The Potential of Spaceborne High Performance Computers for Onboard Space Physics Data Analysis

Fall 1999, AGU Poster SM12B-06.

Michael L. Rilee[1],

S. Curtis, B. Farrell, A. Figueroa-Viñas,

J. Houser, M. Kaiser, M. Reiner[1]

NASA Goddard Space Flight Center

[1]*Raytheon ITSS*

Contact: 301-286-4743, Michael.L.Rilee@gsfc.nasa.gov.

Presented at the

1999 Fall Meeting of the AGU, Dec. 13-17, San Francisco, CA.

Moscone Center, Hall D

1999-12-13

Abstract

Future Space Science missions will involve multiple spacecraft seeking to sense phenomena that occur across a wide range of spatial and temporal scales. Some missions will be required to provide real-time feedback for space weather studies or monitoring. The operational and communication needs of constellations of spacecraft with highly capable sensors are immense and for some missions are currently beyond our technical ability.

To examine the problem of science data flow at its source, the Solar-Terrestrial (ST) Probe Science Application Team of NASA's Remote Exploration and Experimentation (REE) project has developed two data analysis applications for a spaceworthy scalable parallel computer being developed by NASA's High Performance Computing and Communication Program. To investigate the possible benefits of spaceborne high performance computing for Solar-Terrestrial missions, the team chose to model data analyses that arise in radio frequency interferometry and plasma particle spectrometry. Radio frequency interferometry was chosen as a fundamental technique for imaging and plasma wave spectrometry. Plasma particle spectrometry was chosen because of its central importance in ST physics. Both kinds of science instruments produce large amounts of high-dimensional data that places severe demands on mission communications infrastructure. The science communication burden may be eased if a way could be found to extract scientifically meaningful information from the raw data in place onboard the spacecraft. Our work shows that the REE computer architecture is well suited to the data analysis problems we have studied.

We present results of our modelling efforts and discuss how well our data analyses perform on the *REE Flight Processor* testbed. In particular, new results from our plasma spectrometry model will be presented: a highly parallel, efficient, robust, and flexible approach to reducing plasma spectrometer data will be presented.

Outline of this Presentation

This presentation is organized into six vertical display tracks:

1. Introductory track (this track),
2. REE/Plasma Moment Application (PMA) basics
3. Calculation: from data to moments,
4. Data and data structure modifications
5. Program flow, results, and conclusion.

Overview of the REE Project

The REE Project, administered by JPL, is a main thrust of NASA's High Performance Computing and Communication Program. Its goal is to develop an on board supercomputing capability.

Performance Goals for the REE Processor:

- REE processor 1999, $>$ 720 MOPS, 24 Watts: $>$ 30 MOPS/W;
- REE processor 2002, $>$ 6 GOPS, 20 Watts: $>$ 300 MOPS/W.

To achieve these goals the REE uses:

- COTS processors; Scalable networking architecture;
- COTS software environment (e.g. Beowulf, MPI, POSIX);
- Various levels of fault tolerant operation;
- Lightweight "middleware" layer for Fault Detection, Mitigation.

REE/Plasma Moment App. Science

The REE/PMA models the dataflow and analysis that couples particle spectrometer data to mesoscale space physics. The most important ‘patrol’ measurements are:

- Fluid moments

  Density, bulk flow, pressure tensor, heat flux;

- Basis function expansions

  Spherical harmonics, physically based trial functions.

Operational goals for PMA

Our vision is that a particle spectrometer making use of REE technology would be able to:

1. Reduce data to moments in real time,
2. React to information thus extracted,
3. Enable full fidelity data to be selected only when necessary, and
4. Dramatically reduce science data downlink volume.

If science data communication requirements are sufficiently reduced, then enginering and control communication requirements become resource and cost drivers.

Baseline instrument model

The baseline instrument model is roughly motivated by the electron spectrometer flown on the ISEE-1 spacecraft (Ogilvie et al. 1978).

- Six electron detectors, three orthogonal axes, are
- Rapidly stepped through energy.
- As the spacecraft rotates, most directions are sampled.

After enough data has been collected, one can start to fashion a picture of the three (velocity) dimensional plasma distribution function (PDF).

Important considerations concerning the PMA instrument model

The PMA instrument model is necessarily a high level model.

- It neglects calibration issues, spacecraft potential effects,
- represents simply the detector efficiency and geometry, and
- currently samples a model, essentially Gaussian, plasma distribution with Poissonian countrate fluctuations.

But it can also represent arbitrary sensor geometries and plasma sampling schemes.

A top-hat[a] particle detector could be modeled by replacing the existing sensor definition routines. Indeed, the number of sensors in the current model has been varied between 2 and 36.

[a] E.g. Young et al. (1998).

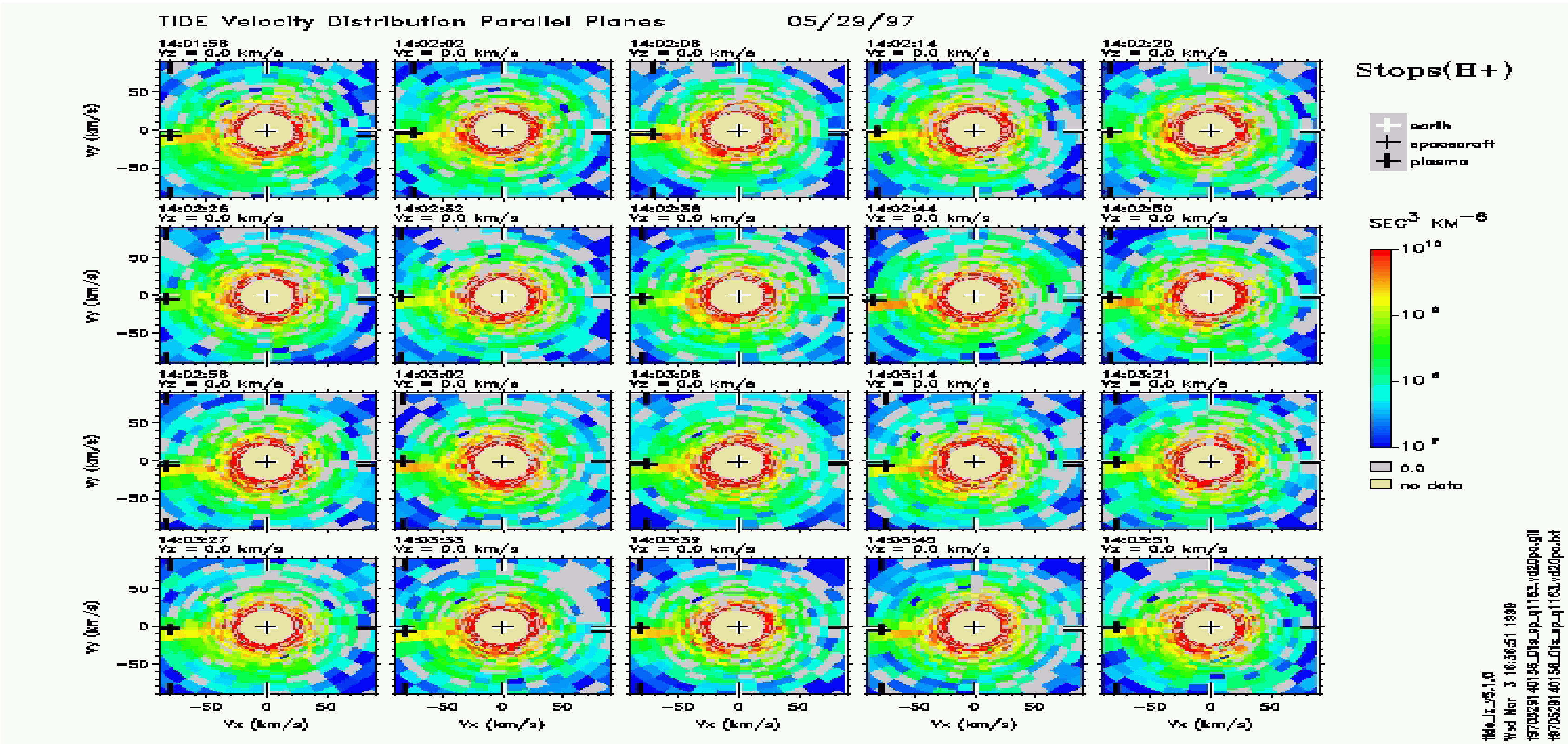


Figure 1: Example data from the TIDE particle spectrometer.

POLAR/Thermal Ion Dynamics Experiment PI T. E. Moore; http://satyr.msfc.nasa.gov/TIDE/

Velocity space sampling (raw)

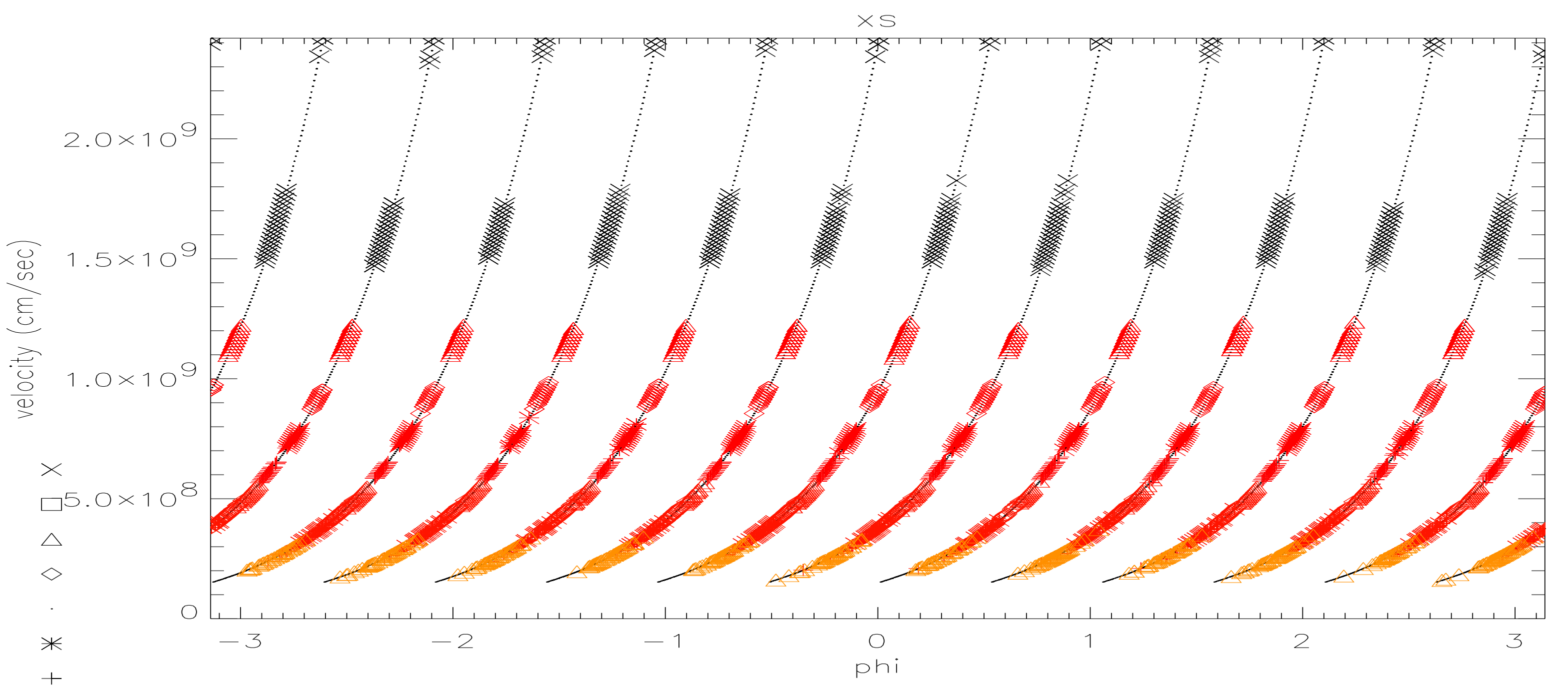


Figure 2: PMA Sampling. Notice the lack of data at very small velocities and the periodicity of the data in phi. Symbols and colors track datapoint values. Data have been projected onto the phi-velocity plane.

Physically based moments[a]

Let ξ_i denote the direction and velocity state of the detector during measurement i: $\xi_i = \{v_i, \theta_i, \phi_i\}$. With this definition we write the estimate distribution function at ξ_i as f_i.

Density:

$$n = \int d^3v\, f(\mathbf{v}) \tag{1}$$

Velocity:

$$\mathbf{u} = \frac{1}{n} \int d^3v\, f(\mathbf{v})\, \mathbf{v} \tag{2}$$

Pressure tensor:

$$\mathcal{P} = m \int d^3v\, f(\mathbf{v})\, (\mathbf{v} - \mathbf{u})(\mathbf{v} - \mathbf{u}), \tag{3}$$

[a] For students of the PMA, at this writing, the code calculates expectations of powers of the velocity $\mathbf{v}$.

where we interpret the dyadic product in Einstein's notation thusly: $\mathbf{ab} = a_i b_j$. The tensor product is defined by: $\mathcal{P} \cdot \mathbf{v} \equiv \mathcal{P}_{ij} v_j$.

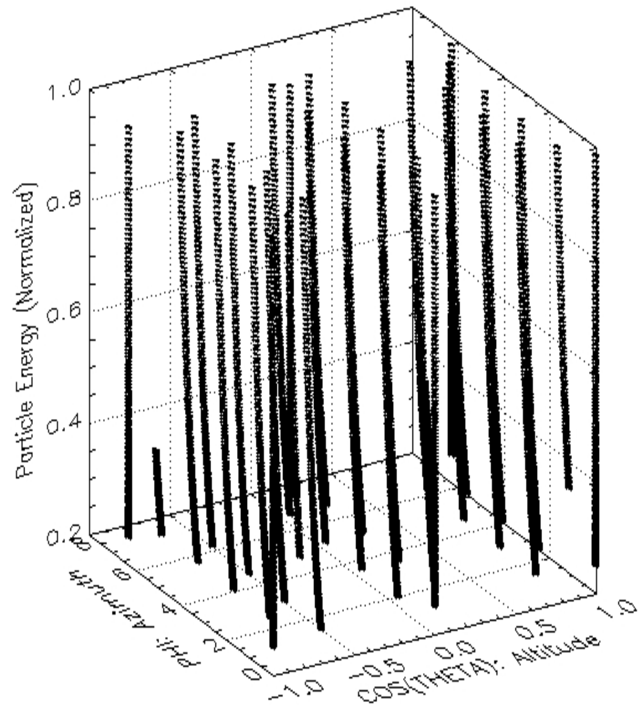
Heat flux:

$$\mathcal{Q} = \frac{m}{2} \int d^3v\, f(\mathbf{v})\, v^2 \mathbf{v} - \left[\frac{1}{2}\mathbf{u}\,\mathrm{Tr}\mathcal{P} + \mathcal{P} \cdot \mathbf{u} + \frac{m\,n}{2}\mathbf{u}\,u^2\right], \qquad (4)$$

where $u = |\mathbf{u}|$.

We can evaluate the integrals Eq. 1–4 by numerical quadrature because we know the integrands as functions of the data: ξ_i and f_i. Example data is shown in Figure (2) above.

Preparation for quadratures: To calculate an integral defined on irregularly sampled data, we use a tetrahedral grid with the nodepoints identified with the datapoints (Hirsch 1990).



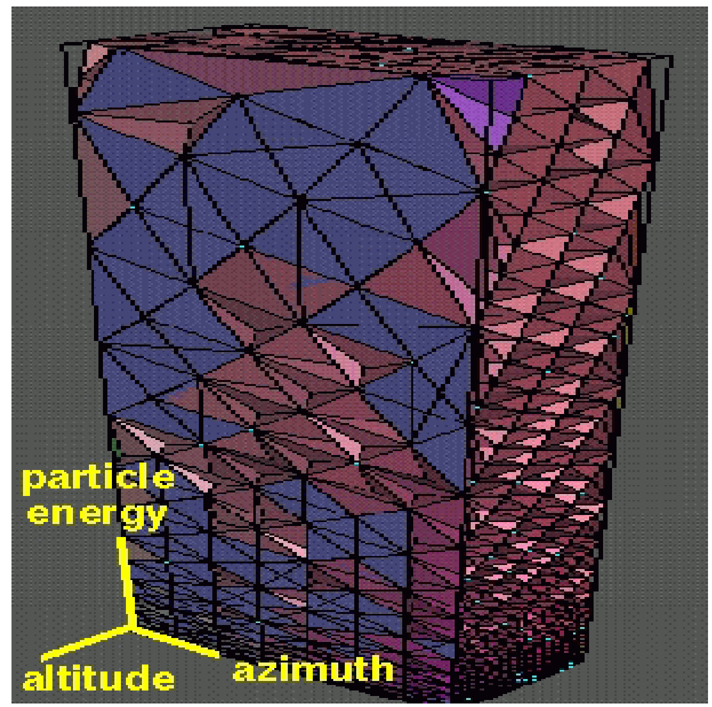


Figure 3: Left: irregularly gridded PMA data. Right: Delaunay triangulation of PMA data.

Velocity space sampling (patched and periodic)

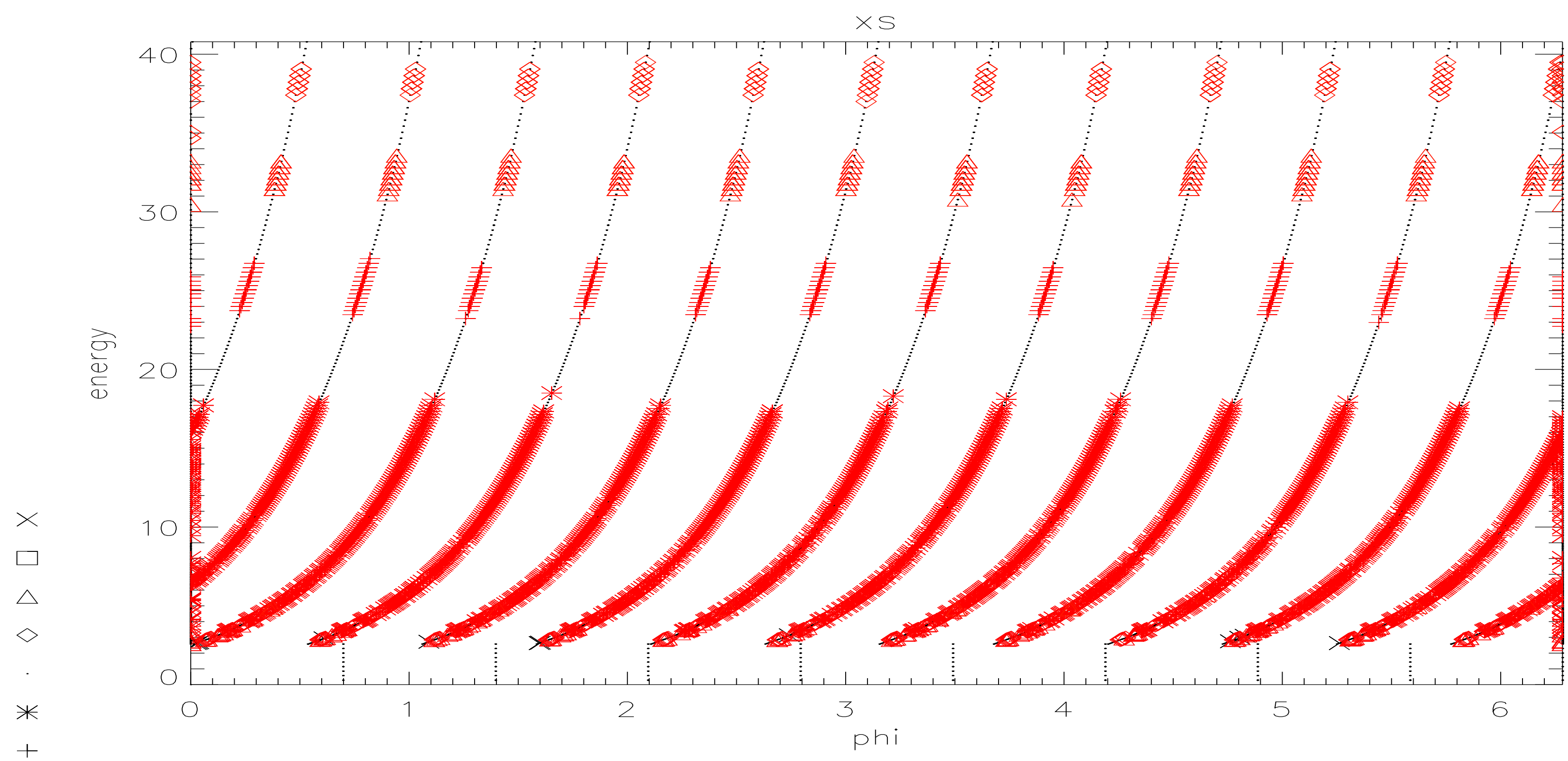


Figure 4: Data augmented with PMA-determined patch and periodic boundary conditions. Data is triangulated, ready for quadrature.

Patching

To improve the accuracy of the numerical quadrature, we fit a model to the data and use this model to estimate the value of the PDF where data is not available.

Singular Value Decomposition[a] is used to fit a 10 parameter model to $\ln f$. The parameters are of the following form:

$$\ln f \approx B_0 + \mathbf{B} \cdot \mathbf{v} + \mathbf{v}^\top \cdot \mathcal{B} \cdot \mathbf{v}, \tag{5}$$

where $\mathcal{B}$ is an upper triangular 3×3 matrix. Note that $\ln f$ is a vector of data, in the computer scientific sense, indexed by $\mathbf{v}$.

[a] The SVD of $\mathcal{V}$ is: $\mathcal{V} = U_\mathcal{V} \operatorname{diag}(\lambda) V_\mathcal{V}^\top$, where λ denotes the eigenvalues of the matrix, $\top$ denotes transposition, $U_\mathcal{V}$ is column-orthogonal, and $V_\mathcal{V}^\top$ is orthogonal. Replacing singular values of λ^{-1} with zero determines an operator that is the closest to being an inverse of $\mathcal{V}$, Ueberhuber (1997).

Hence the model $\ln f$ is linear in the parameters B, Eq. 5 can be rewritten:

$$\ln f = \mathcal{V}\, B, \tag{6}$$

where $\mathcal{V}$ is defined to make Eqs. 5 and 6 equivalent. Furthermore, $\mathcal{V}$ depends on the velocities $\mathbf{v}$ at which the data was obtained. We could write:

$$\ln f_j = \left(1,\, \mathbf{v}_j,\, \mathbf{v}_j^T..\mathbf{v}_j\right)\, \mathbf{B}. \tag{7}$$

One can test the patch to f by checking for kinks or non-monotonicity.

The locations of the patch points can be seen at low energies in Figure 4 above.

Periodicity in phi

The triangularization scheme to which we had access does not obviously support periodic boundaries. The correct course of action is to modify the interconnection matrix defining the triangulation so that the data structure is truly periodic in phi.

The more quickly realizable approach we performed is as follows.

1. Prior to triangularization create boundary points on the $\phi = 0$ and $\phi = 2\pi$ boundaries.

2. After triangularization set boundary datavalues by interpolation.

3. Make sure points aliased between the two phi-boundaries are self consistent.

Data distribution

We attempt to keep CPU-CPU communication to a minimum because off-CPU/off-node communication latencies and bandwidths can severely degrade computational performance.

Researchers wish to apply trial functions (e.g. spherical harmonics) to energy shells of the PDF. Therefore, we keep PDF data from the same energy shell on one processor to reduce communication costs. Interfaces between energy shells are shared by two CPUs.

Program flow

1. Model observations on one processor.
2. Read observations and distribute across processor nodes.
3. Use a parallel SVD from ScaLAPACK () to obtain patch-fit.
4. Create points on phi boundary and at low energies.
5. Triangulate each shell (in ϕ, θ, energy coordinates) on each processor using the non-parallel routines from GEOMPACK (Joe 1991).
6. Set values at those points where measurements do not exist.
7. Evaluate local quadratures to determine local contributions to plasma moments.
8. Reduce local contributions to moments to one (root) CPU.
9. Save and assess the plasma moments.

Results

We have been shaking down PMA for the past three months, and it is currently running without incident. In fact, the major questions regarding the results of PMA now deal more with the instrument model and the sampling scheme than with the PMA data analysis. The accuracy of the moments depends on the quality of the data obtained.

Though little effort has been invested in tuning the code, the code can make good use of multiple CPU computers (Figure 5).

Finally, the ratio of the input data (PDF samples) to the output data (plasma moments) has been measured to be as much as ~ 5000.

Speed-up scalability for PMA

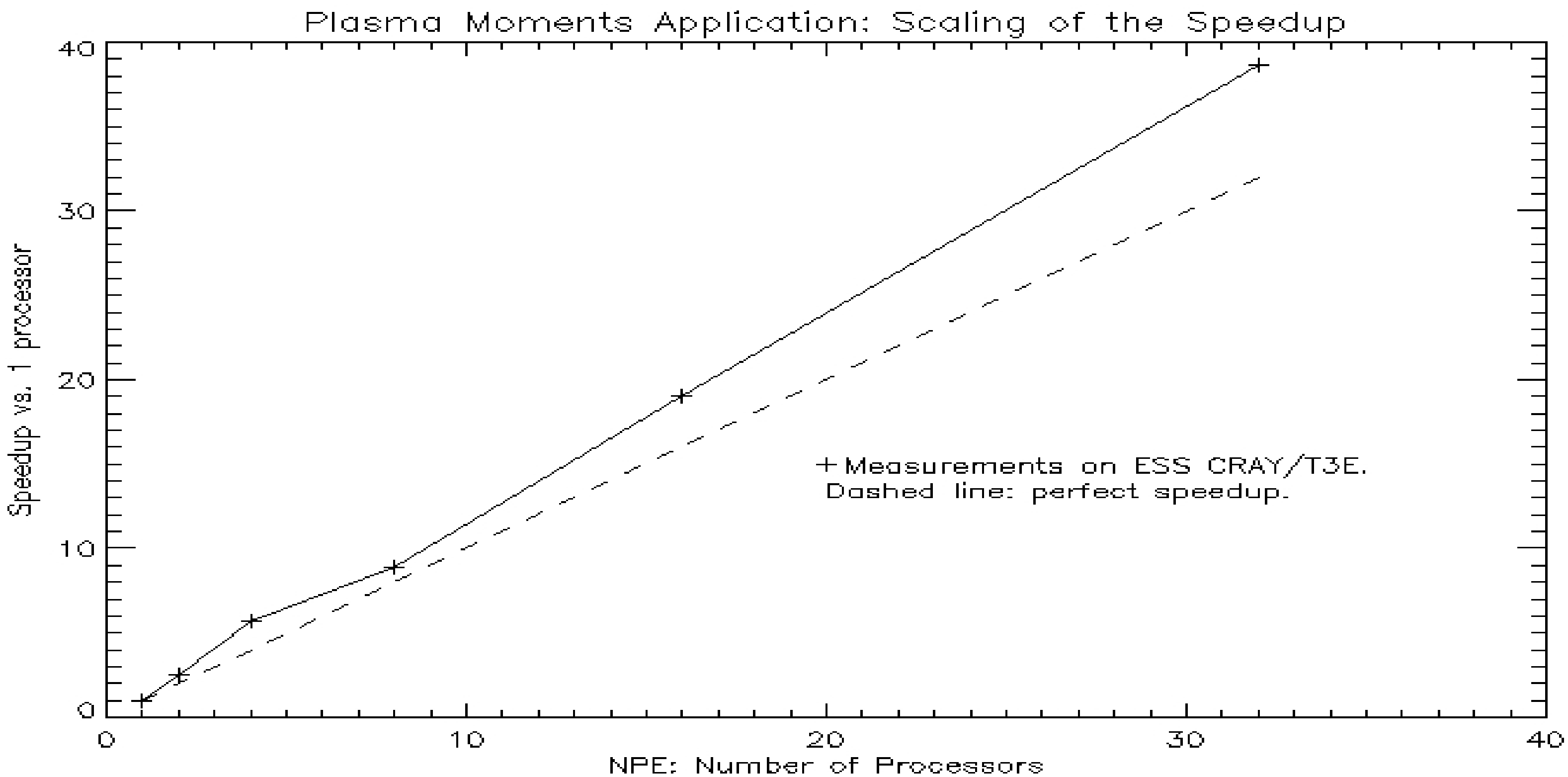


Figure 5: PMA speedup for different numbers of processors.

Summary

We have implemented a program that models particle spectrometer observations and then reduces these observations to plasma moments.

The observation model can support arbitrary velocity space PDF sampling schemes.

The data analysis step operates well on multiple processors provided there is sufficient data or work for each processor to perform. We showed how well PMA speeds up as the number of CPUs increase.

Reduction to plasma moments provides large science-based data compression.

Plans

Our plans are essentially threefold:

1. Continue to develop and improve the code,
2. Enhance the realism of the models, and
3. Experiment with control schemes that can use PMA output for autonomous science instrument control.

Acknowledgements

This work was supported by the NASA/HPCCP'S REE Project administered at JPL under the direction of R. Ferraro.

References

- Blackford, L. S. et al., 1997, *ScaLAPACK Users' Guide*, SIAM, Philidelphia. Also http://www.netlib.org/scalapack/slug.
- Hirsch, C., 1990, *Numerical Computation of Internal and External Flows*, J. Wiley & Son.
- Joe, B., 1991, GEOMPACK - a software package for the generation of meshes using geometric algorithms, *Adv. Eng. Software*, 13, pp. 325-331.
- Ogilvie, K. W., Scudder, J. D., and Doong, H., 1978, *IEEE Transactions on Geoscience Electronics*, Vol. GE-16, No. 3. p 261-265.
- Ueberhuber, C. W., 1997, *Numerical Computation*, two vols., Springer Verlag.
- Young, D. T. et al., 1998, in *Measurement Techniques in Space Plasmas: Particles*, Geophysical Monograph 102, AGU, pp. 313-318.