Квота на дисковое пространство



Моделирования взаимодействия Ni-N2

Size: 8 128 512 + 423 840 = 8 552 352 particles,
Temperature T_{Ni} = 273.15 K, T_{N2} = 273.15 K



The problem is split into gas dynamics and molecular dynamics:
Flow and Particles

Fault-tolerant environments for checkpointing

Automatic (based on BLCR) system level checkpoint :

- MPICH, MVAPICH, OpenMPI

Semi-automatic, user level checkpoint :

- C^3 - Cornell Checkpoint pre-Compiler, (Greg Bronevetsky, Daniel Marques, …)
- ULFM(FT-MPI)

Egwutuoha, I.P. A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems. / I.P. Egwutuoha, D. Levy, B. Selic, S. Chen // The Journal of Supercomputing. — 2013. — Vol. 65, No.3. —P. 1302-1326.

Cappello, F. Fault tolerance in petascale/exascale systems: Current knowledge, challenges and research opportunities // International Journal of High Performance Computing Applications. — 2009. — Vol. 23, No. 3. — P. 212–226

ULFM - User-Level Failure Mitigation

Current MPI 3.1 itself provides no mechanisms for handling processor failures.

ULFM is designed according to be the minimal interface necessary to restore the complete MPI capability to transport messages after failures.

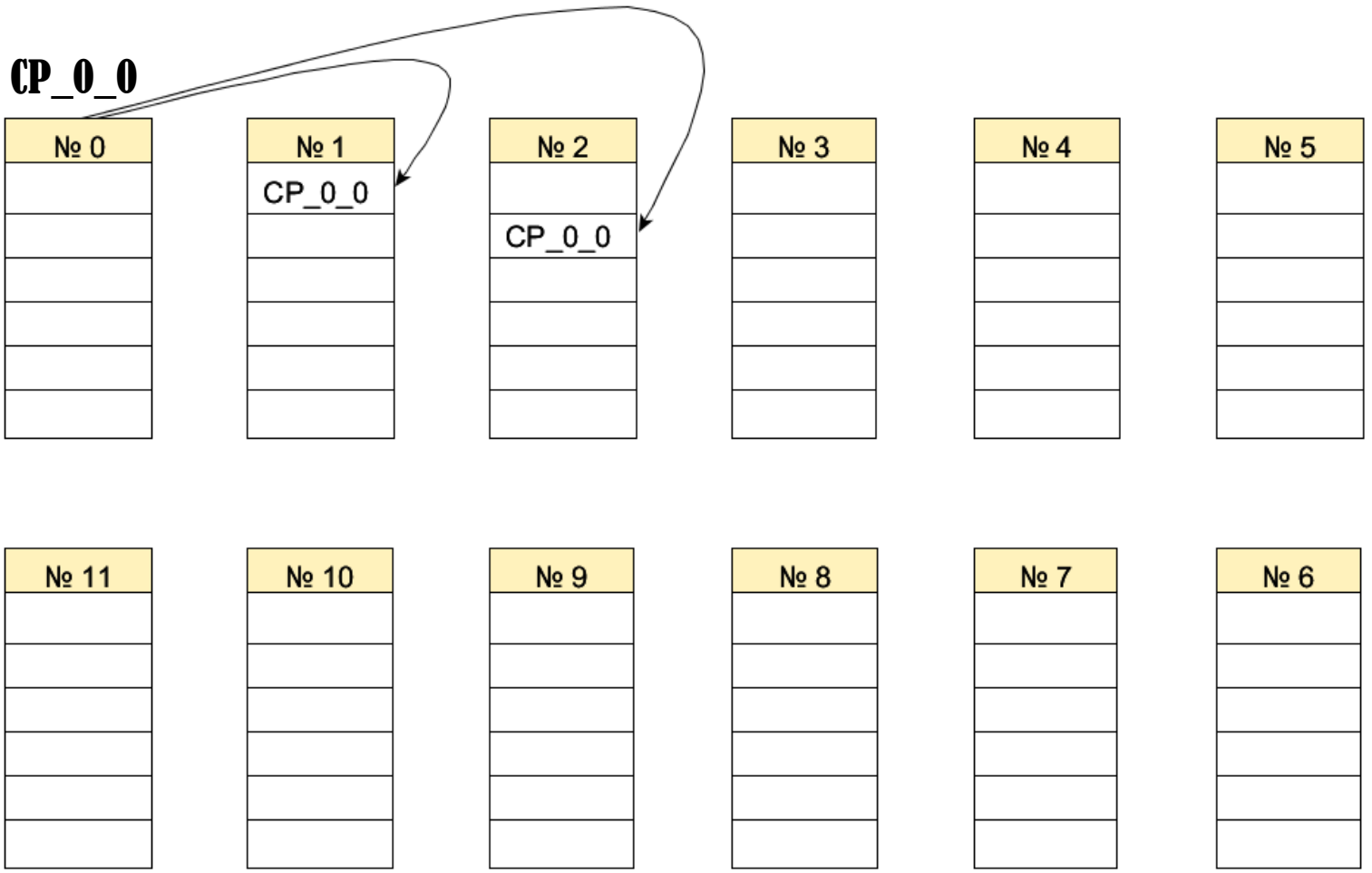
ULFM functions:

- **MPI_COMM_REVOKE**
- **MPI_COMM_SHRINK**
- **MPI_COMM_FAILURE_GET_ACKED**
- **MPI_COMM_FAILURE_ACK**
- **MPI_COMM_AGRE**

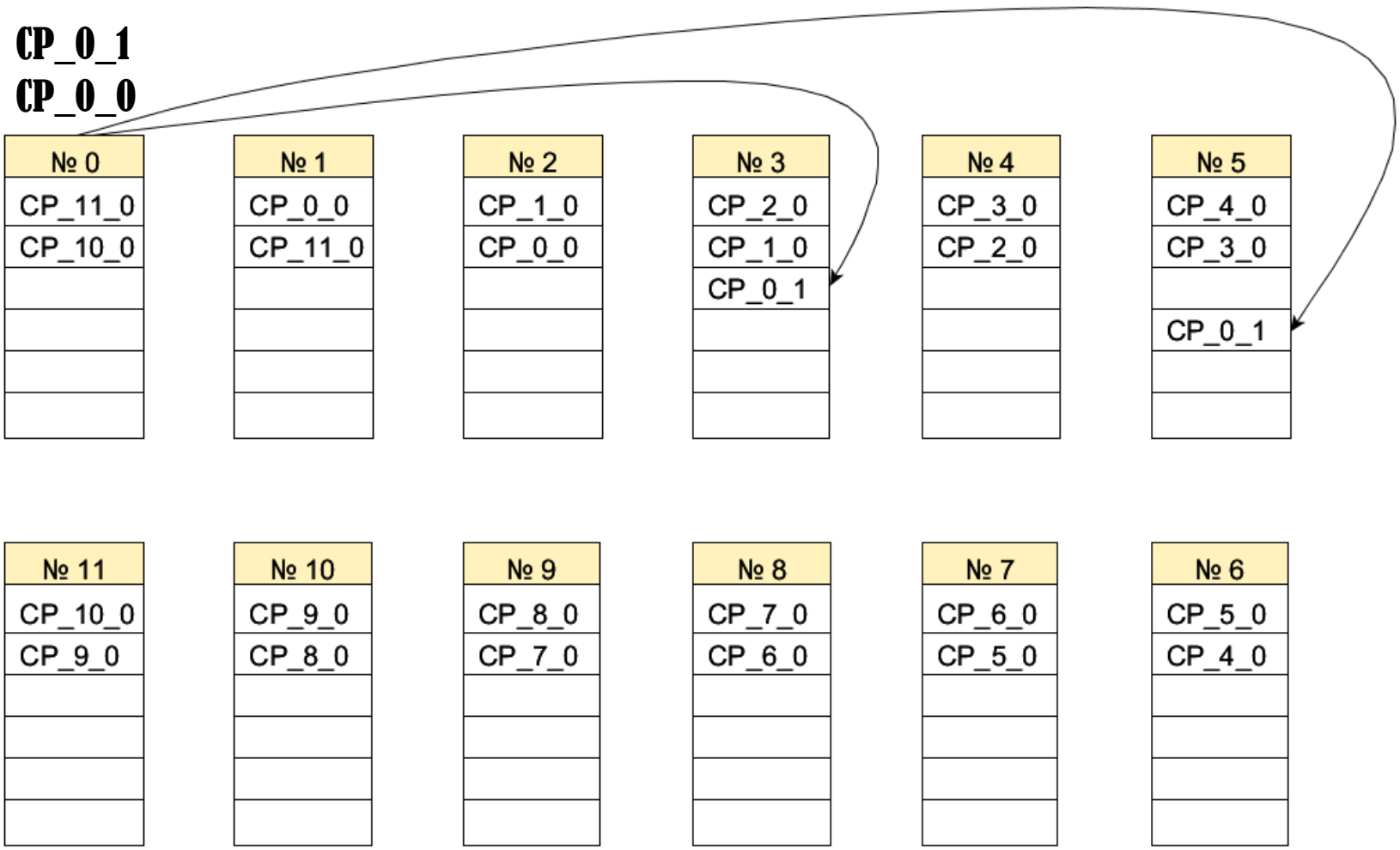
http://fault-tolerance.org/

ULMF is a part of new version of MPI (MPI 4.1)

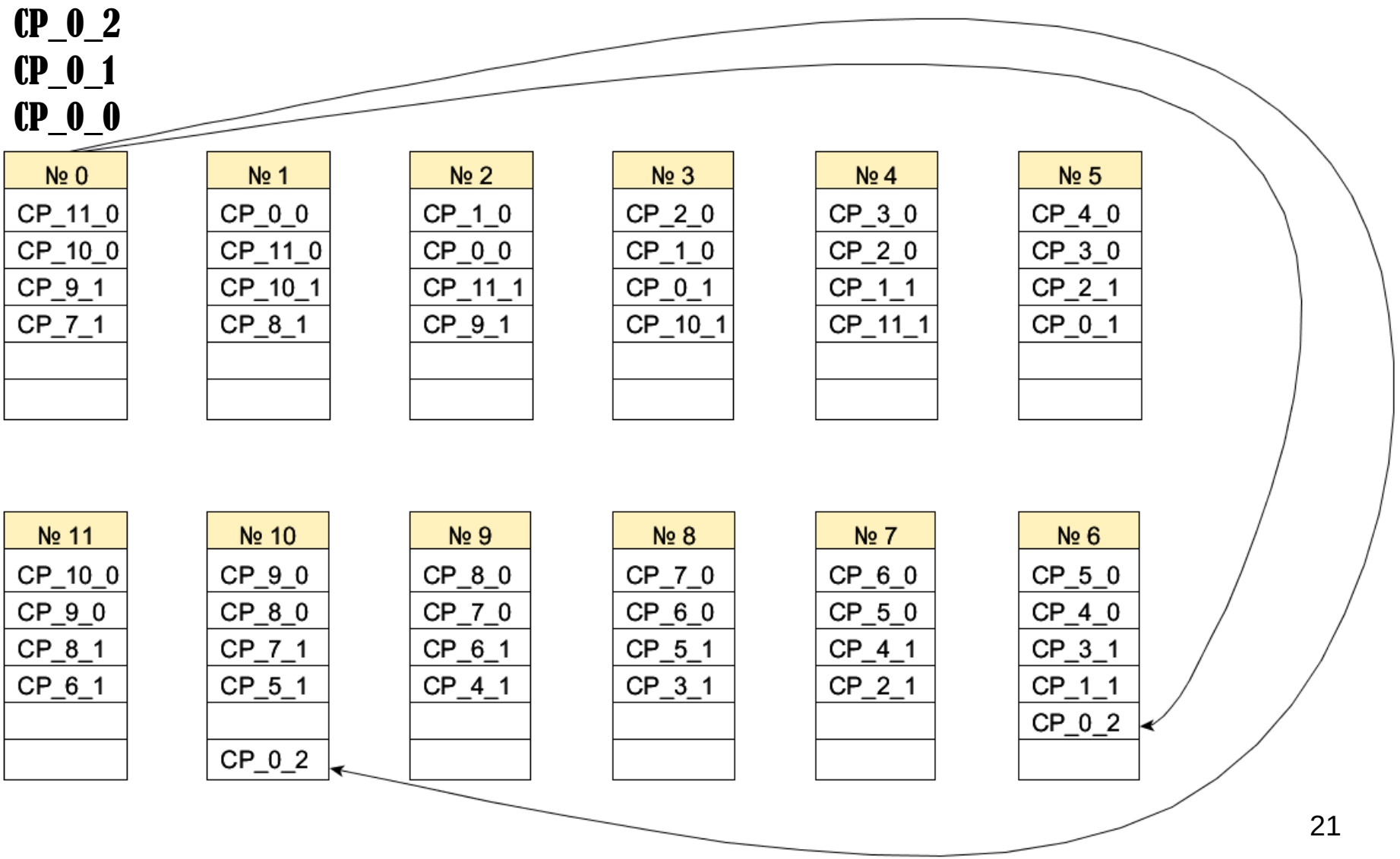
First control point



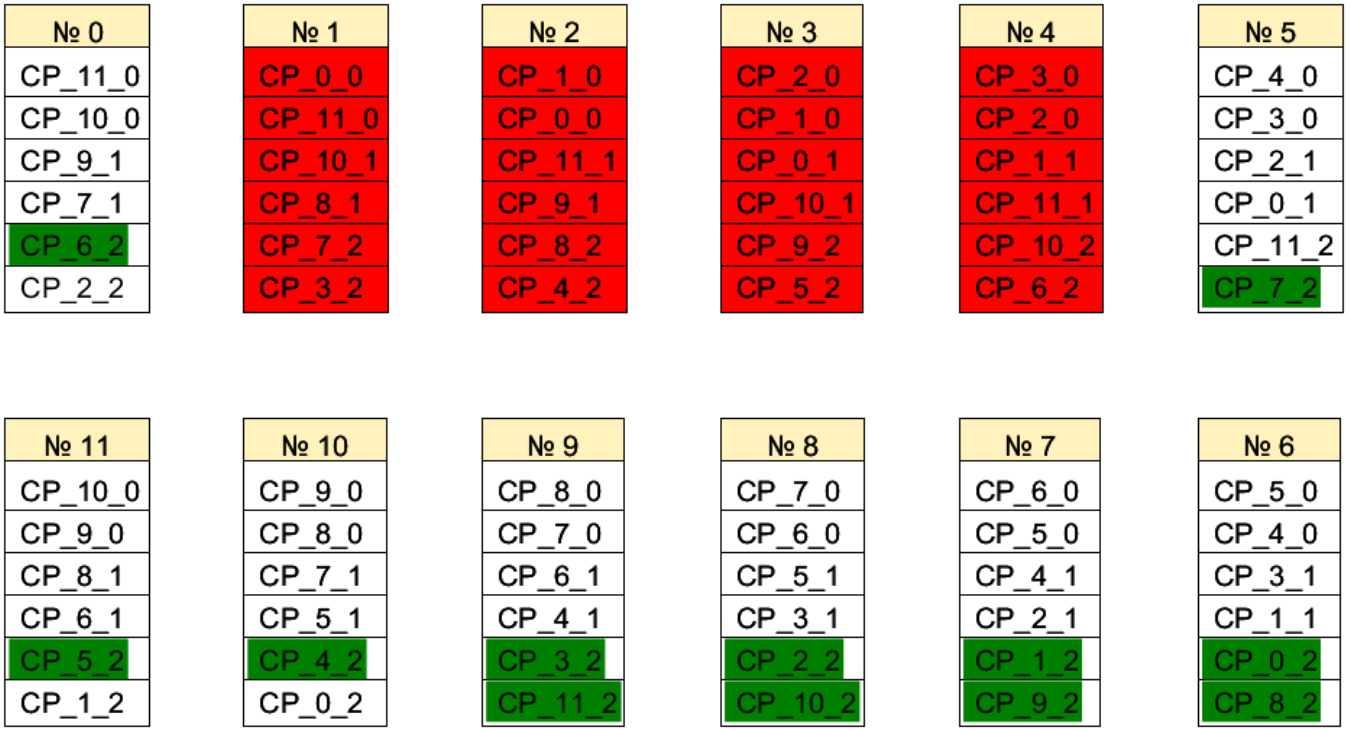
Second control point



Third control point



Even if four processors fail, it will be possible to resume the calculation



But, all processors will have to repeat the calculation of some steps

Недостатки контрольных точек

- Большое время записи глобальной контрольной точки
- Неснижаемая потеря времени на повторный расчет шагов, выполненных после записи контрольной точки
- Необходимость многократного пересчета одних и тех же данных при повторных отказах

- Время расчета увеличивается в случае возникновения отказа

HPC вызов

- Разработка принципов управления контрольными точками, при которых время накладных расходов меньше чем MTBF
- Разработка алгоритмов, дающих возможность продолжать расчет даже при регулярных отказах части процессов
- Обеспечение независимости времени расчета от возникновения отказов, в том числе множественных

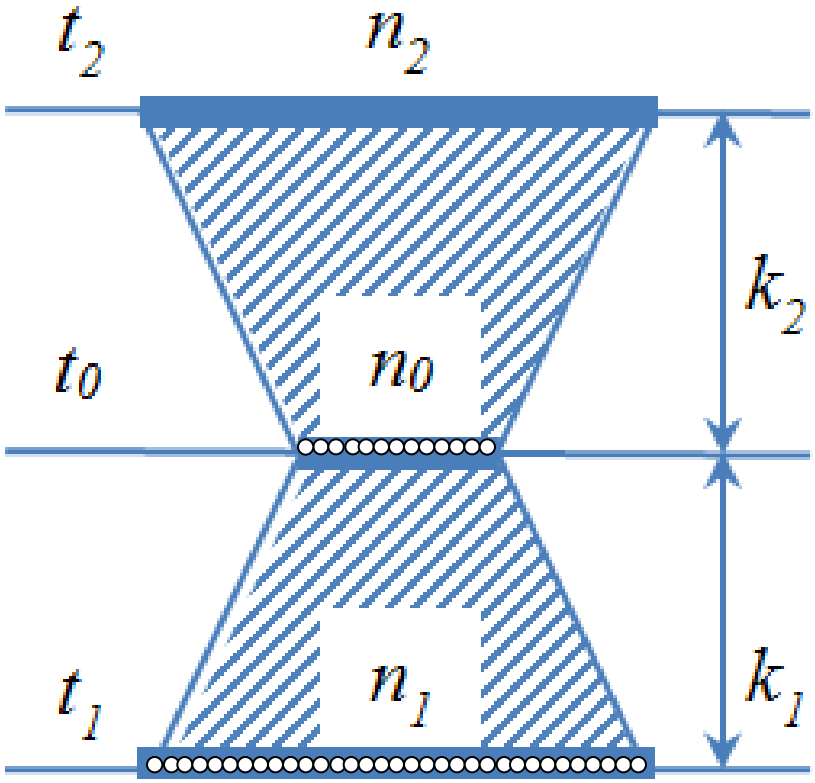
Одномерное гиперболическое уравнение

$$\frac{\partial^2 \Phi}{\partial x^2} - \frac{1}{C^2}\frac{\partial^2 \Phi}{\partial t^2} = F(x,t)$$

Две характеристики $x - ct$ и $x + ct$, определяющими область, влияющую на решение $\Phi(x,t)$ в точке (x,t)

Геометрические размеры области на момент $(t - \Delta t)$, определяющие $\Phi(x,t)$,

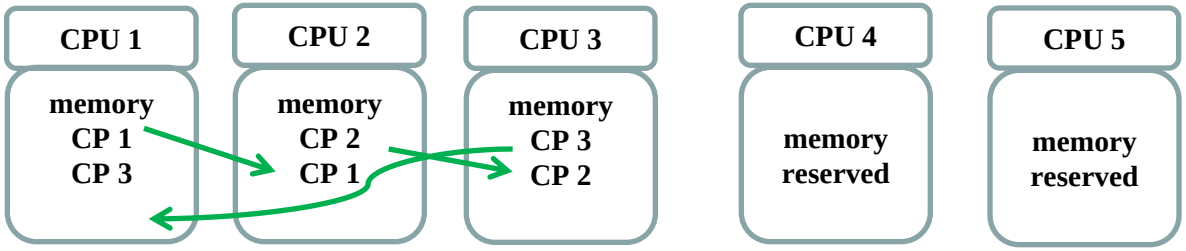
The domain n_1 allows to recover the data lost due to a processor failure using accelerated recalculations



The data in the domain n_1 determine the solution $\Phi(x,t)$ in the domain n_0 at the time t_0.

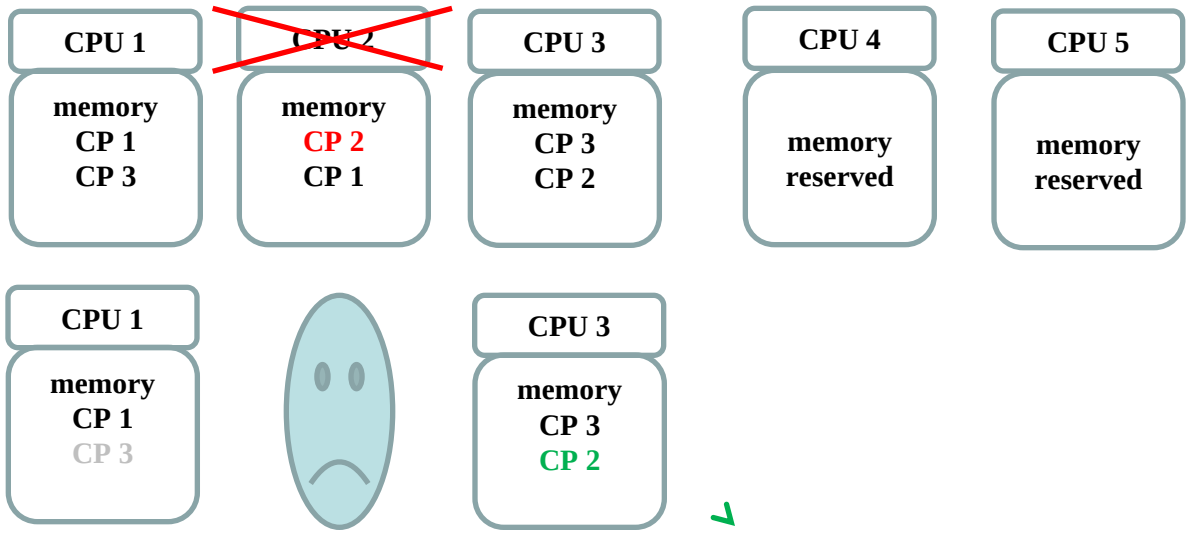
Fault tolerance approach

If we do not want to rollback we must have several copies.
We store them in the local memory of other working processors.



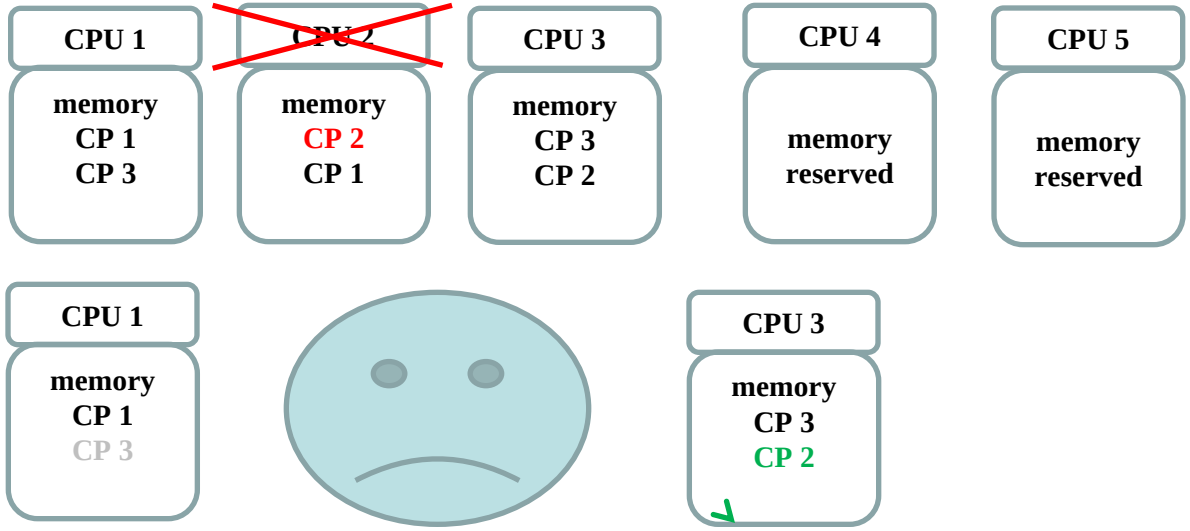
Fault tolerance approach

If we do not want to rollback we must have several copies.
We store them in the local memory of other working processors.



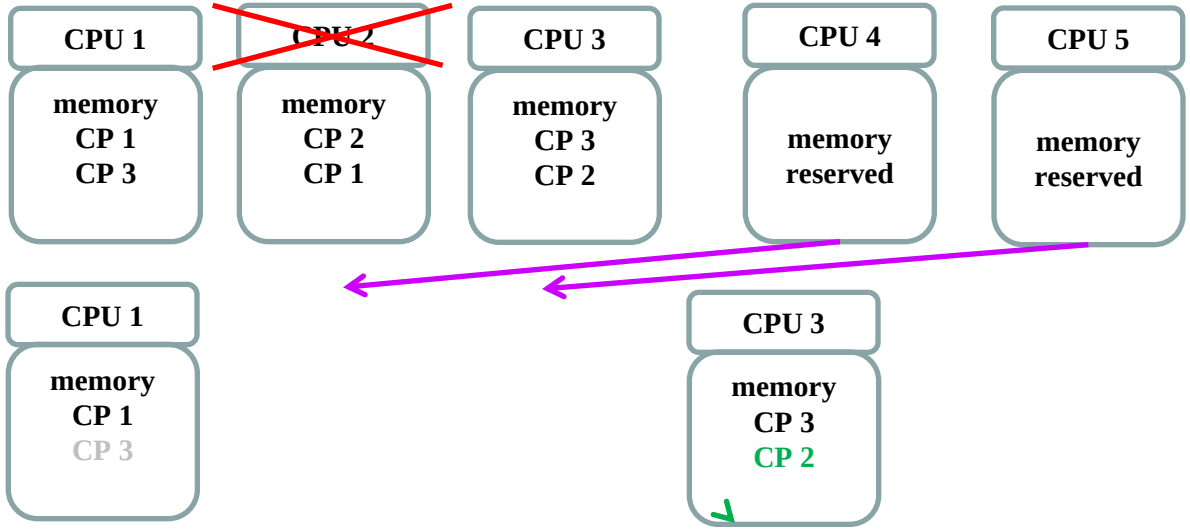
Fault tolerance approach

If we do not want to rollback we must have several copies.
We store them in the local memory of other working processors.



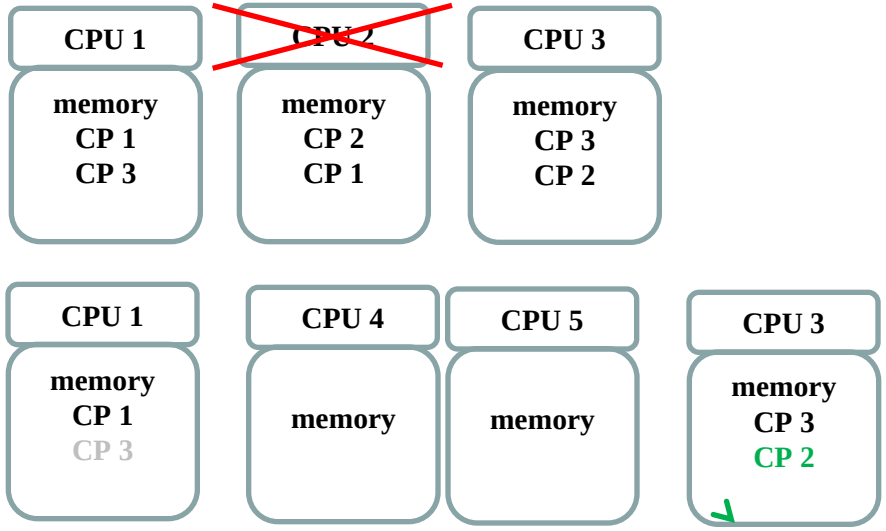
Fault tolerance approach

If we do not want to rollback we must have several copies.
We store them in the local memory of other working processors.



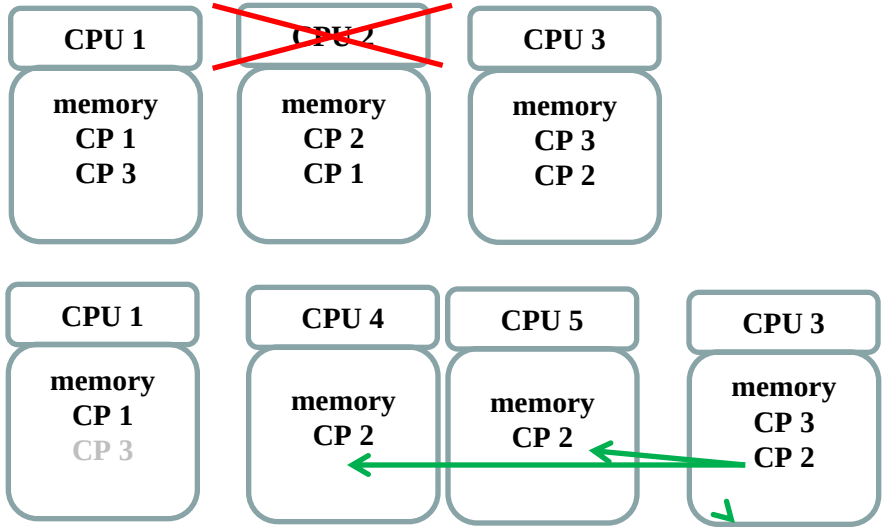
Fault tolerance approach

If we do not want to rollback we must have several copies.
We store them in the local memory of other working processors.



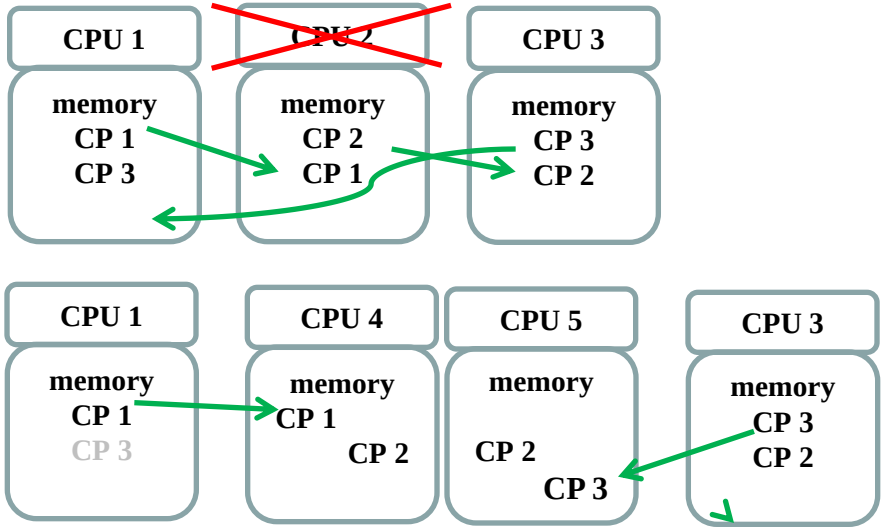
Fault tolerance approach

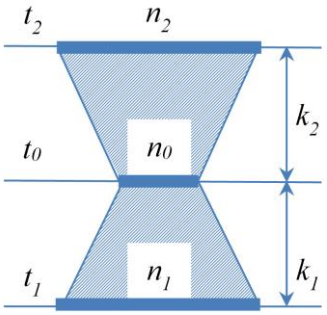
If we do not want to rollback we must have several copies.
We store them in the local memory of other working processors.



Fault tolerance approach

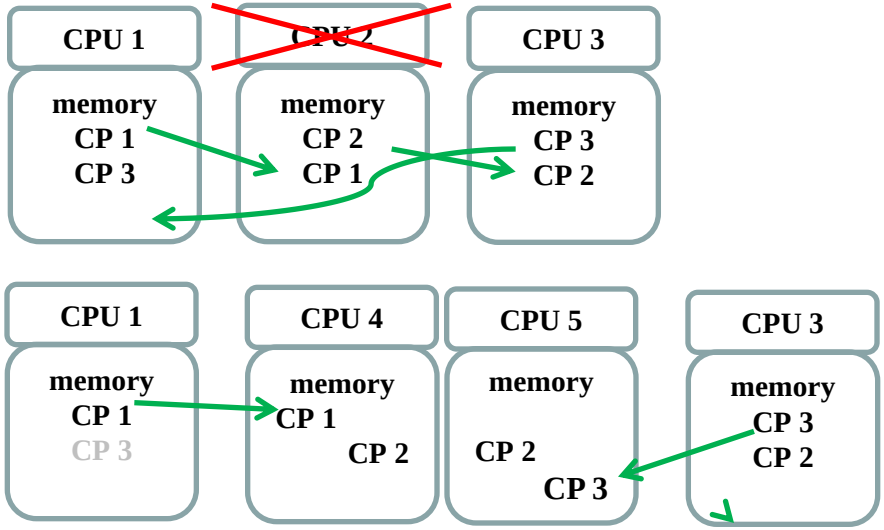
If we do not want to rollback we must have several copies.
We store them in the local memory of other working processors.

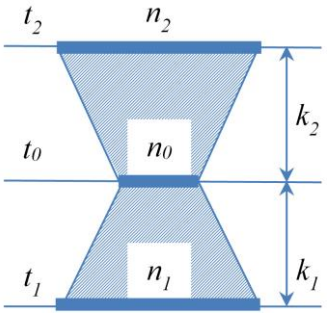




Fault tolerance approach

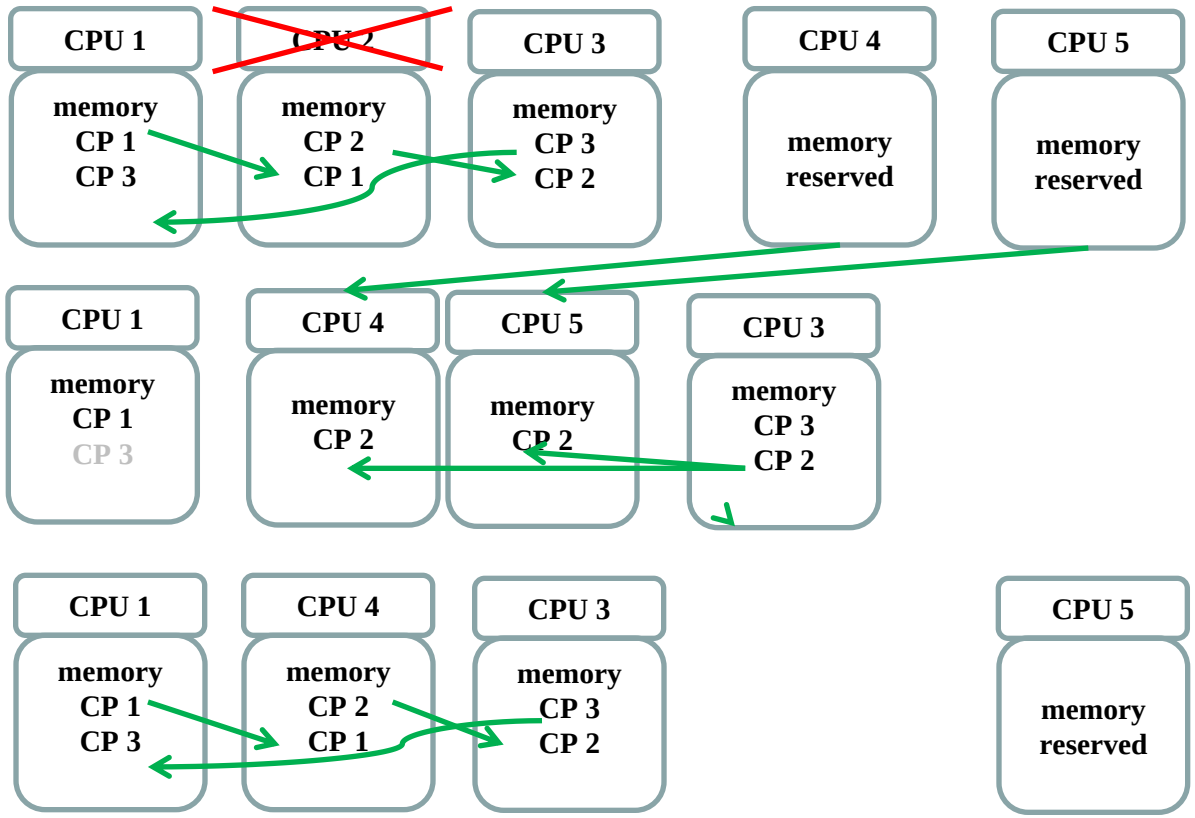
If we do not want to rollback we must have several copies.
We store them in the local memory of other working processors.





Сохранение контрольных точек локально – в память вычислительных узлов

Исключение необходимости перезапуска всех процессоров



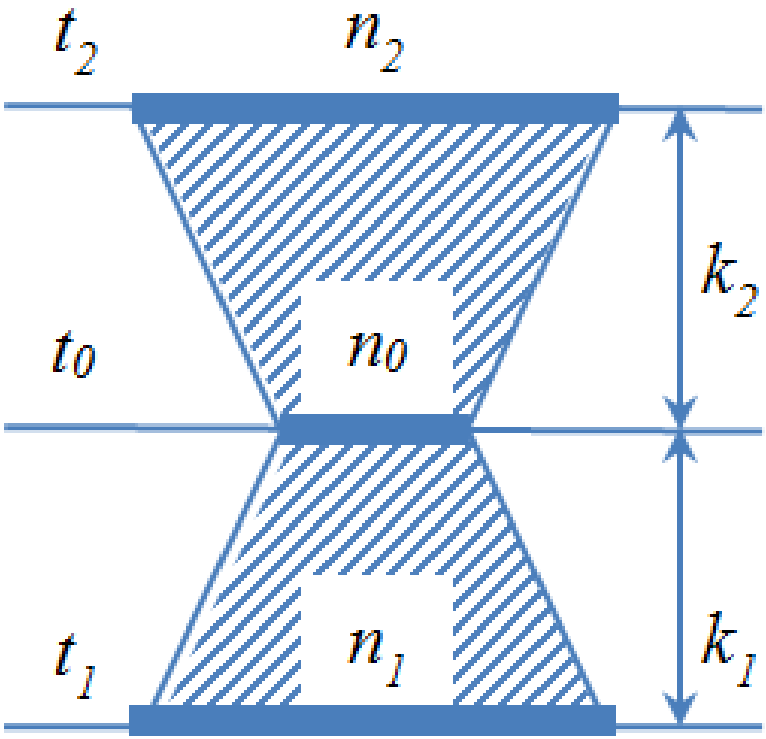
Одномерное гиперболическое уравнение

$$\frac{\partial^2 \Phi}{\partial x^2} - \frac{1}{C^2}\frac{\partial^2 \Phi}{\partial t^2} = F(x,t)$$

Две характеристики $x - ct$ и $x + ct$, определяющими область, влияющую на решение $\Phi(x,t)$ в точке (x,t)

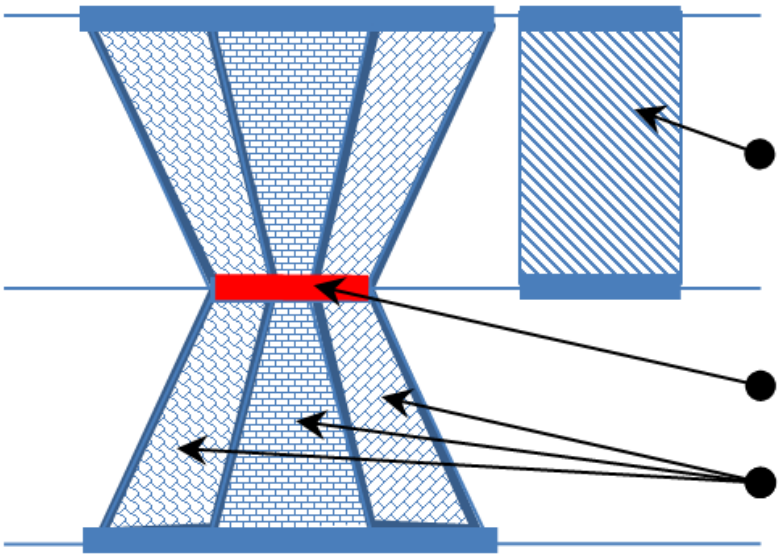
Геометрические размеры области на момент $(t - \Delta t)$, определяющие $\Phi(x,t)$,

Область ускоренного расчета при возмещении потери данных, вызванной выходом из строя одного процессора



Подход применим для гиперболических систем и для любых явных разностных схем

Замена неисправного процессора тремя запасными



Полное восстановление расчета

Продолжение работы исправных процессоров

Момент поломки процессора

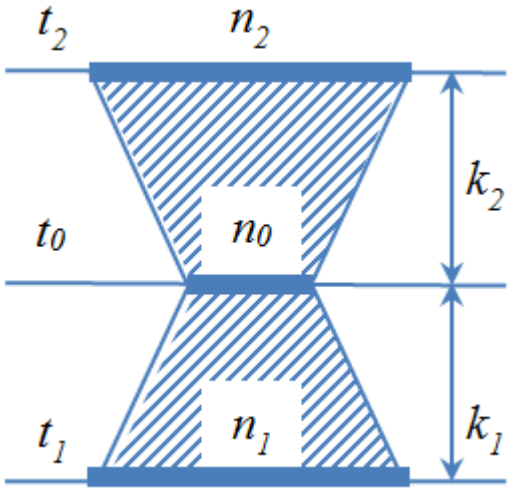
Утерянная часть расчетной сетки

Перерасчет части данных

Последняя локальная контрольная точка

Оценка числа дополнительных процессоров

$$p_d > \frac{1}{d+1}\frac{1}{k_2}\sum_{j=1}^{2} k_j \sum_{i=0}^{d} {\alpha_j}^i$$



$$\alpha_j = 1 + 2\gamma\frac{k_j}{n_0}$$

$\gamma = \frac{C\Delta t}{h}$ - число Куранта

d – *размерность пространства*

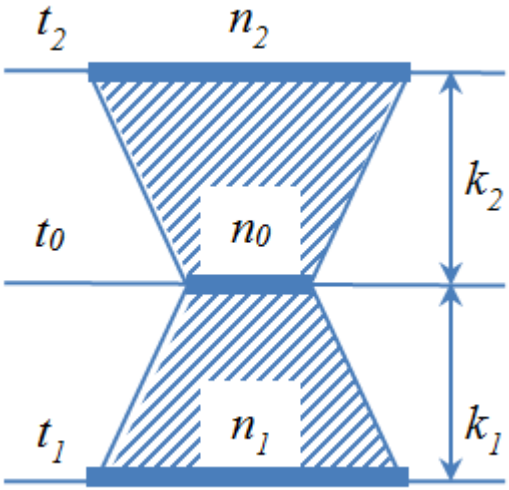
n_0^d - *число точек обрабатываемых одним процессором*

Estimate for the number of additional processors required for recalculation (single processor failure) $k_1 = k_2$

$$p_d > \frac{2}{d+1}\sum_{i=0}^{d}\alpha^i$$

$$\alpha = 1 + 2\gamma\frac{k_1}{n_0}$$

$$\gamma = \frac{C\Delta t}{h}$$ - Courant number



n_0^d – the number of calculation points initially processed by each processor

$d = 1, 2\ or\ 3$ – dimension of the simulated space

How much is α ?
Let the γ be equal to 1

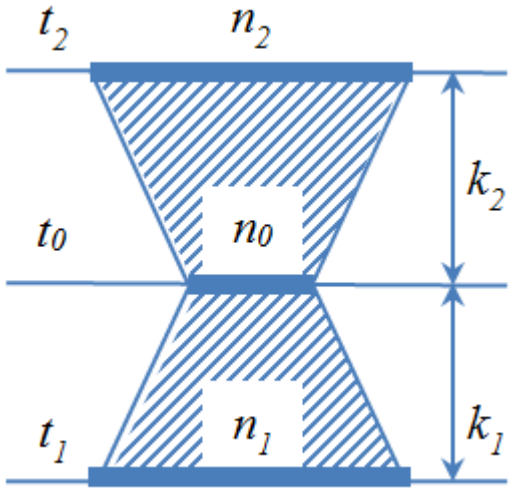
$$p_d \geq \frac{2}{d+1}\sum_{i=0}^{d}\alpha^i$$

$$\alpha = 1 + 2\frac{k_1}{n_0}$$

$$n_0 = \sqrt[d]{n}$$

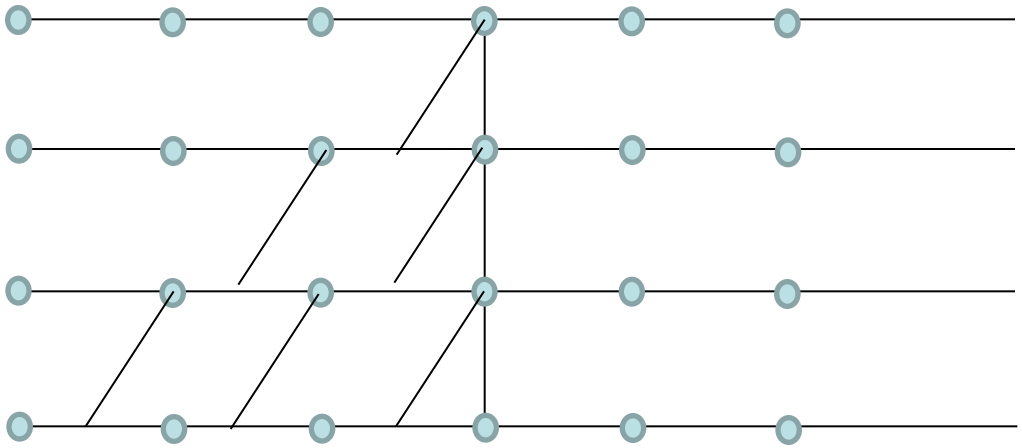
$$\alpha \approx 1$$

$$p_d \geq 2$$



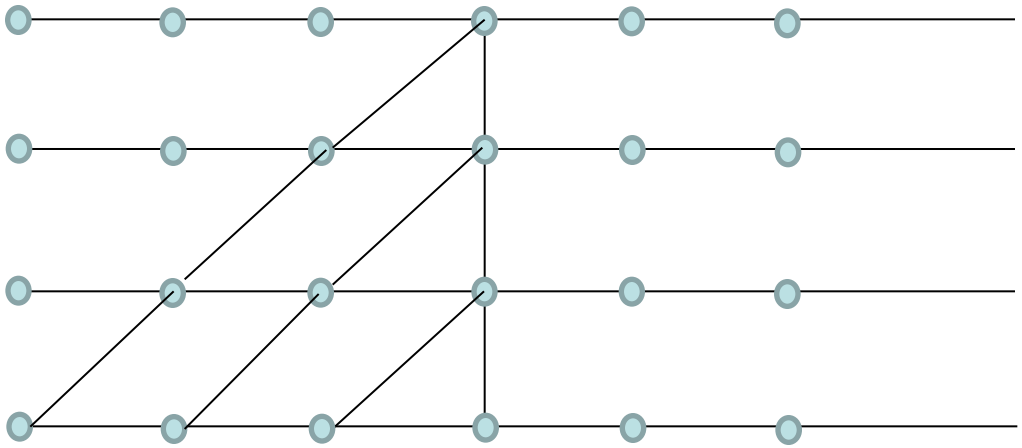
Let the γ be equal to 1

$\gamma < 1 - approximate\ result$



$\gamma = 1$ – matching result

- Only hyperbolic?
  - No, any explicit scheme



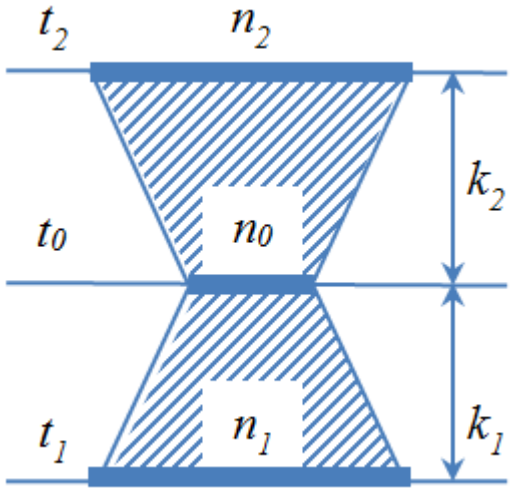
- Each step increases the zone of dependence by one cell

How many additional processors do we need?

$$\gamma = \frac{C\Delta t}{h}$$

$$p_d \geq \frac{2}{d+1}\frac{1}{k_2}\sum_{j=1}^{2} k_j \sum_{i=0}^{d} \alpha_j^i$$

$$n_0 = \sqrt[d]{n}$$



$$\alpha = 1 + 2\gamma\frac{k_j}{n_0}$$

$$\alpha \approx 1$$

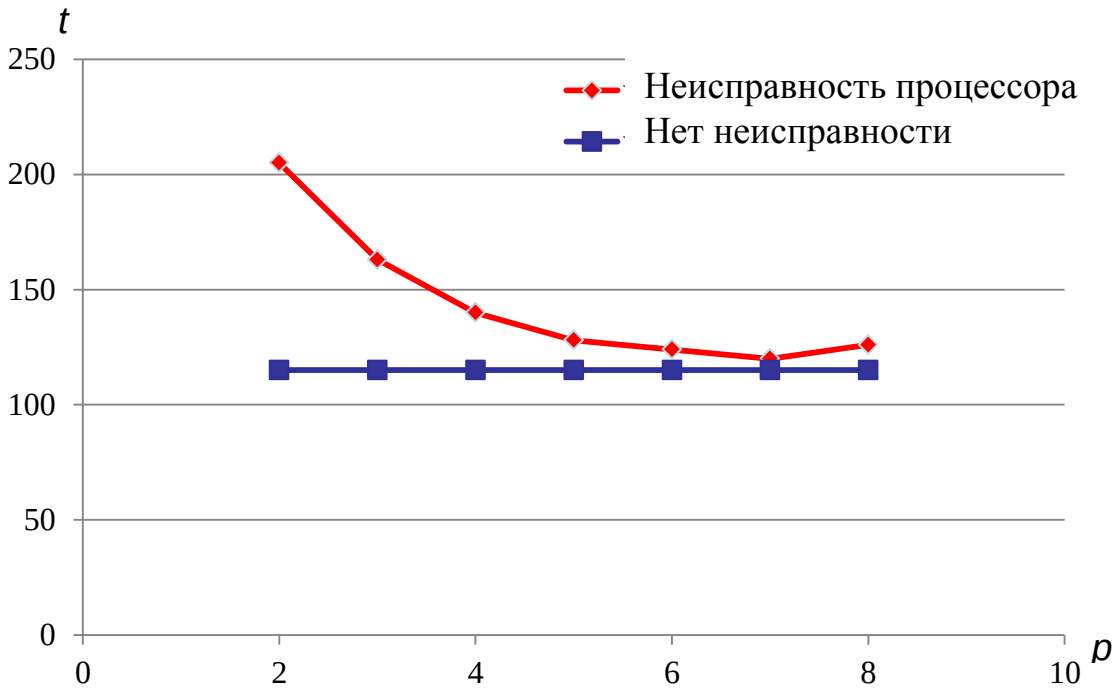
$$p_d \geq 2$$

n_0^d – the number of calculation points initially processed by each processor

$d = 1, 2\ or\ 3$ – dimension of the simulated space

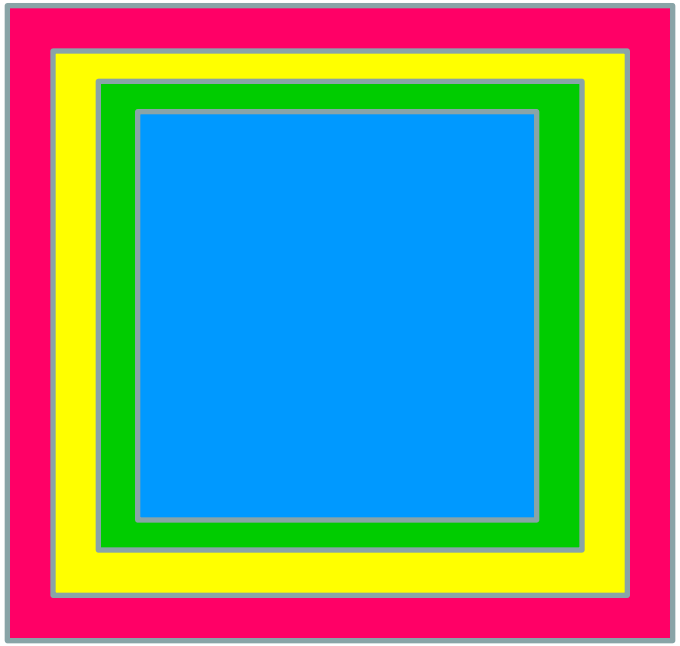
Вывод

Предложен метод, обеспечивающий независимость времени расчета от факта возникновения отказов, в том числе множественных, отличающийся низким уровнем накладных расходов



Additional part is only the border

$$p_d \geq 2$$



Заключение

- *Эффективное использование высокопроизводительных вычислительных систем требует создания качественно новых алгоритмов решения прикладных задач и средств описания и создания параллельных программ*
- *Рассмотренные методы и инструментальные средства позволяют существенно повысить, как эффективность использования суперкомпьютеров, так и эффективность создания широкого круга параллельных приложений*
- *Дальнейшее развитие возможно на пути тесного взаимодействия специалистов по вычислительным методам, прикладному и системному программированию*

Контакты

Якобовский М.В., чл.-корр. РАН, проф., д.ф.-м.н.

Заместитель директора по научной работе
Института прикладной математики
им. М.В.Келдыша Российской академии наук

mail: lira@imamod.ru

web: http://lira.imamod.ru