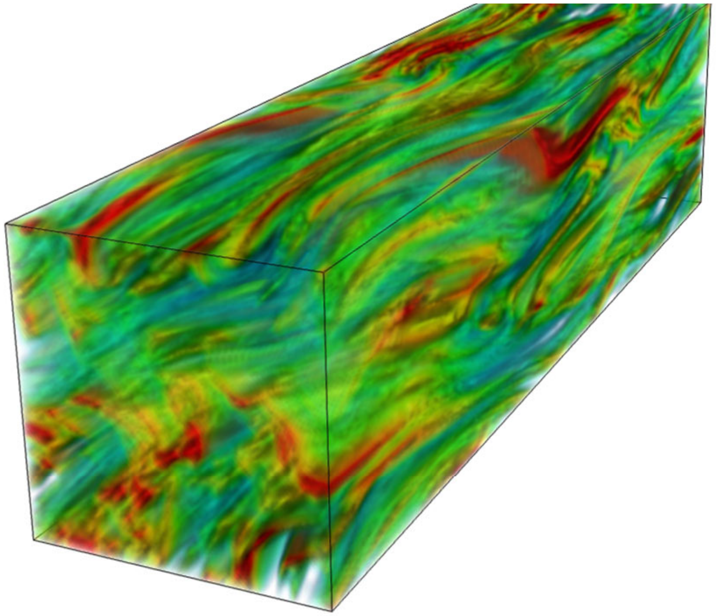
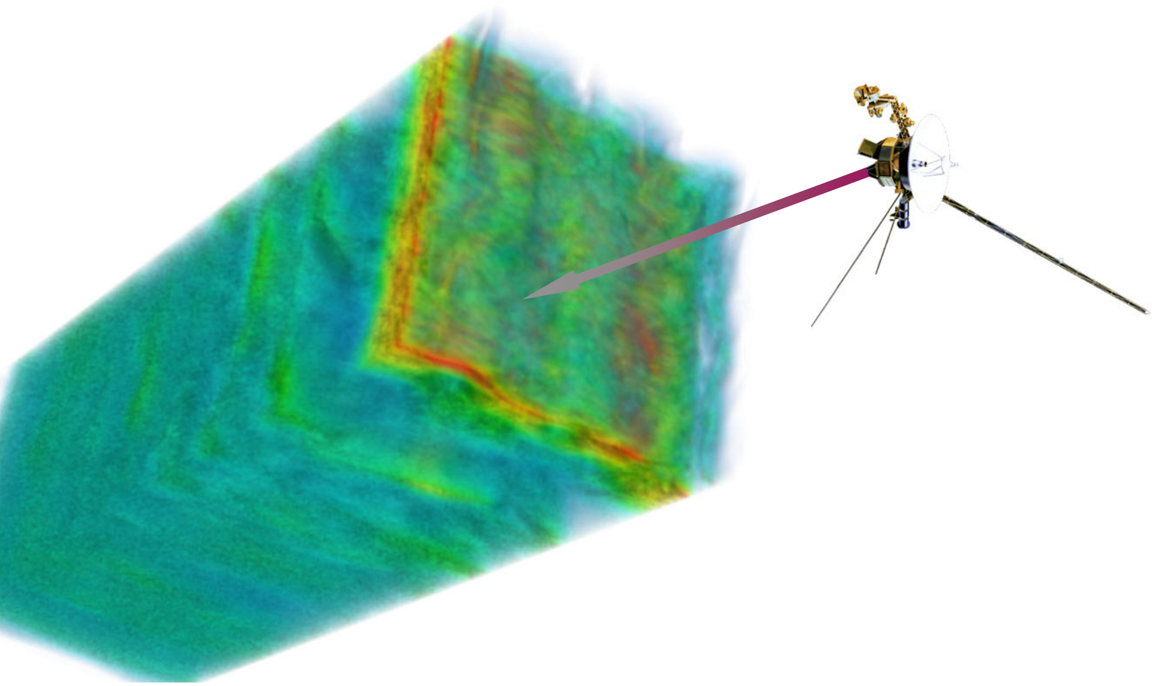
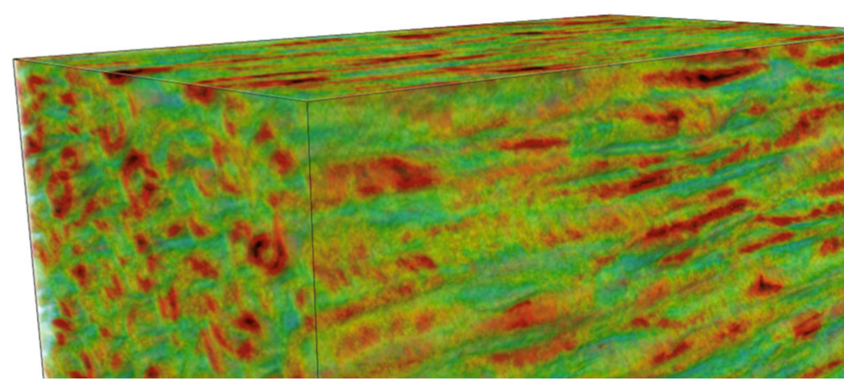


Kinetic Simulations of Collisionless Plasmas

Rahul Kumar

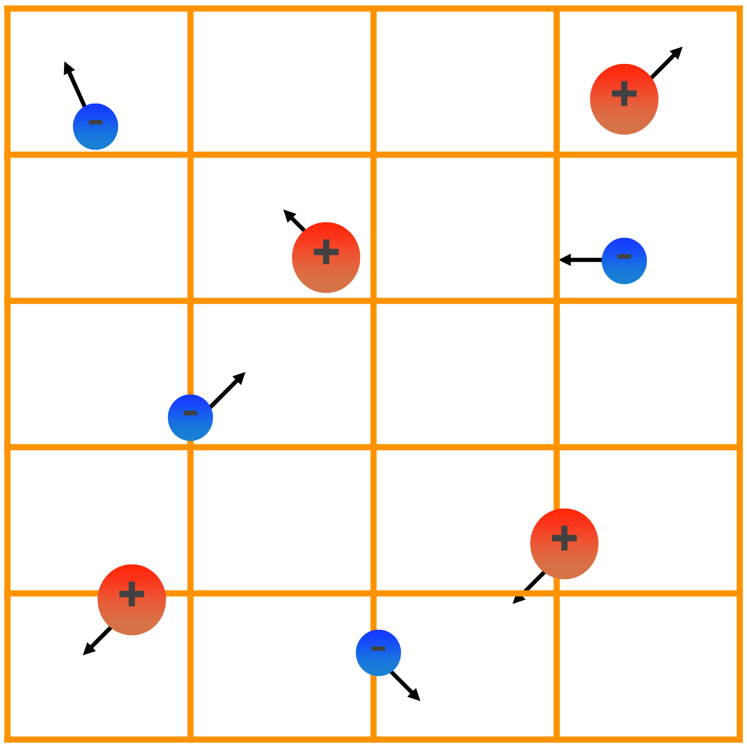
PRINCETON UNIVERSITY





Plasma Physics on Computers

Particle-in-cell (PIC) method



1950s-60s - Early development

Dawson, Hockney, Buneman, and others

1970s - Electrostatic PIC, methods develop

1980s-90s - EM PIC method gains ground

Maxwell's equations

$$\nabla \cdot \mathbf{B} = 0 \qquad \nabla \times \mathbf{B} = \frac{1}{c}\left(4\pi\mathbf{J} + \frac{\partial \mathbf{E}}{\partial t}\right)$$

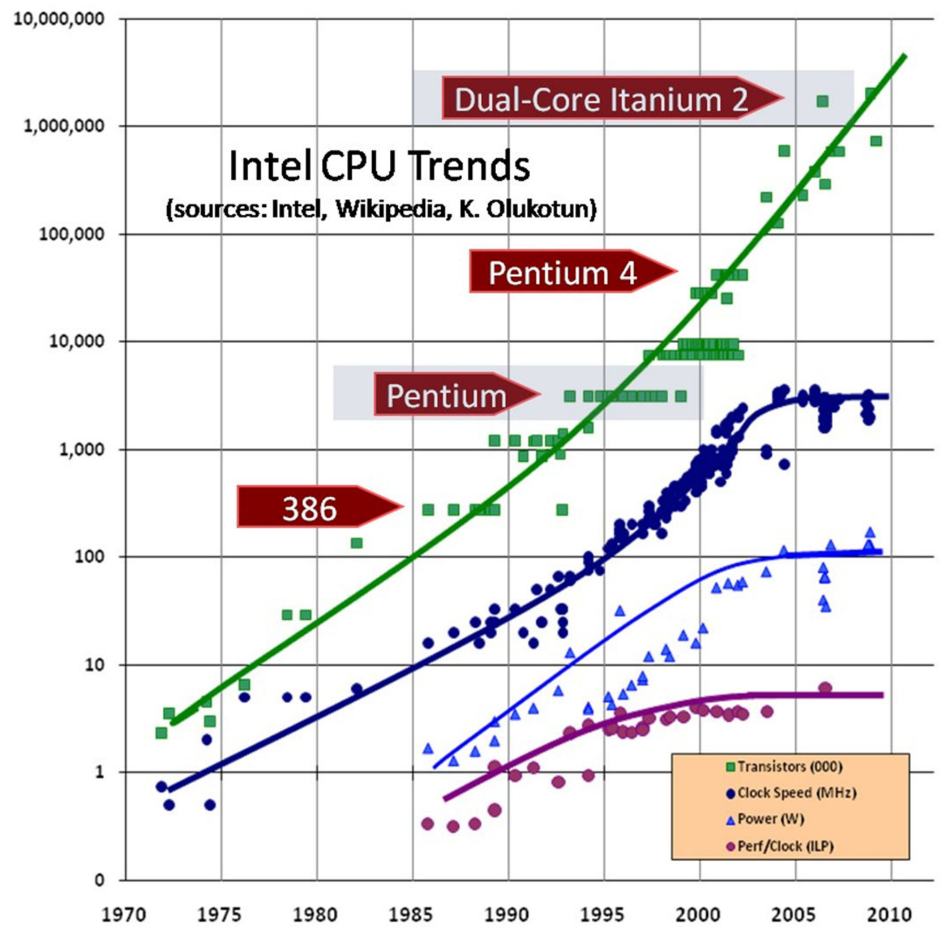
$$\nabla \cdot \mathbf{E} = 4\pi\rho \qquad \nabla \times \mathbf{E} = -\frac{1}{c}\frac{\partial \mathbf{B}}{\partial t}$$

Lorentz force

$$\vec{F} = q(\vec{E} + \frac{\vec{v}}{c} \times \vec{B})$$

Computing Architectures

Marching Forward



Trend

Individual Processing Units are not getting any faster

More Processing Units are available

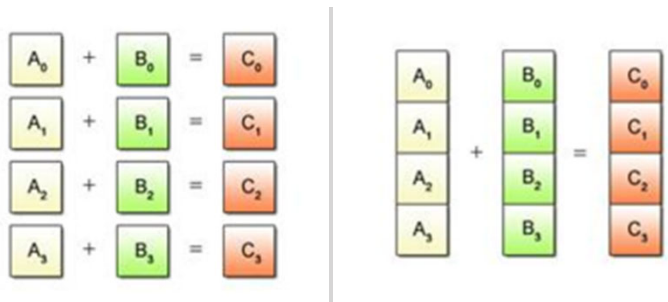
image source: web

Computing Architecture

New programming challenges

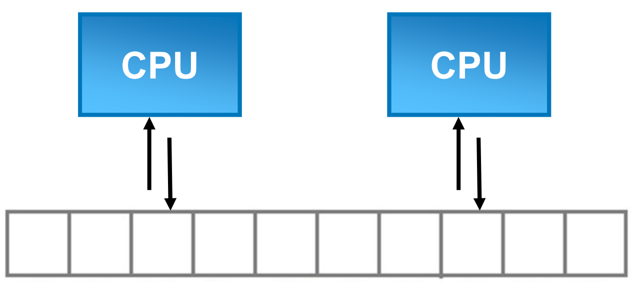
>> Processing more data in fewer instructions/cycles (SIMD)

scalar operation



vector operation

>> Independent tasks



MPI, OpenMP

>> data accessibility

Cache

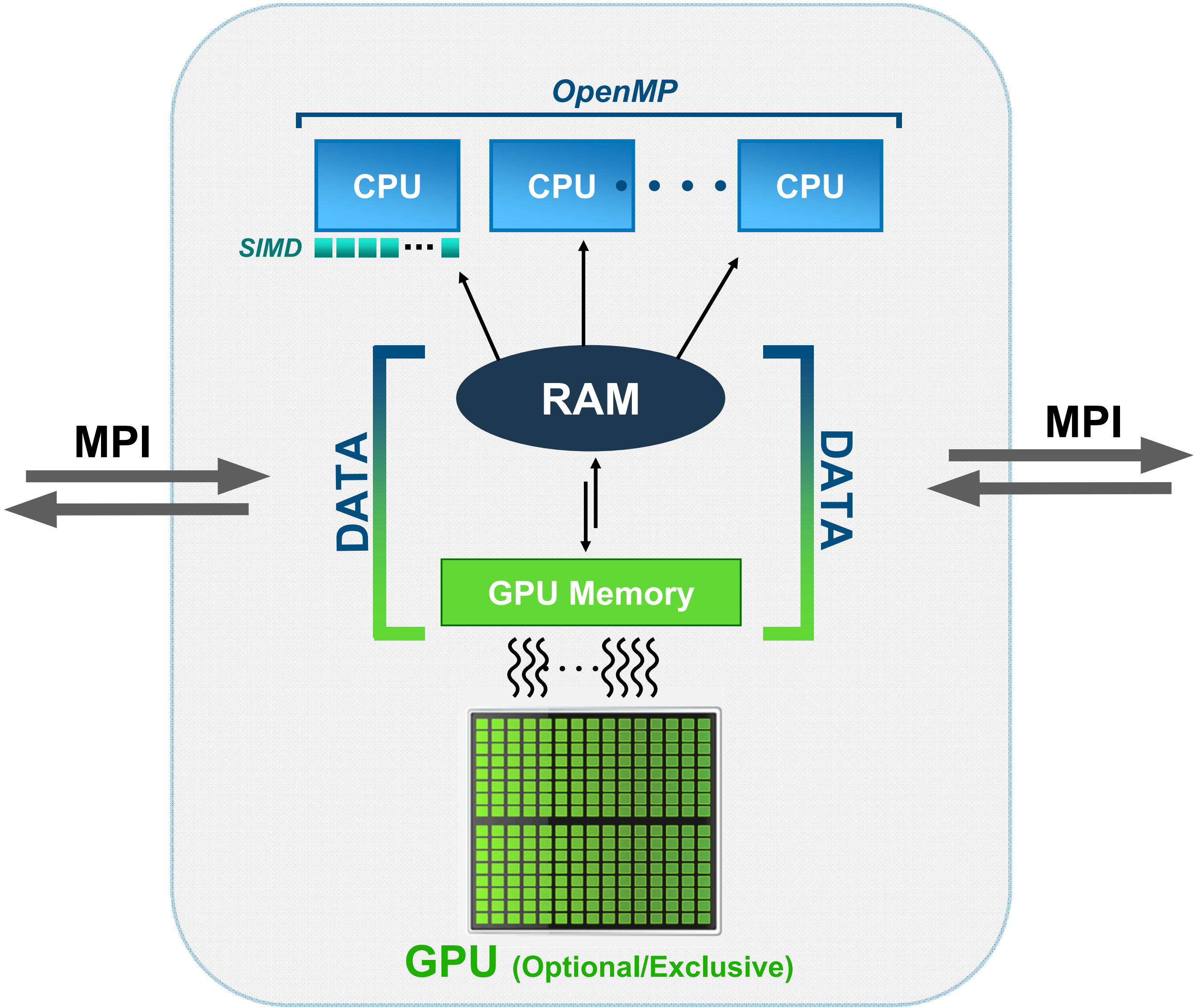
RAM

Disk

PICTOR

Particle-in-Cell Code

https://github.com/rahulbgu/PICTOR



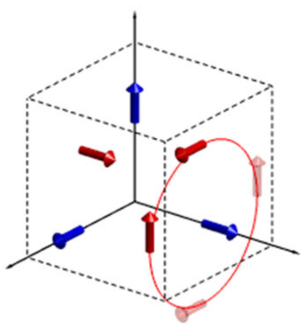
PICTOR

Particle-in-Cell Code

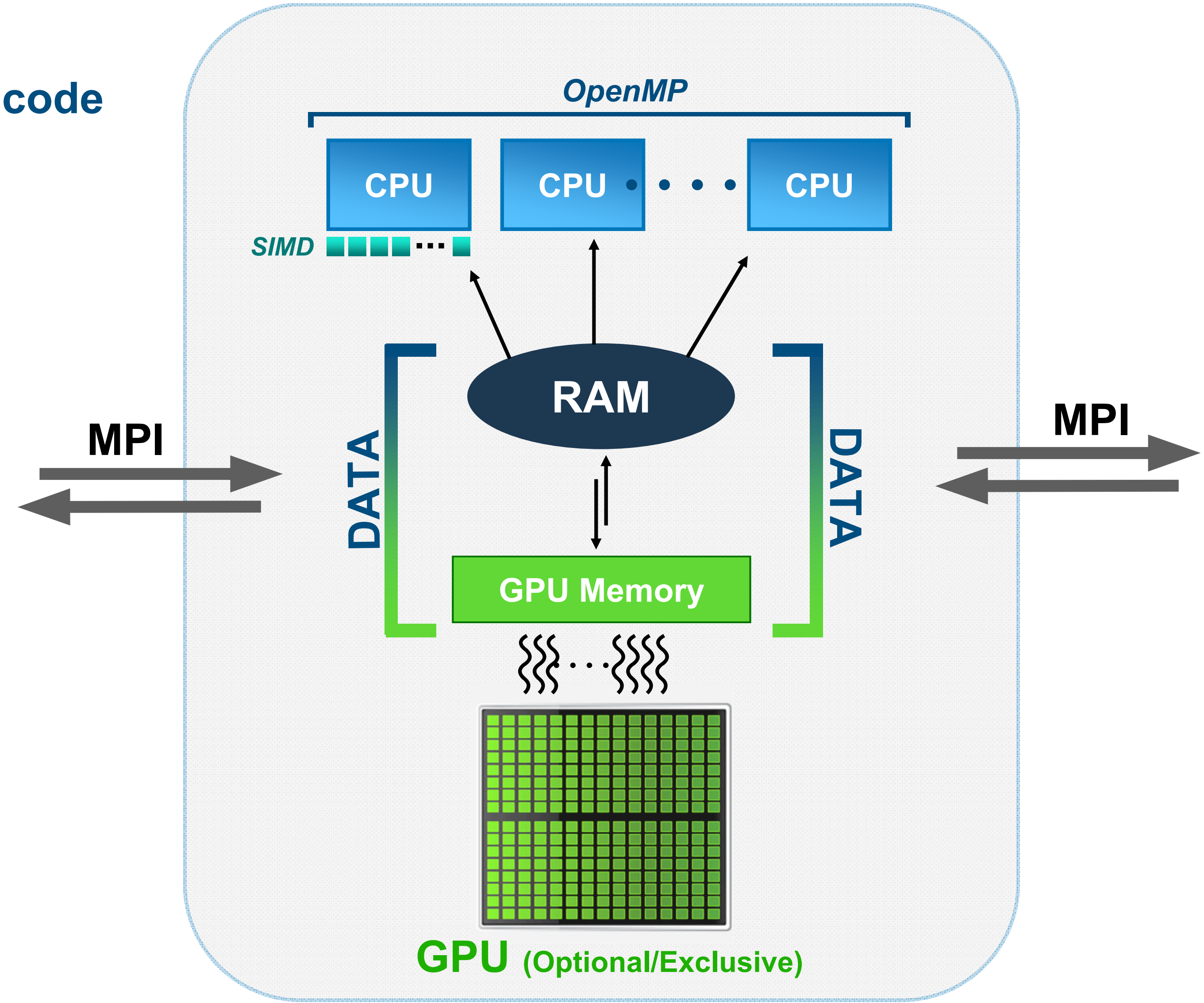
https://github.com/rahulbgu/PICTOR

Electromagnetic code

- *> Second order FDTD*
- *> Boris Pusher*
- *> Yee Grid*



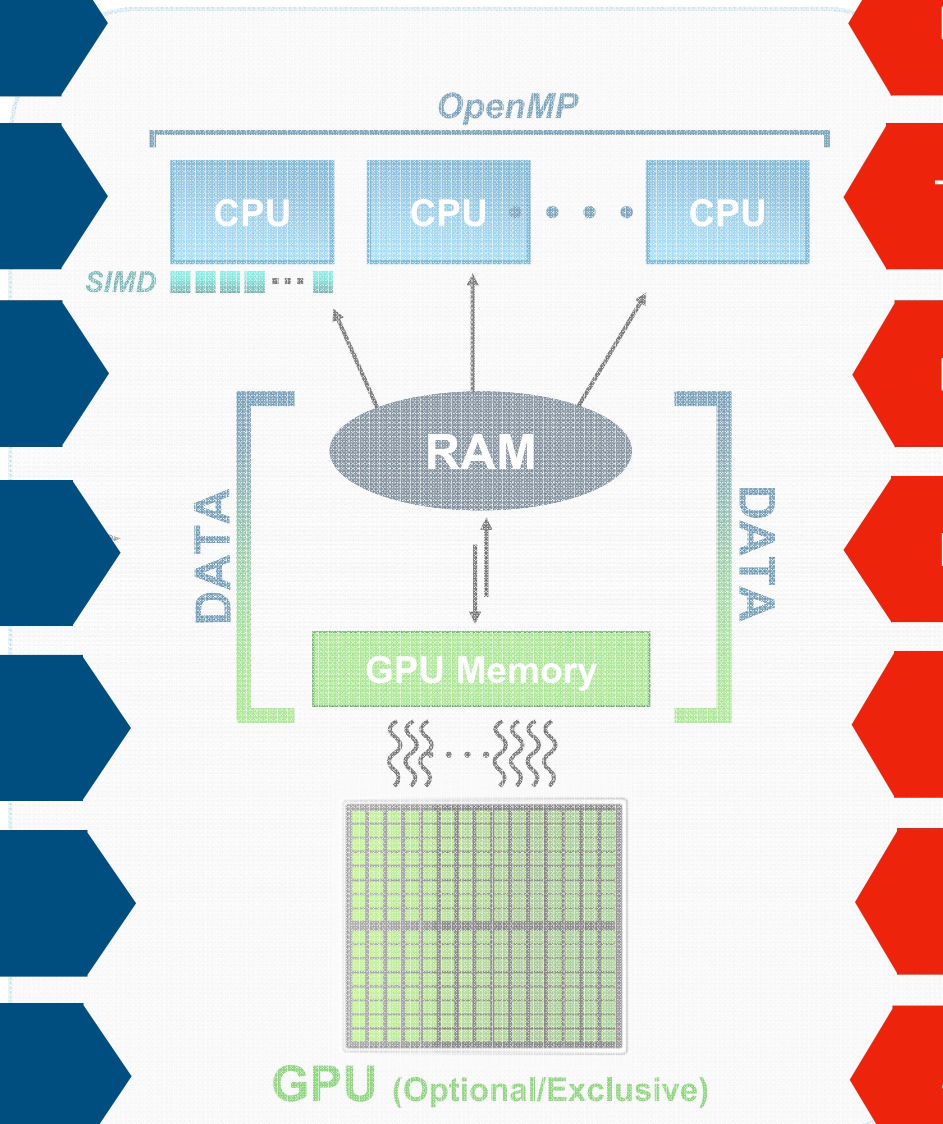
- *> Charge conserving current deposition*



PICTOR

Particle-in-Cell Code

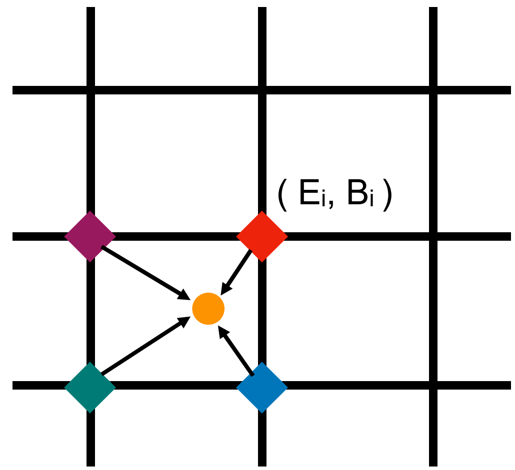
- Precision Control
- Dynamic Load Balancing
- Machine Level Optimization
- Cache Usage Optimization
- Memory Usage Optimization
- OpenMP+SIMD+GPU
- MPI + Parallel HDF5



- Multiple Species
- Test Particles
- Particle Tracking
- Moving/Fixed Boundaries
- Comprehensive Output
- Simple/Flexible Setup
- Shock, Turbulence, ..

Adopting to SIMD

Interpolation

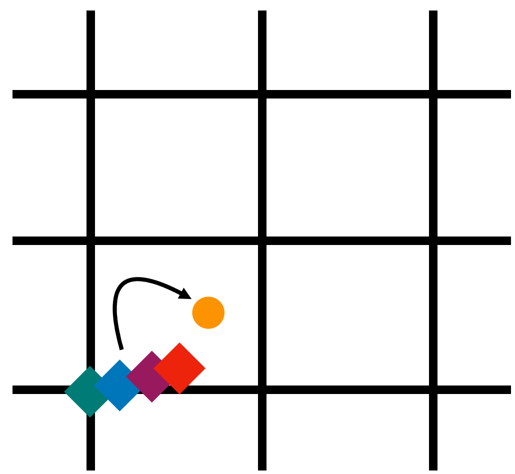


for

$\mathbf{prtl\_E_x} \;\mathbf{+=}\; wt * E_x$

SIMD

Reshape



~~***for***~~

$\mathbf{prtl\_E_x} \;\mathbf{=}\; \mathbf{\_sum\_(}\; wt \;.\!*\; E_x \;\mathbf{)}$

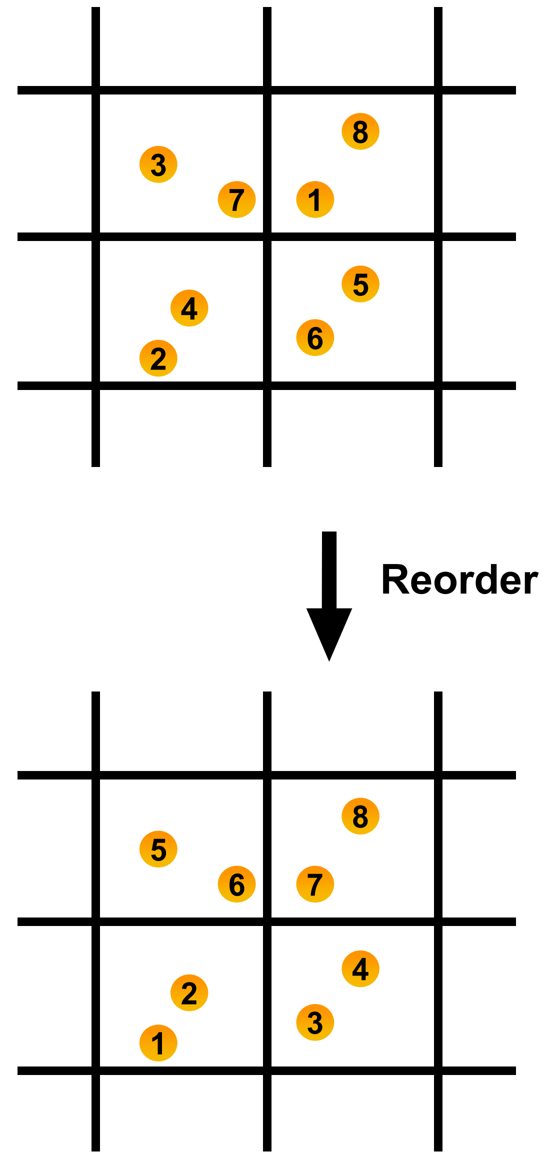
Improving Memory Access

Sorting particle data according to their locations significantly improves cache hit

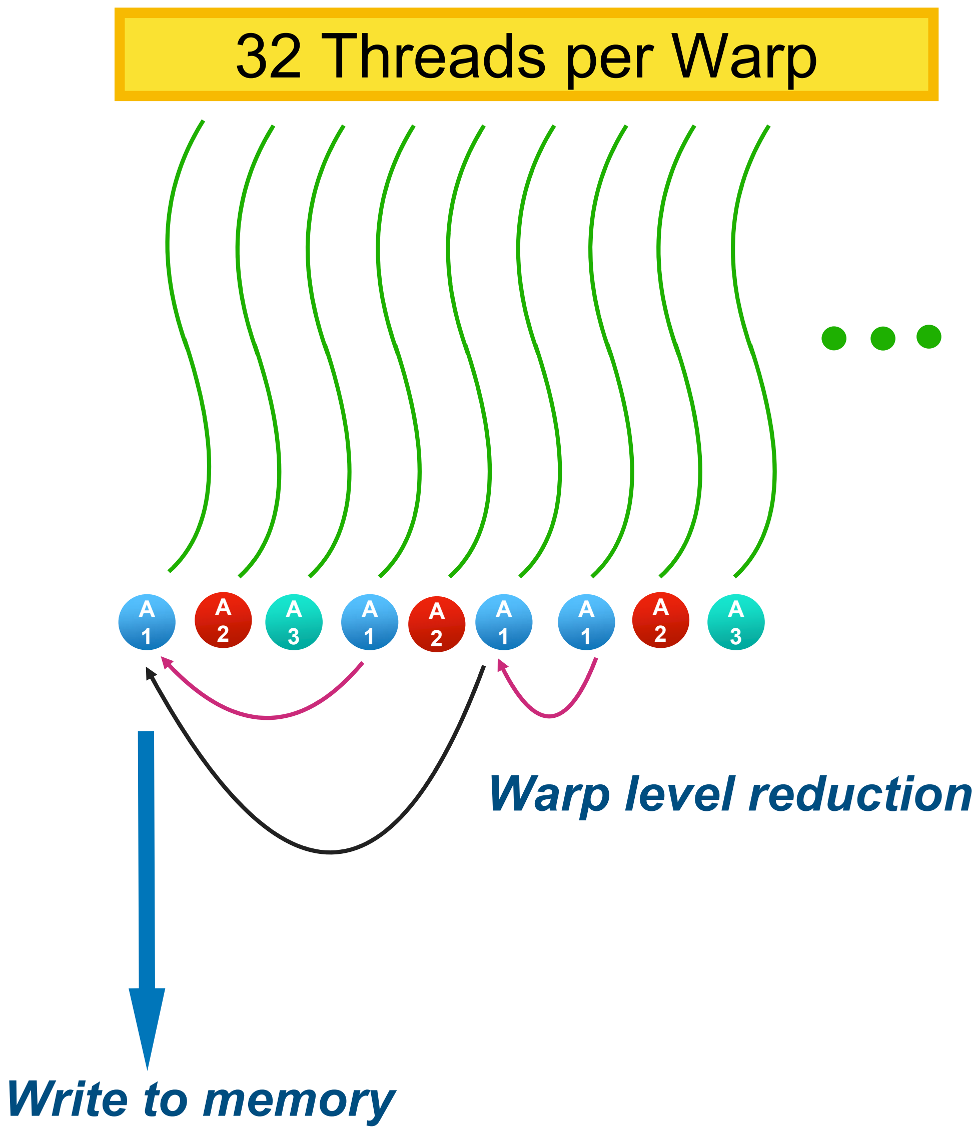
A simple *Bucket sorting* improves performance significantly both on CPU and GPU

Sorting keeps the particle data compact

Strip-mining of particles data (with CPU threads)



Low-level Optimization

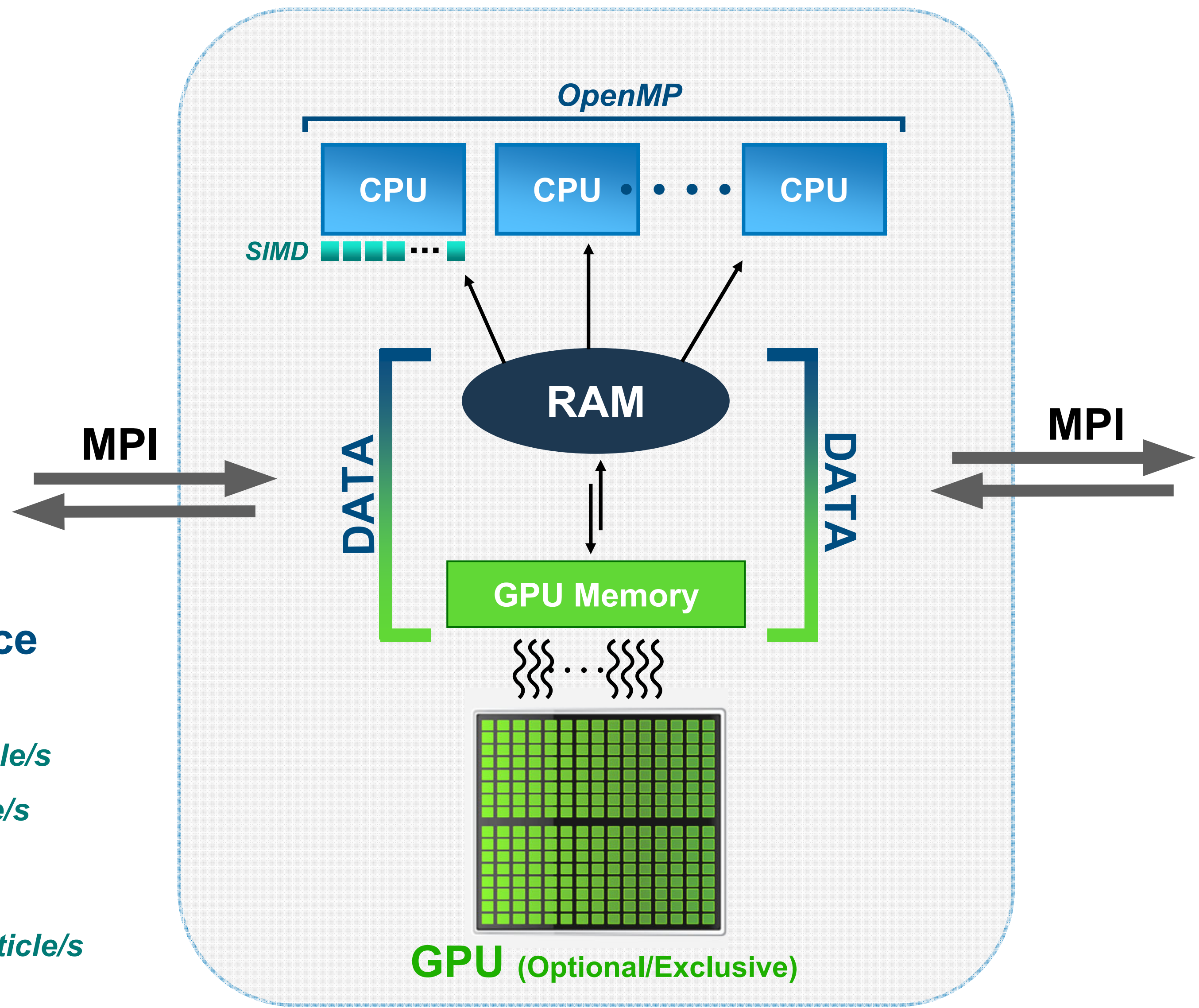


Reduces conflicts among concurrent threads/data traffic

PICTOR

Particle-in-Cell Code

https://github.com/rahulbgu/PICTOR



Performance

> CPU 2.4 GHz
Xeon
2D: ~20 M particle/s

3D: ~8 M particle/s

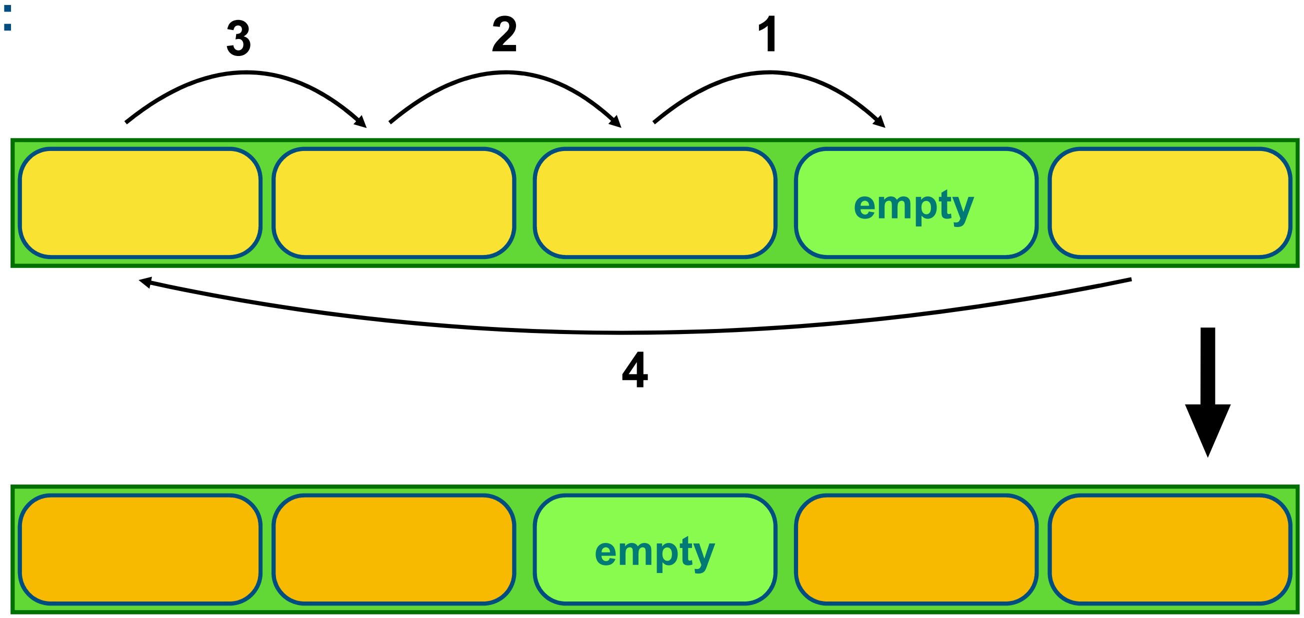
> GPU Tesla K20X

3D: ~40-50 M particle/s

Improving GPU memory usage

Strategy: divide available memory in chunks

Particle sorting:



Exascale Computing
the next frontier

Scalability

Reduce internode communications and global communications

Concurrency

Designing algorithms for very high-concurrency

Perfomance

Minimize data movement

Input/Output

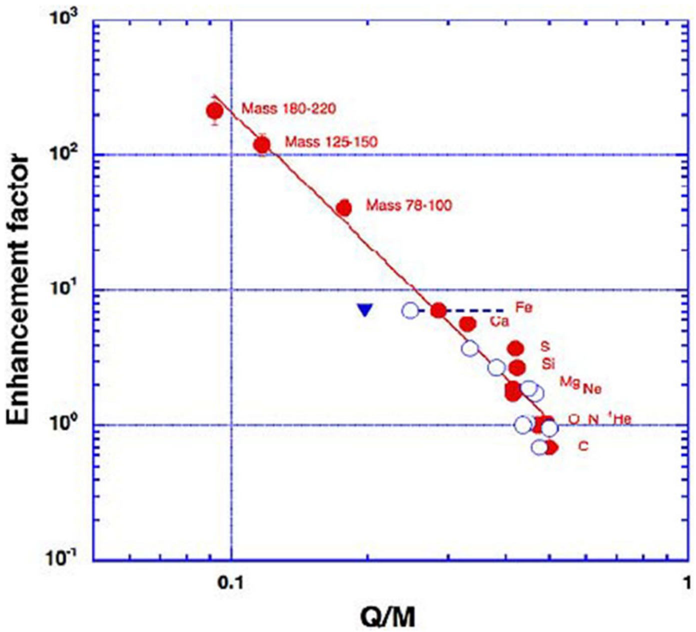
Asynchronous writing of the output file

Keeping codes simple !

Optimizing total (Computer+Human) time

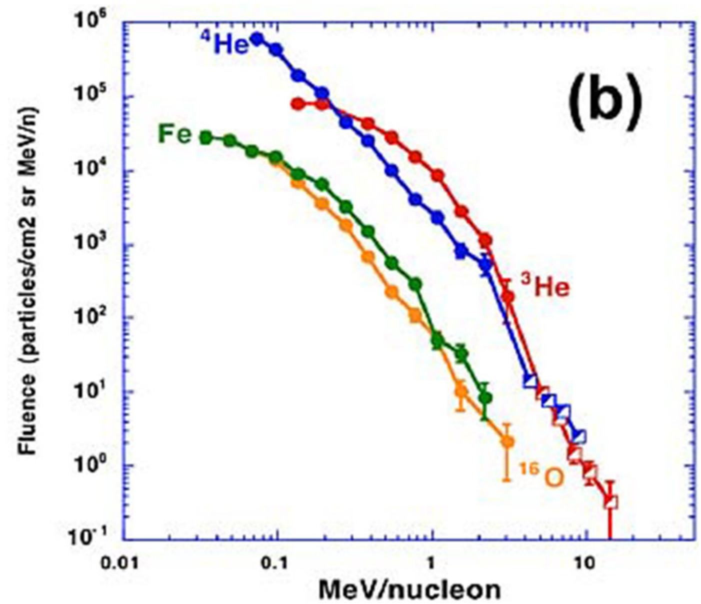
Prefential Heating and Acceleration of Heavy Ions

Impulsive Solar Flares



Mason et al., ApJ 2004

Energy Spectrum



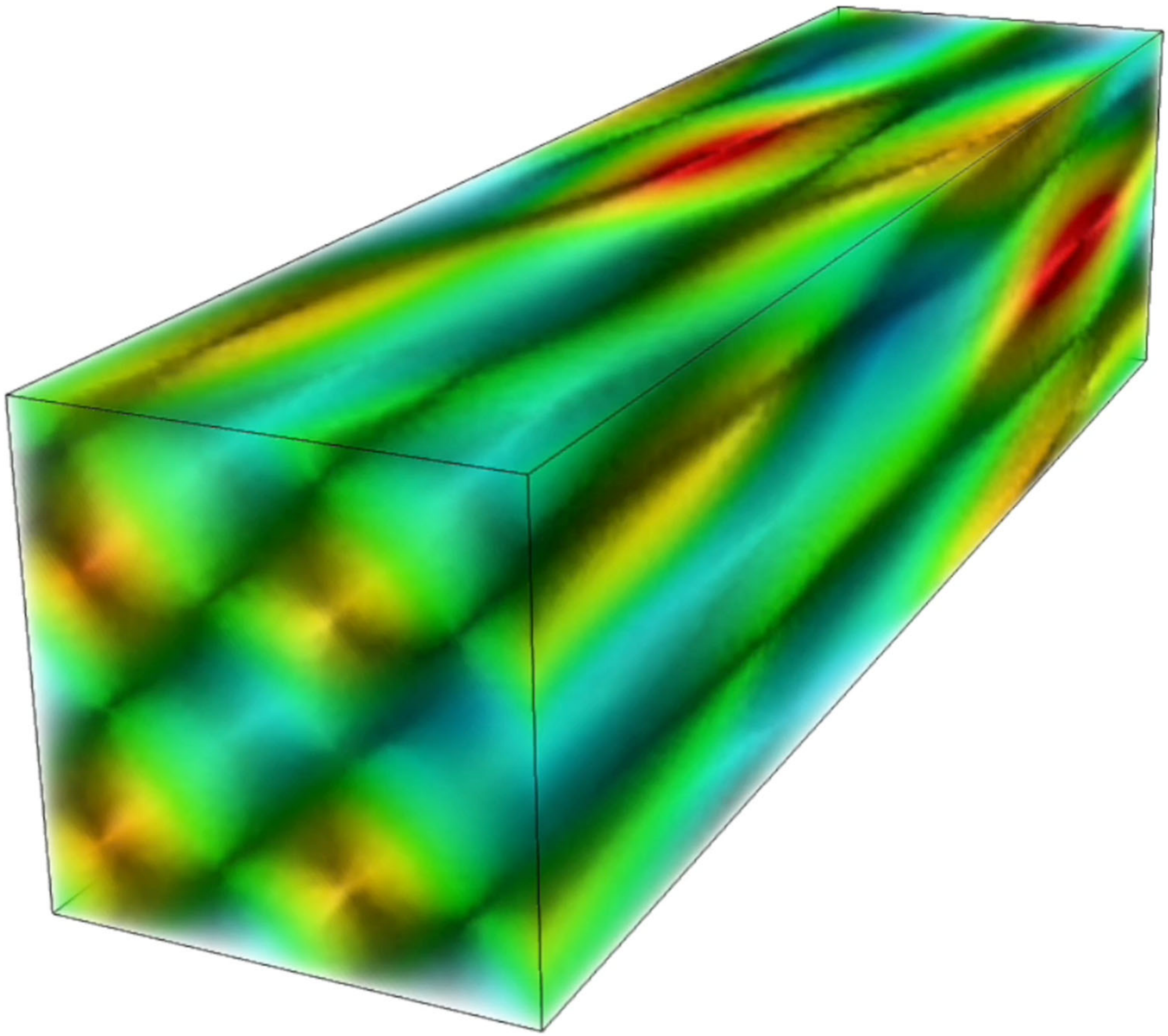
Mason et al., ApJ 2002

Is it due to Cyclotron damping of Alfvenic turbulence?

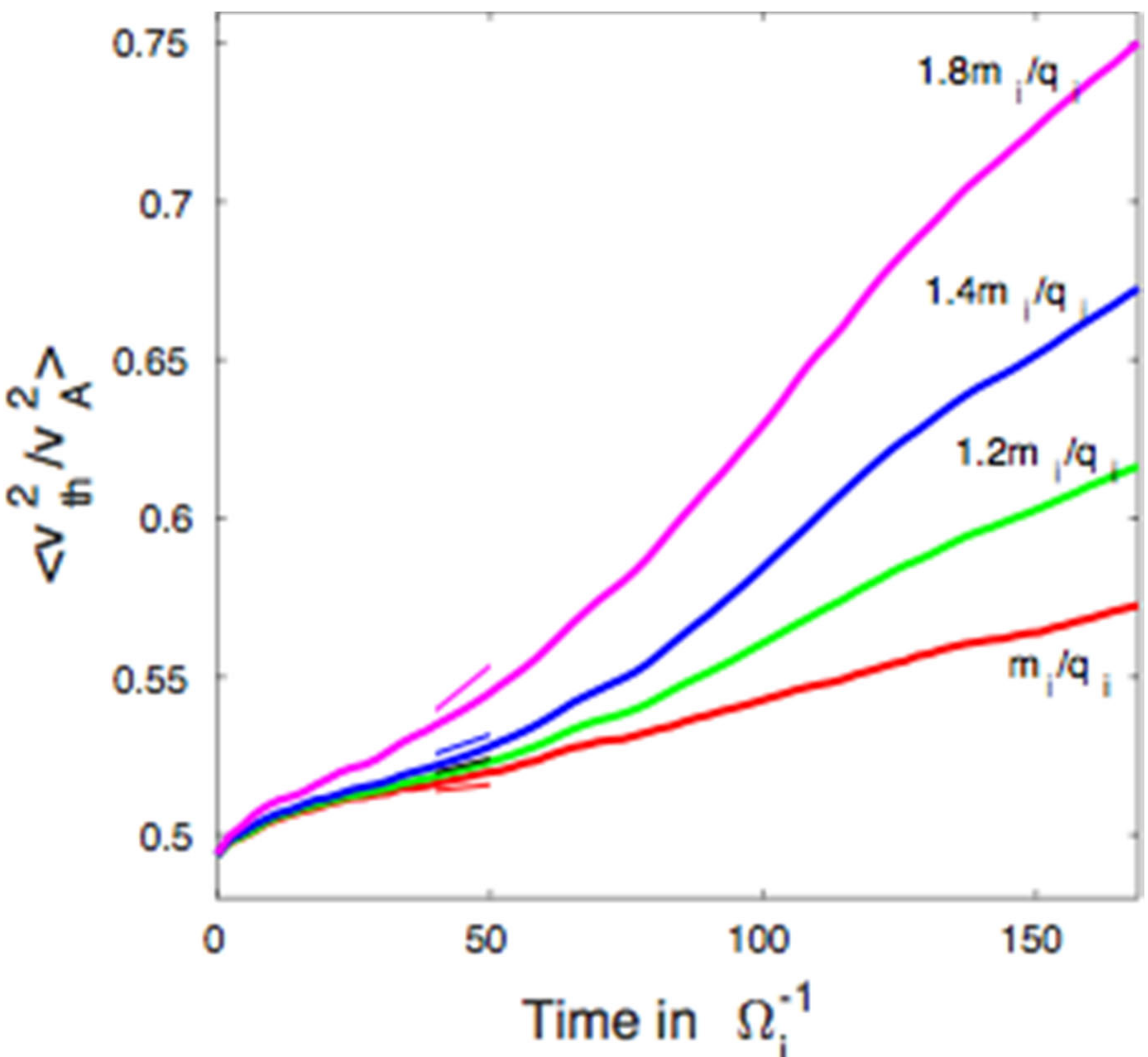
nulate the turbulence and see how different species behave in the turb

Anisotropic Turbulent Cascade

Parallel Velocity



Average Energy/Mass Vs. Time



$$\vec{v}_{th} = \vec{v} - \vec{E} \times B_0 \hat{x} / B_0^2$$

Resonance Condition

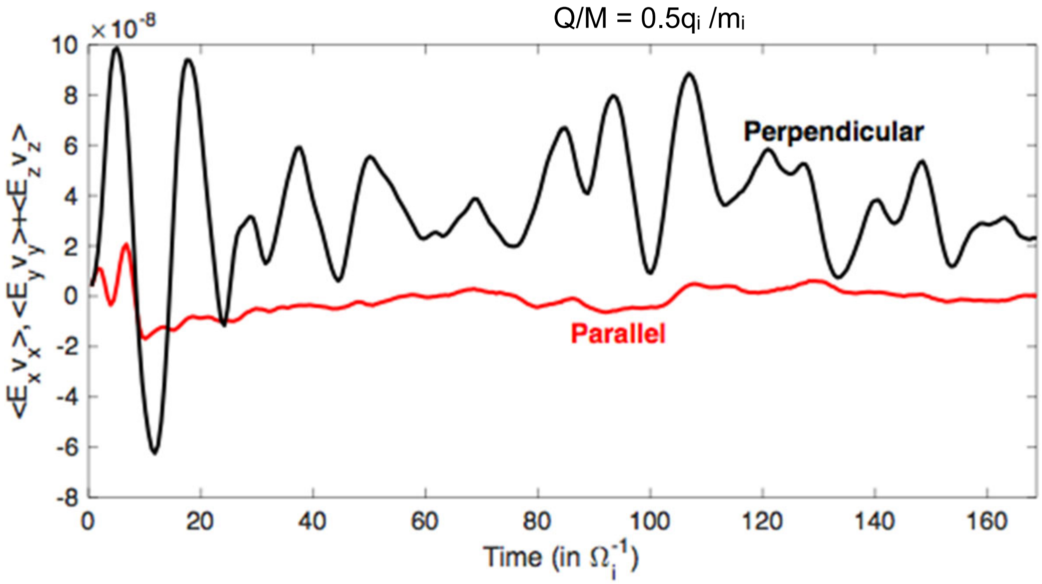
$$\omega \pm k_{\|} v_{\|} = \omega_c$$

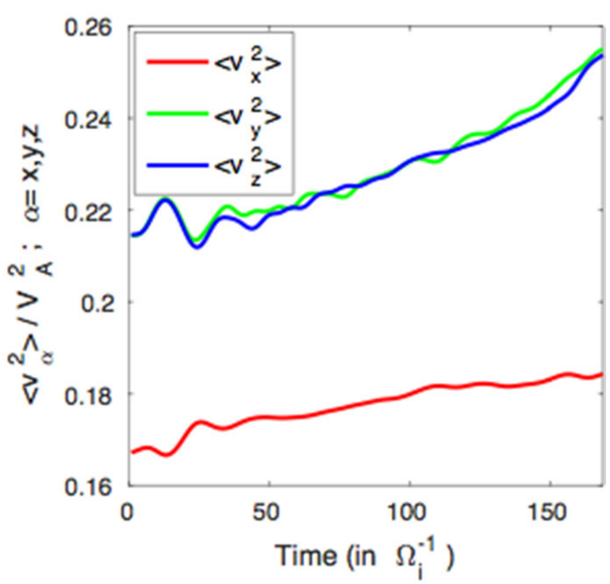
Heavier ion species (smaller Q/M) heat up at faster rate!

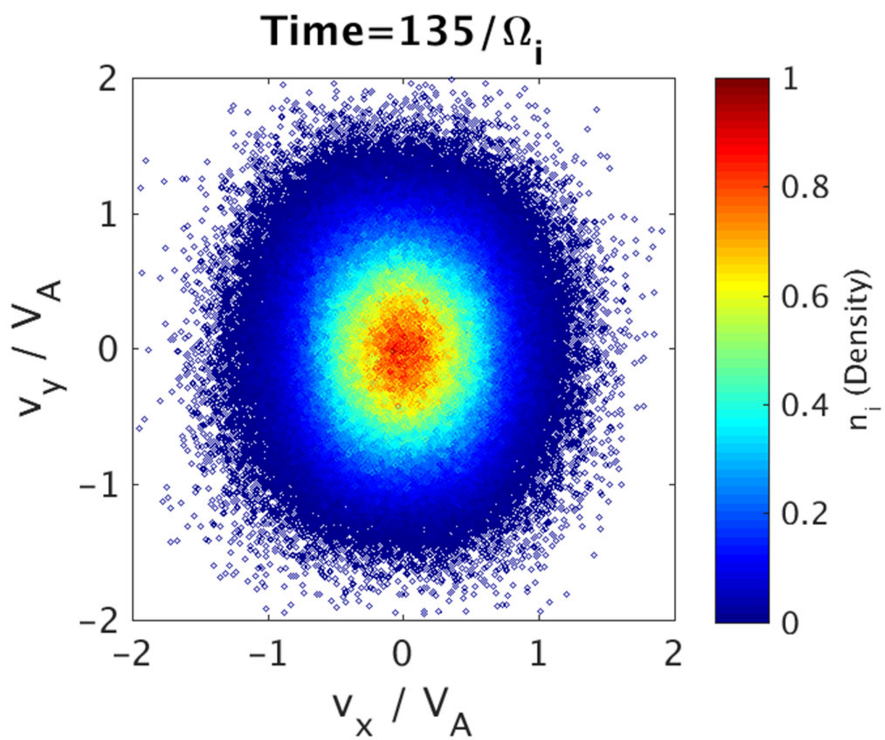
Perpendicular Heating

Heating is mostly due to the transverse electric field

Velocity space distribution is anisotropic, $T_\perp > T_\parallel$

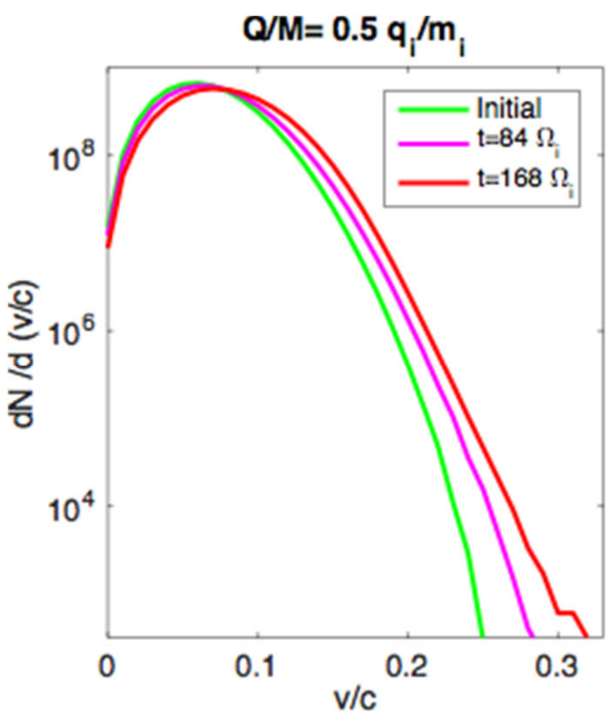


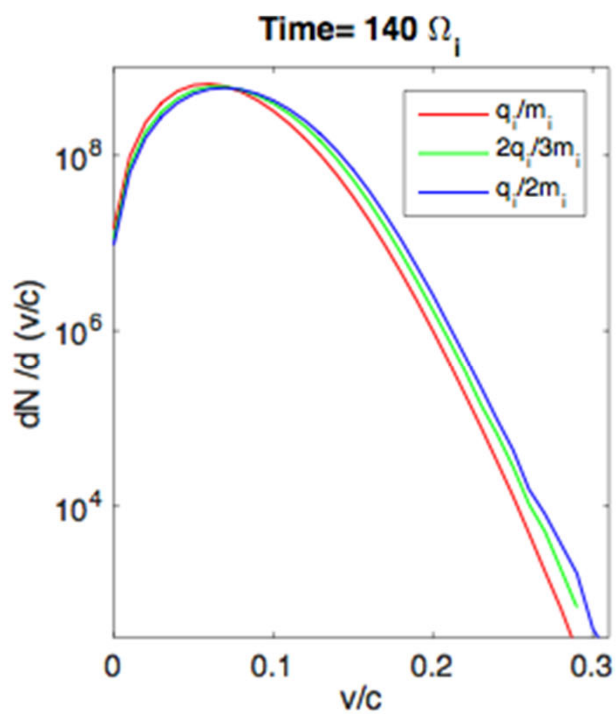




Acceleration to non-thermal energies

Some particles are accelerated to non thermal energies





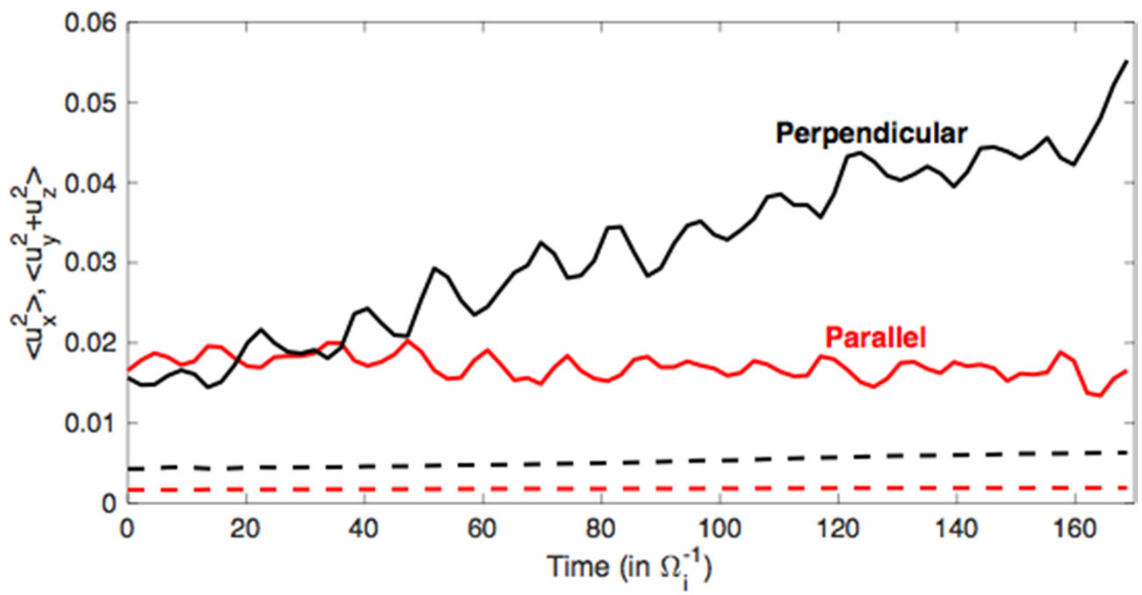
Heavier ions are accelerated to slightly higher energies

Acceleration to non-thermal energies

What is so special about the accelerated particles?

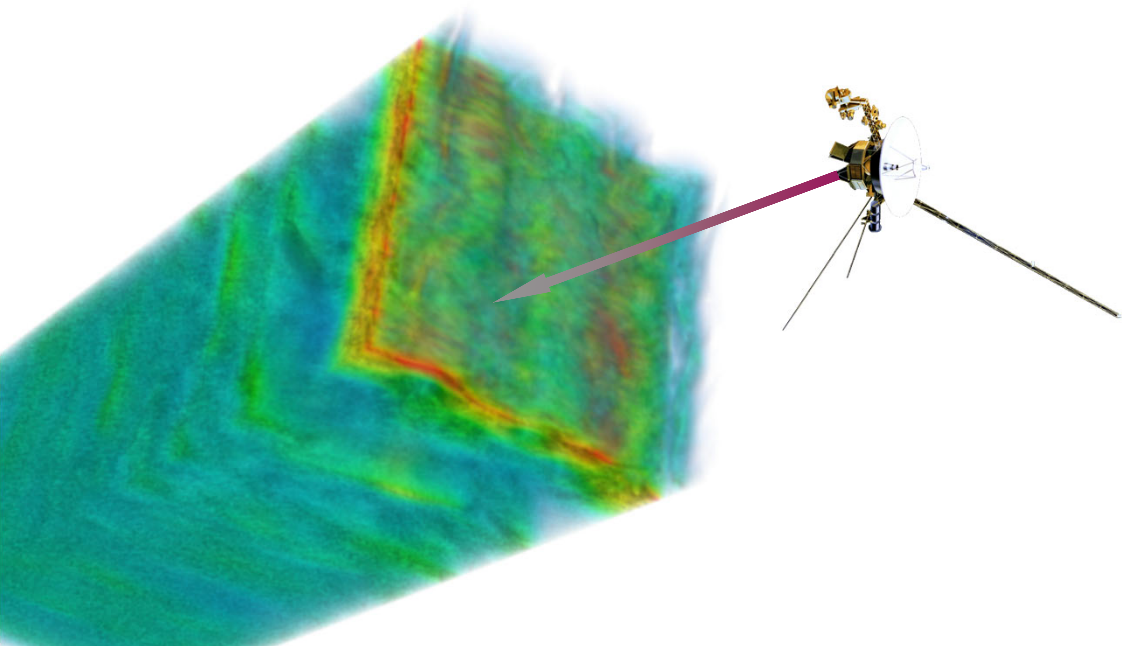
Resonance Condition : $\omega \pm k_{\parallel} v_{\parallel} = \omega_c$

Particles that end up in the non-thermal tail vs. all

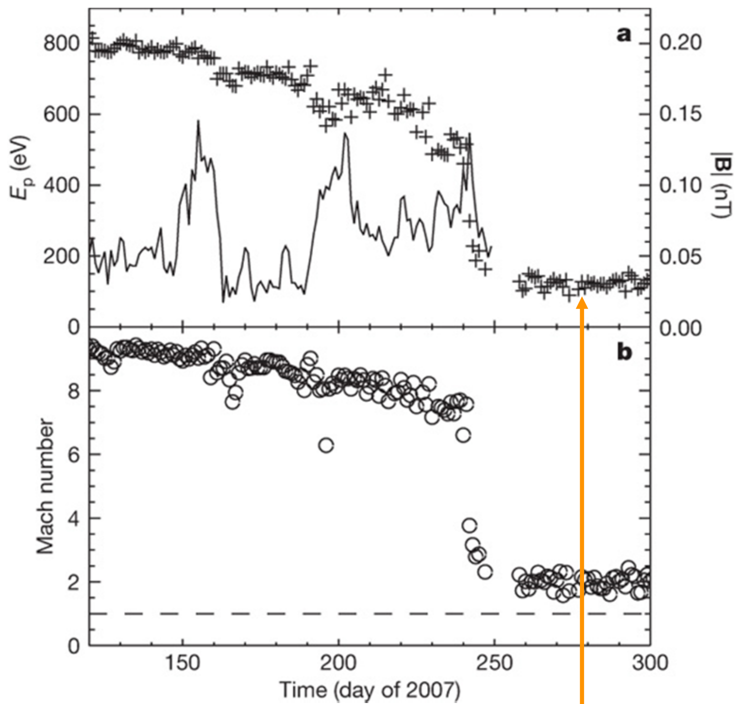


Accelerated particles had relatively large *initial speed* to begin with!

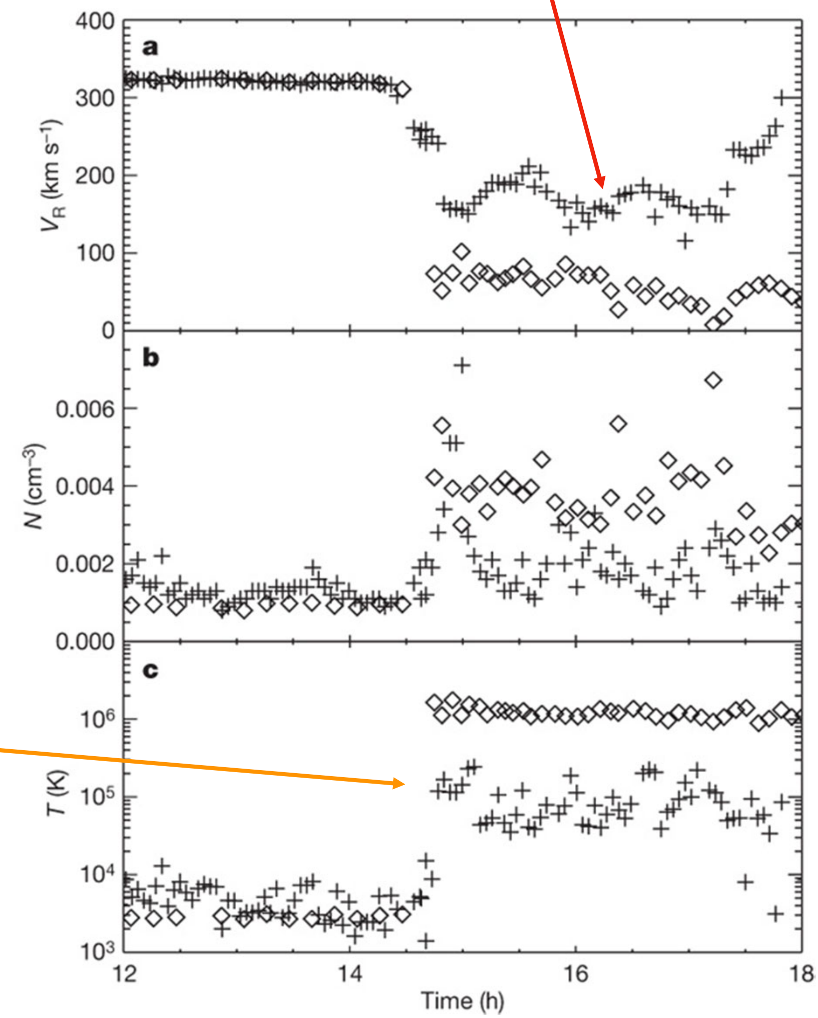
Dynamics of The Solar Wind Termination Shock



An Unusual Shock



The downstream was moving faster than expected (low compression)

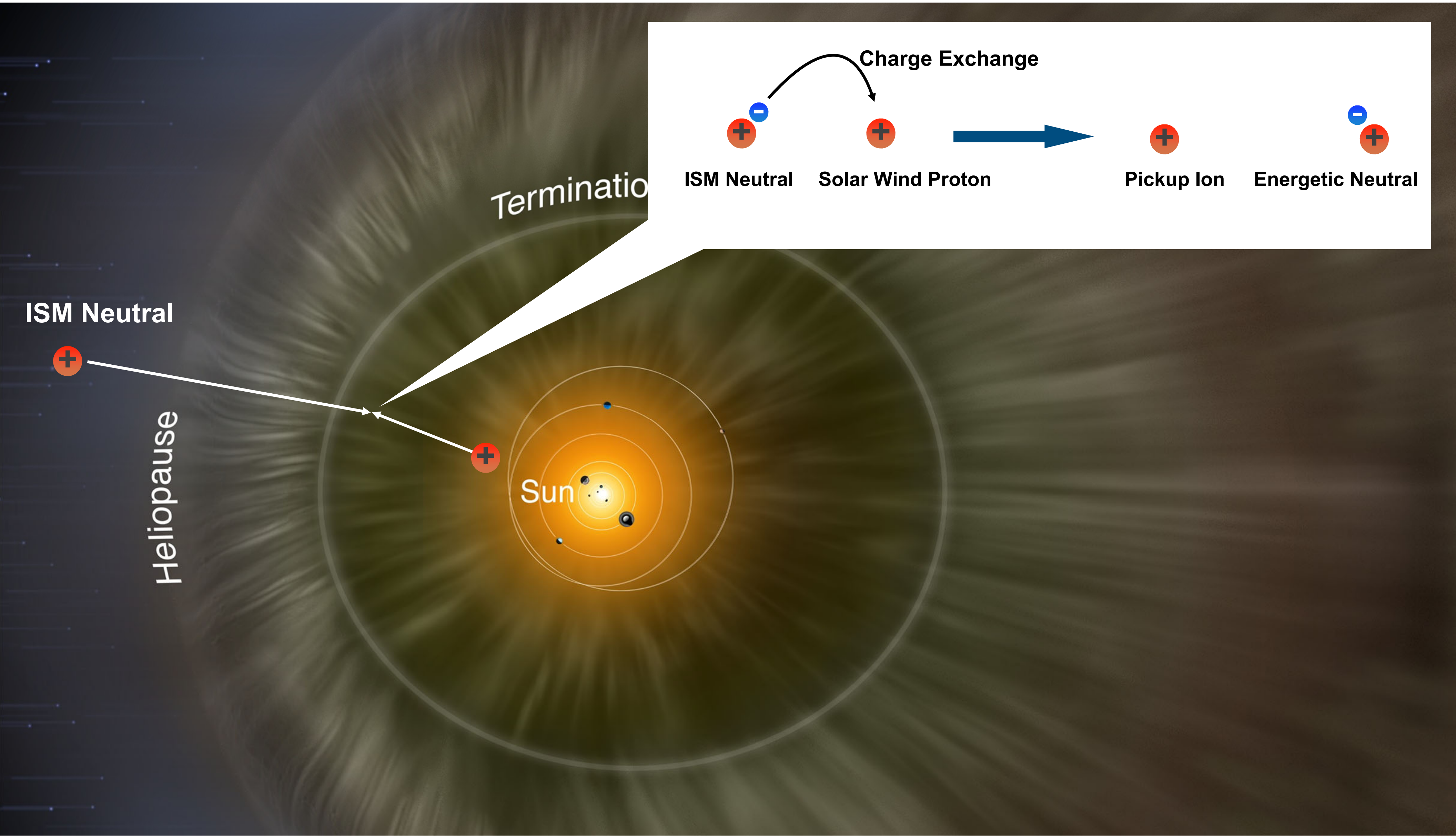


Most of the dissipated energy did **not** go into heating SWIs

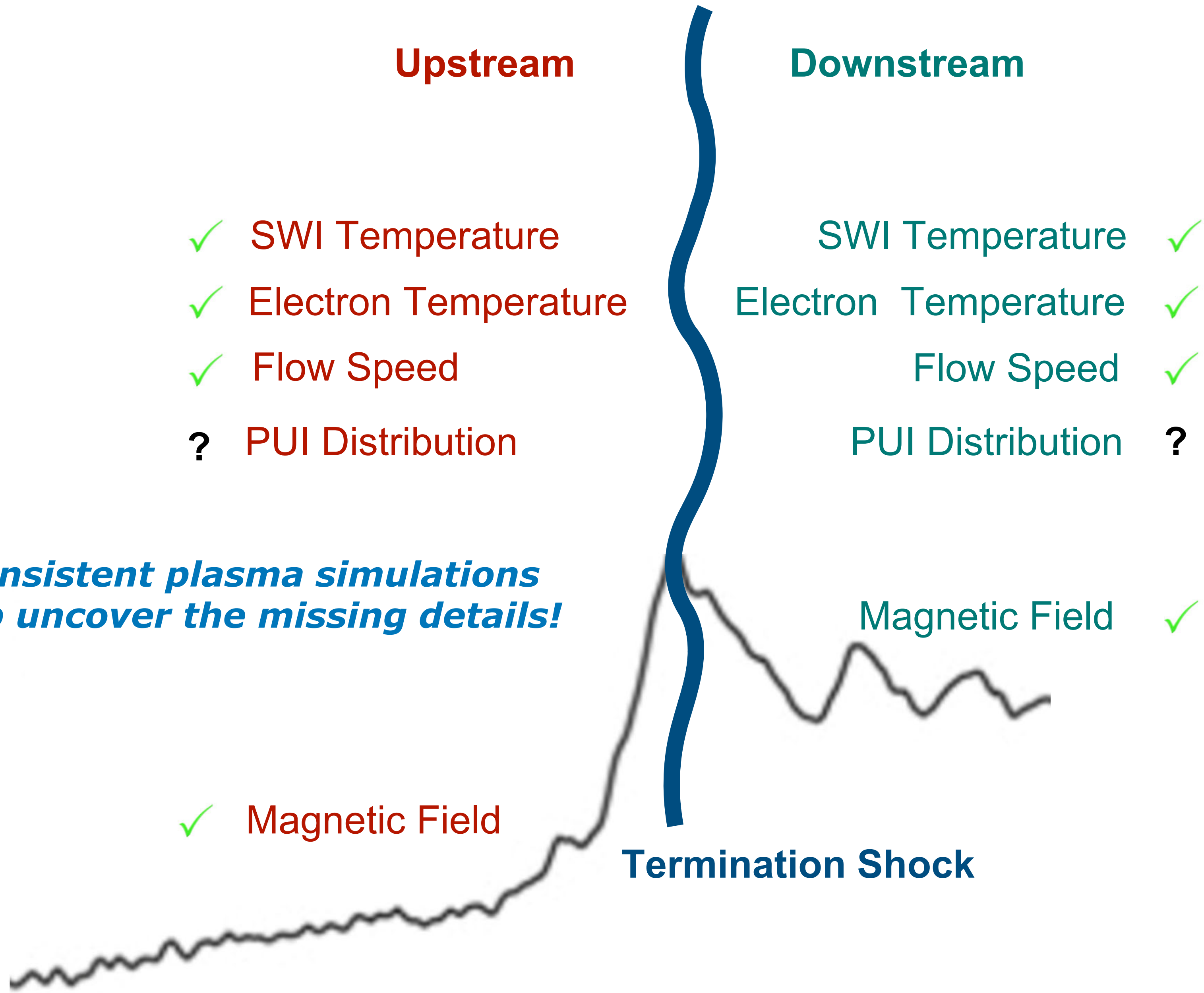
JD Richardson *et al*. *Nature* **454**, 63-66 (2008)

Why So Different?

Ans: Pickup Ions

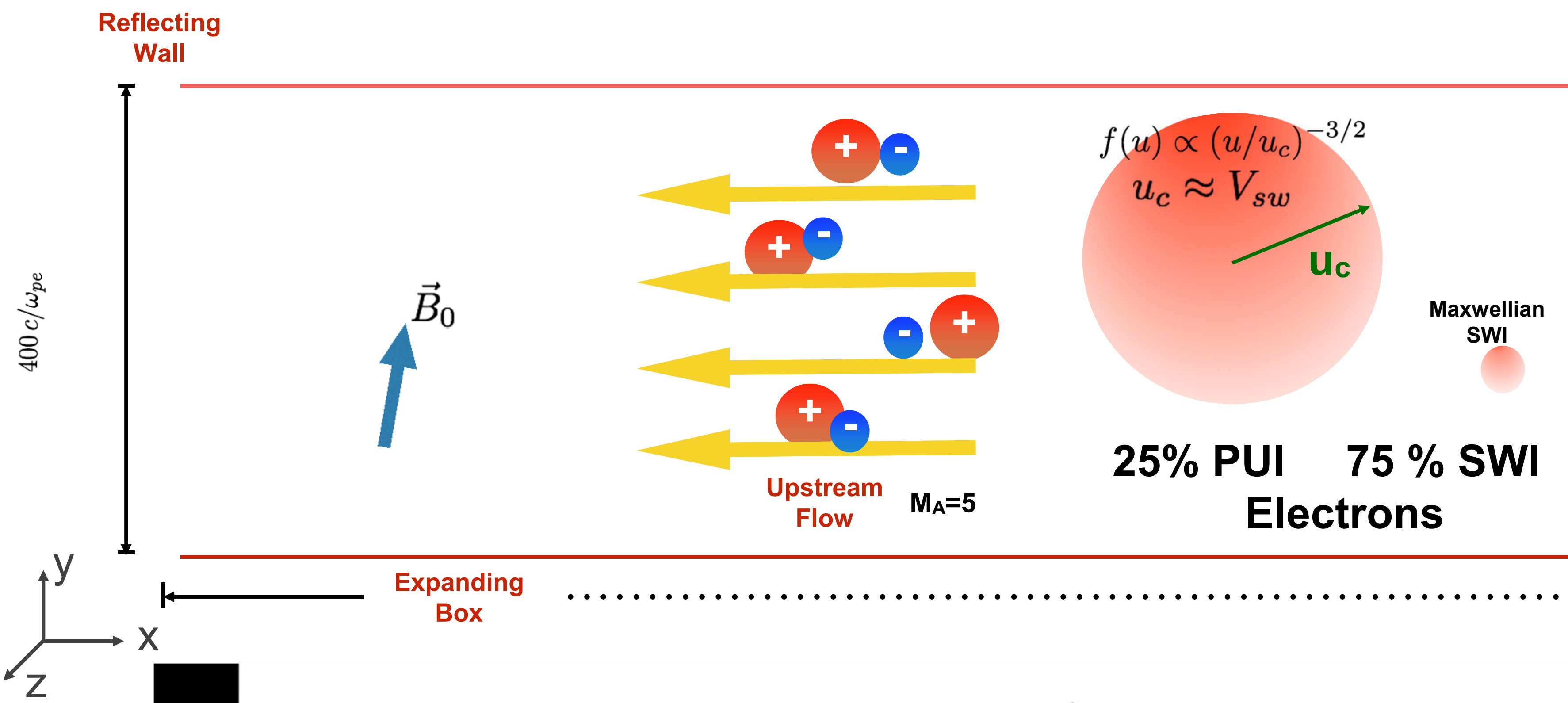


No Pickup Ion Data from Voyager 2



Self-consistent plasma simulations can help uncover the missing details!

Shock Simulation

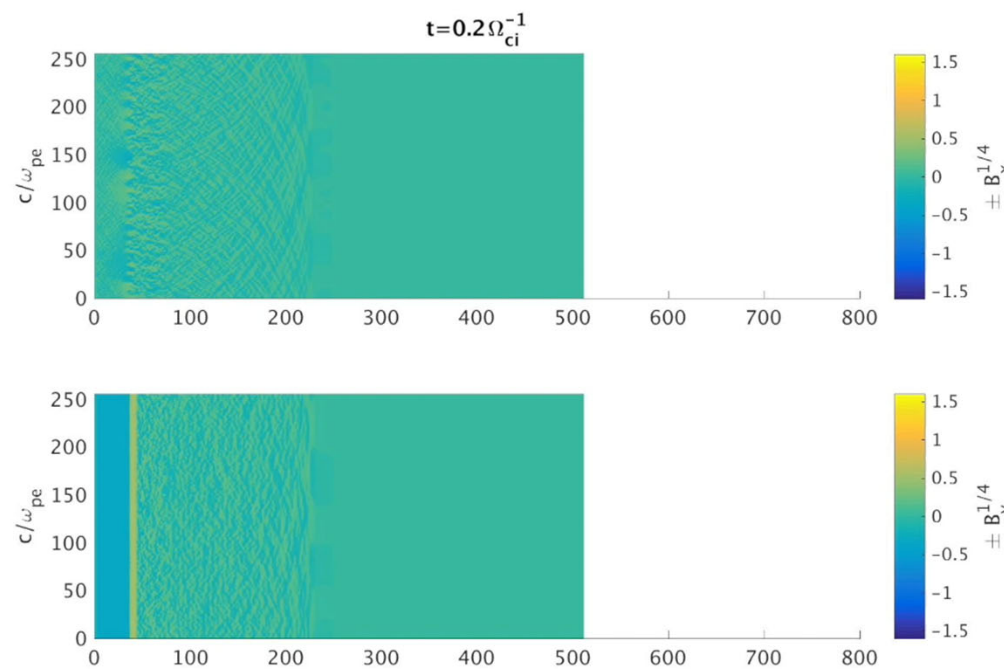


Numerical Parameters

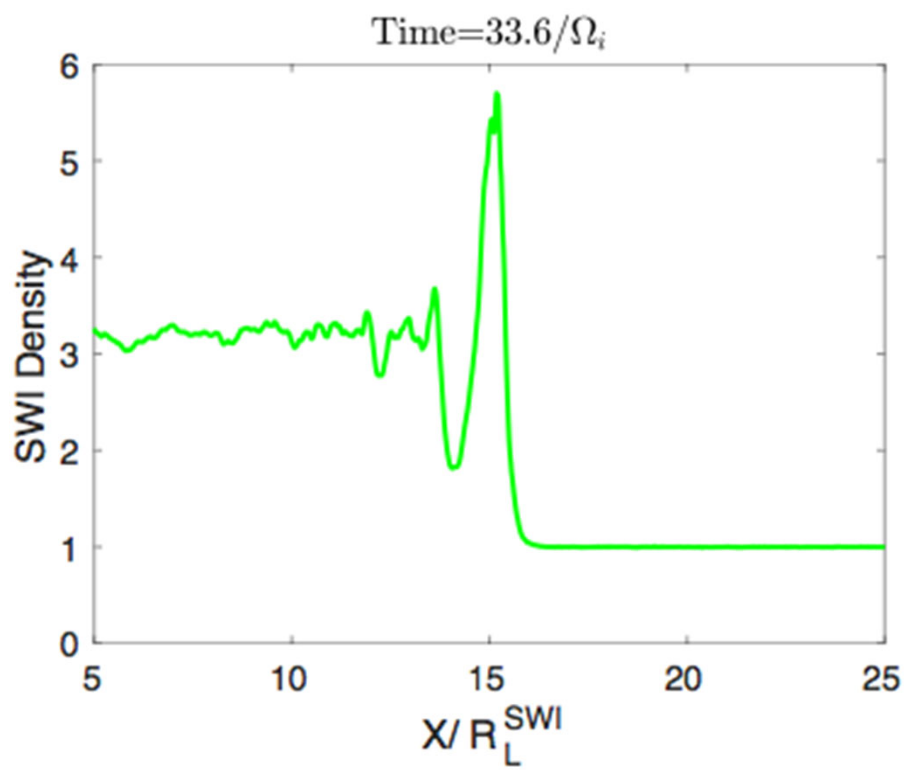
M(+) / M(-) = 100

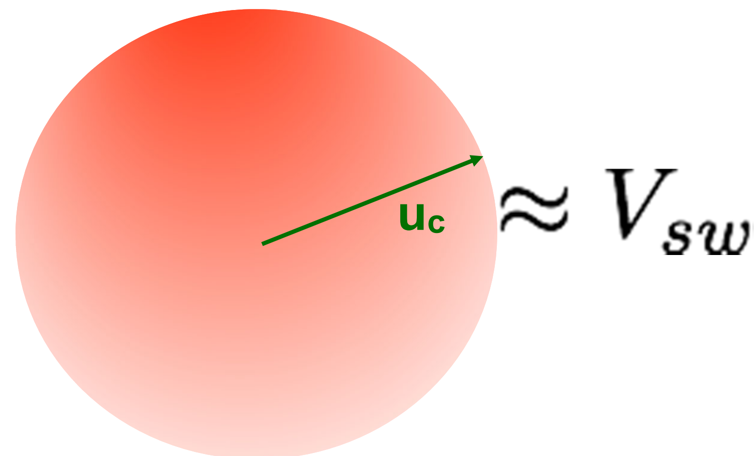
32 electrons/cell

V_A/c = 0.01



Simulations Vs. Observations





Filled-Shell PUI distribution function

Shock compression ratio is 3.2 (Voyager 2 data suggests 2.5)

SWIs are also hotter than observed

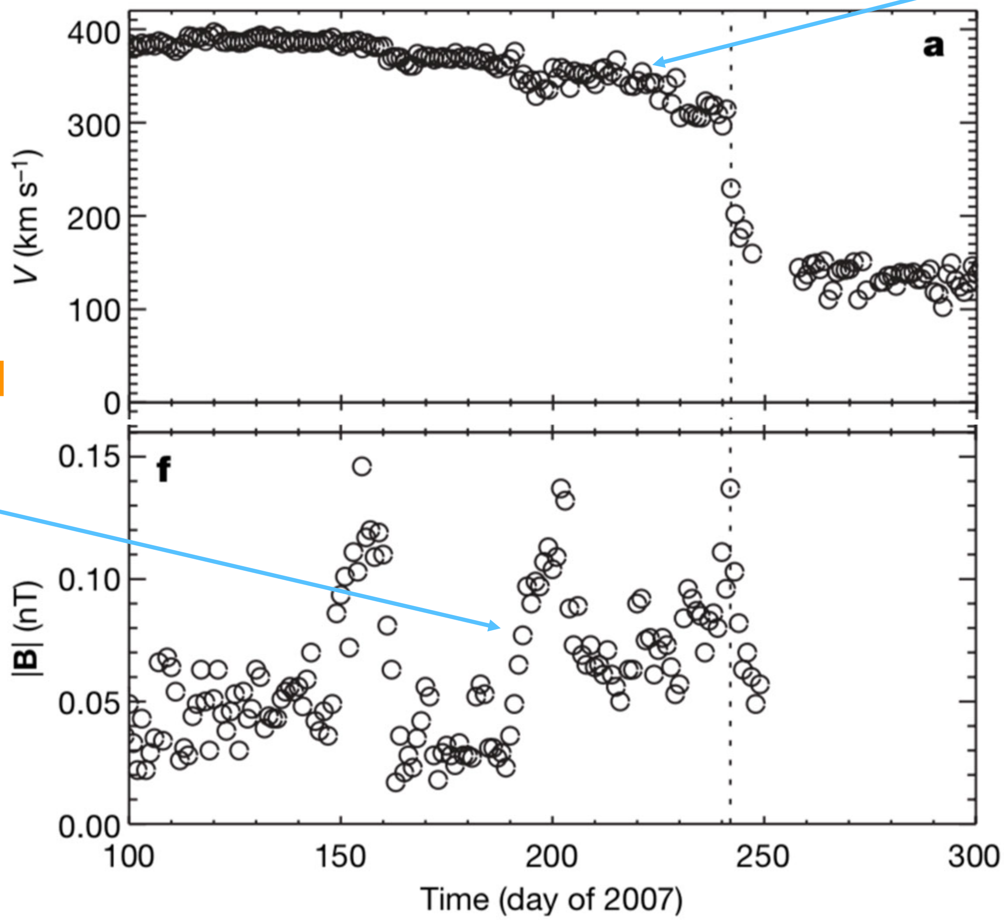
A clear disagreement between numerical simulations and Voyager 2 observations !

A Closer Look At The Data

The wind slowed down

Most pickup ions were produced when the Solar wind was moving faster !

Magnetic field strengthened

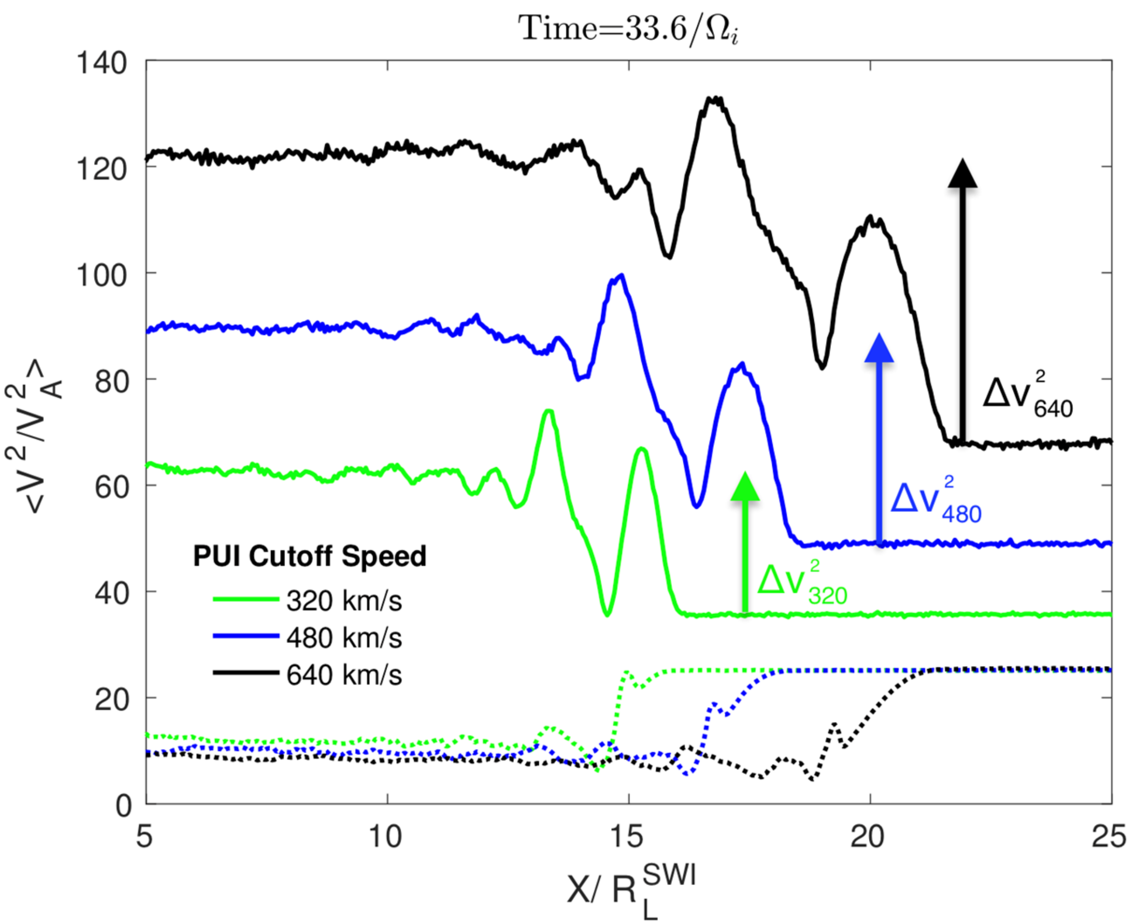


Upstream plasma is compressed and heated !

JD Richardson *et al*. *Nature* **454**, 63-66 (2008)

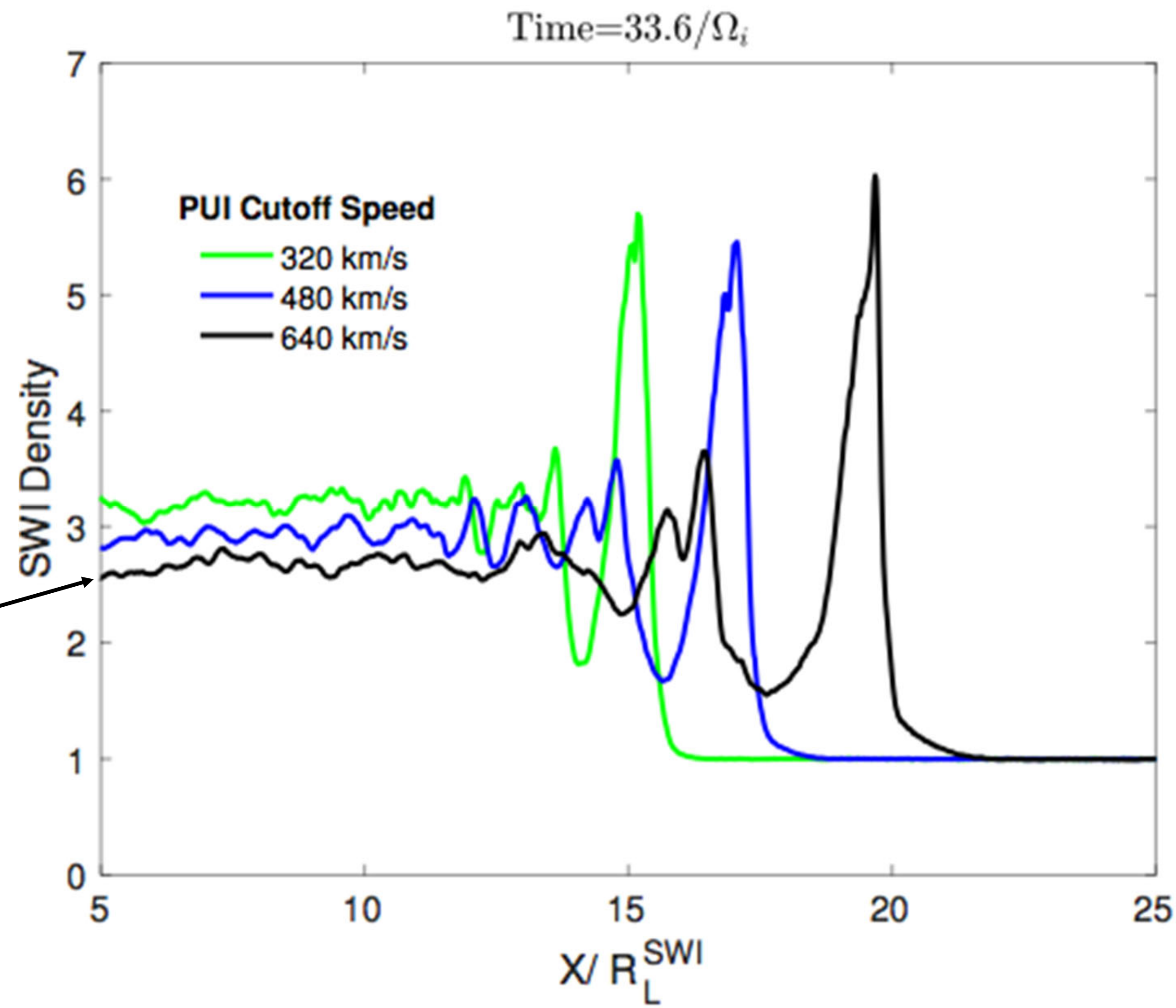
Pickup ions are expected to become more energetic

Energy Partition Vs. Cutoff Speed



The gain in energy for PUIs at the the shock is larger if they were already more energetic upstream of the shock

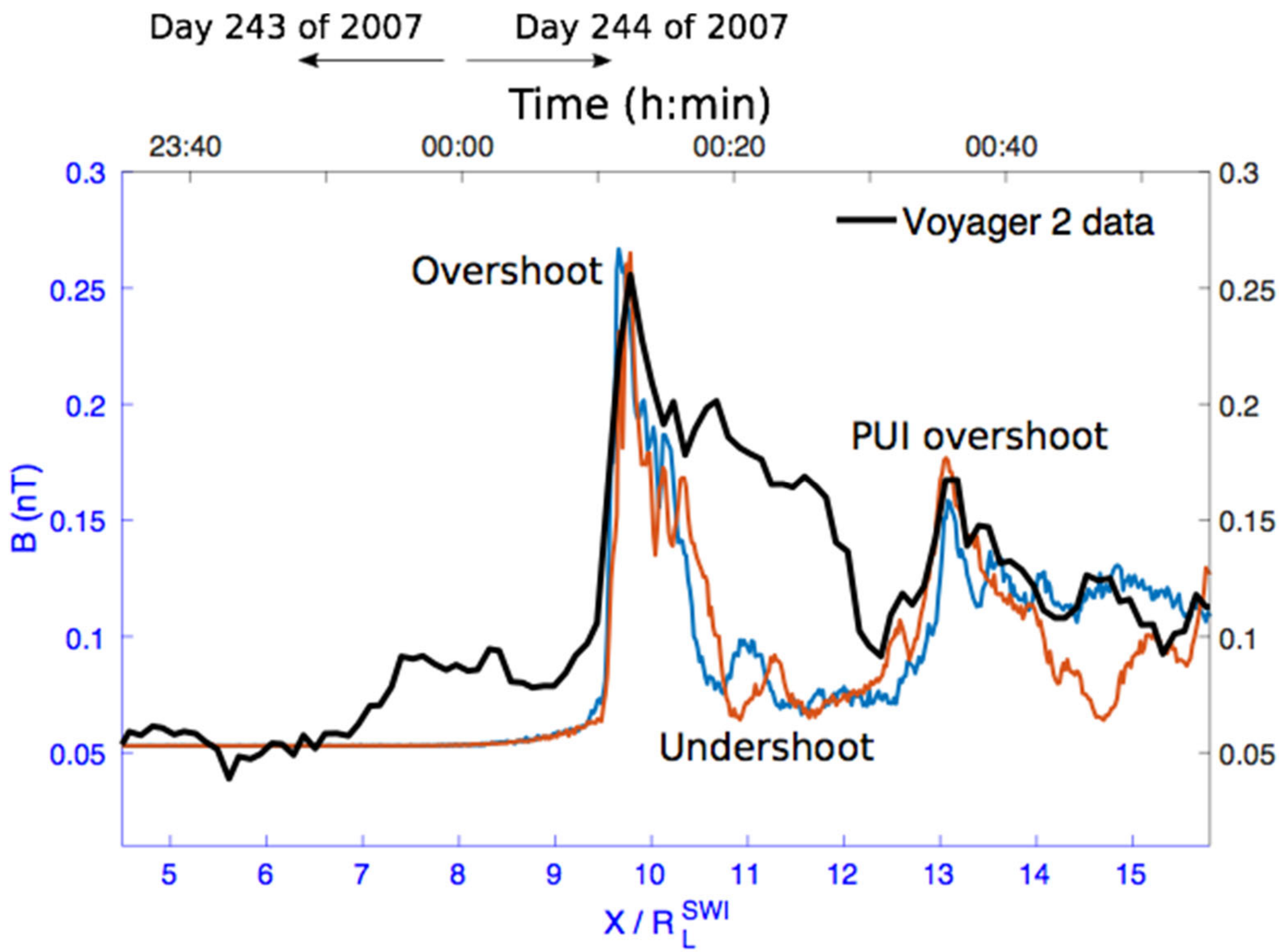
Density Compression



The observed compression was about 2.5

More energetic upstream PUIs lead to lower density compression

Magnetic Field Structure



Understanding the Energy Partition