科学技術計算分科会 選出
SS 研 HPC フォーラム 2010 より

海外招待講演

# Extreme Computing: Challenges, Constraints and Opportunities

Anne Trefethen
Oxford University

# Extreme Computing: Challenges, Constraints and Opportunities

Anne E Trefethen

## Preamble

This paper is a handout to support the presentation at the Fujitsu HPC Users Group Meeting on 26th August, 2010.

## Abstract

Scientific applications require an ecosystem of computational infrastructure that can allow distributed collaboration and computation, large-scale simulation and analysis and appropriate consideration of data driven science. The ecosystem requires the integration of high-performance computing, in some cases to the exascale, cloud computing, databases, high bandwidth networks and the software and people infrastructure that enables the effective use of the components.

In this presentation we will consider some of the challenges and constraints that drive the development of the computational infrastructure and its components – including computational models, energy and knowhow; and the opportunities that are presented, in terms of new science applications and new algorithmic approaches. There is a particular focus on energy.

## Key words

Exascale, extreme computing, energy efficiency, cloud computing

## Introduction

Advanced computing is an essential tool in addressing scientific problems of national and scientific interest, including climate change, the virtual human, new materials, next-generation power sources and astrophysics, but as importantly it is equally essential to solve commercial and industrial problems in financial modelling, engineering, and real-time decision systems.

Extreme computing has moved from being a somewhat esoteric interest of a few scientists to a necessity for any computational scientist or developer who uses simulation as a tool. A shift has occurred in recent years as the processor chip designs mimic the architecture of high-performance computers, with multiple processing cores on a single chip, making efficient programming of a single processor computer as complex as it once was to develop software for a high-performance computer.

The interface to extreme computing is also changing with increasing availability of infrastructure and services through Cloud Computing enabling access to both high-performance and high-throughput computing.

The computers and computer systems, however, are still largely designed with little thought to the applications that might use them and often arrive to the users with little software infrastructure in place. The development of the required software infrastructure is becoming increasingly complex and if we are to be effective in the use of these systems there is a need to address this through appropriate abstractions, programming languages and tools to support application developers and users.

A further pressure point for extreme computing is the constraint that energy requirements are having, and will continue to have, on the developments of computer architectures and their use. This becomes increasingly important both in terms of the general computing infrastructure which is becoming a significant component of national use of electricity [1] and for the high-end computer systems, with the predictions of the exascale energy requirements up to 100MW [2] – more than would be required by a small town. Given that a standard dual-core laptop is equivalent to one of the Top500 machines of only 12 years ago we can expect these issues to be pervasive as we go forward.

This presentation provides an overview of the challenges that are faced in extreme computing, together with some of the ongoing activities to provide a roadmap for future developments. Focussing on energy as a driver there is a consideration of some of the research and activities across the ecosystem of computing and in particular we consider that required for the Square Kilometre Array [3] which provides an excellent co-design vehicle[1].

Ultimately our interest is in providing a framework to allow the development of applications and algorithms that are energy-aware and will allow the user to make choices regarding the objective optimized in the software – be it energy performance, price performance or indeed computational performance.

[1] A *co-design vehicle* is a concept described in [ref] and is essentially an application that is developed in conjunction with the design of the software and hardware infrastructure together.

## Extreme Computing Challenges and Constraints

There have been many studies during the last several years that have addressed the requirements of extreme (or exascale) computing [4, 5]. Here we will provide a general overview with some specifics from a recent study in the UK focussed on the development of a roadmap for high-performance computing [6] and the roadmap of the International Exascale Software Program [7] but the reader is referred to the original texts for the full discussion and details.

The HPC/NA roadmap activity in the UK brought together computational scientists, mathematicians and computer scientists to understand what might be the key issues for the future of extreme computing both in the UK and within an international context. The activity held a number of workshops and had individual communications with scientists and groups in the UK and a review of the literature from other national activities was included to provide the international perspective. It did not address hardware challenges directly, but the whole discussion was of course motivated by the changing face of computing hardware. The IESP roadmap provides a good overview of the hardware expectations [8]. As we go forward, we can expect that power and cooling limits will continue to constrain clock speeds; we will see a shift to multi/many core with attendant hierarchical parallelism an often with the additional complexity of hardware accelerators. This can be seen by the evidence of the Top 500 machines where multi-core machines and those with GPUs and the like are an increasing proportion [9]. Memory bandwidth will continue to be a bottleneck and this is likely to get worse. Chip design and chip to chip communication capabilities are likely to change however with 3D stacking and integration designs, for example [10] and the introduction of high-speed optical communications [11].

The HPC/NA roadmap activity found there were five major themes that dominated the discussions.

### Theme 1: Cultural Issues

There are gaps between disciplines that meant that often the team effort that might be required to bring a large-scale application together was not optimal and that appropriate knowledge and expertise might not be captured within such an effort. This is a common issue and many national activities are investing to ensure that community actions are supported and interdisciplinary teams created.

Differences in discipline communities were also identified in that some application domain scientists are used to sharing models and codes, and reusing other software developed by other groups; while for other domains this approach is almost completely alien with codes being entirely developed within a particular group and little use being made of libraries or other third-party software.

Similarly there is a need to ensure that the software activities are not only focussed at a national level but are within an international context. It is the case that several application codes in the UK are dependent up on software that has been developed in programmes in other countries and vice versa. These dependencies are a risk to the sustainability of the software and that risk can only be mitigated by international collaboration.

## Theme 2: Applications and Algorithm development

There are many issues that need to be addressed in the area of application and algorithm development. These include ensuring that a component approach to development is taken where appropriate such that algorithms that might underpin multiple application areas are developed as such. Having noted that, many applications involve multiple models at different scales or for different elements of the application and integration of components is not possible without appropriate standards for data models and formats, interoperability of programming models, and lack of knowledge of error propagation through the integrated system. Clearly this is not a problem unique to extreme computing, but is one that as a community we have yet to address fully.

The memory hierarchy of high-performance computers is on course to get even more complex which will drive the needs for hierarchical algorithms to deal with bandwidth across the memory hierarchy together with software strategies to mitigate high memory latencies. This will in turn drive the need for algorithms to be dynamically adaptive, perhaps as components of an active library, and of course scalable much beyond the present status quo. There are few applications that can scale to petaflops levels let alone toward exascale. Such scaling can only be achieved with appropriate portioning and load balancing and effective data management. Data-intensive science [12] is increasingly important and the ability to manage large amounts of data effectively and efficiently is crucial [13]. Moving data takes energy, at every level of the system – this is a point we will return to later.

## Theme 3: Software Challenges

Development of software is becoming increasingly complex and without appropriate software engineering, in the context of an exascale system, will be un-maintainable. Application scientists are not usually software engineers but there is a need to ensure that by some means appropriate practice is adopted. Frameworks and tools to support software development will be needed together with compilers and code generation tools that can provide a layer of abstraction for application scientists.

## Theme 4: Sustainability

The HPC/NA Roadmapping activity identified a general concern regarding the sustainability of application codes, software libraries and skills (we consider skills in the next section). This issue is integral to that of programming models, interoperability and also the cultural issues above. Scientists are naturally loathed to invest a great deal of effort in the development of software that will last only the lifetime of a particular computer architecture, compiler or other dependent component of the environment. We can only address this issue as a community as we agree in the adoption of standards and open source mechanisms.

## Theme 5: Knowledge Base

It is important to ensure that computational scientists have the right set of skills, and this set of skills needs constantly updating. Within the UK activity it was also discovered that there was a lack of awareness by some members of the community of existing libraries/packages. Maintaining a strong knowledge base will require education for graduate students as well as the opportunity for life-long learning. The report on exascale computing for energy and environment [14] notes "The current belief is that the broad market is not likely to be able to adopt multi-core systems at the 1000-processor level without a substantial revolution in software and programming techniques for the

hundreds of thousands of programmers who work in industry and do not yet have adequate parallel programming skills."

### Other issues:

Since the last of the roadmapping workshops Cloud Computing [15] has matured and is now very much part of the computational environment. It offers a different interface to distributed, and in some cases, high-performance computing and as such is part of the evolving ecosystem for application development. As data-intensive applications dominate it is possible that the business models underpinning Cloud computing might offer benefits to scientists. These issues will likely be resolved as computational scientists become engaged with the various vendors but there is no question that the scale of Cloud computing infrastructure may offer benefits in terms of energy efficiency and from that point of view we will return to them as part of the ecosystem later.

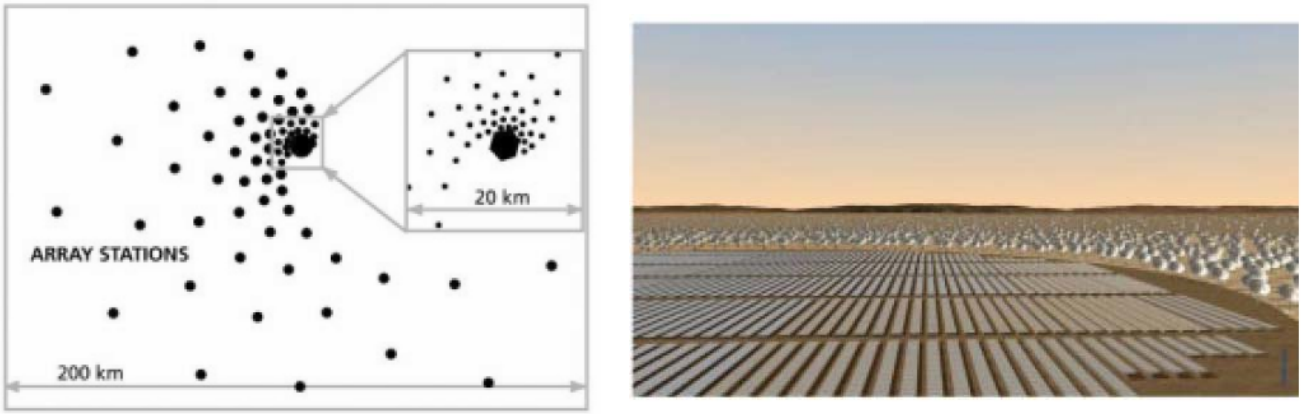
## Co-Design Vehicles and the Square Kilometre Array

During the workshops and activities of the Department of Energy in the USA addressing the requirements for exascale computing, the notion of a co-design vehicle has been adopted. At present, other than in a few specific cases [15, 16, 17], a computer system is designed with little thought for the applications that will use it, and the software is generally designed to try to fit the hardware. In the co-design model the two are seen as part of the same system and the design of each is done through collaboration. In the extreme this might result in computer systems that are only useful for a given application such as [15, 16, 17] but the model can be adopted to enable more generic progression with the evolution of the computer architect and algorithms being better aligned and providing better capability and specifically energy performance.

We are interested in the computing challenge provided by the development of the Square Kilometre Array [?], the next generation radio telescope. The telescope is in the design phase with the anticipation that construction of the first phase will begin in 2016 with the full telescope completed and in operation by 2022. The SKA will likely be located in either Australia or South Africa, in a desert so as to have little or no interference, but is a collaborative effort of over 50 groups in 19 countries.

The present design [18,19,20] has a combination of aperture arrays in the core and up to 3000 phased array feeds on dishes and a collecting area of approximately one square kilometre with receptors extending out to a distance of 3000km from the centre of the telescope (figure 1). It will allow a sensitivity of more than 50 times that of existing telescopes, and 10,000 times the survey speed and will provide data to answer fundamental science questions on gravitation and magnetism, galaxy formation and even the question of life on other planets. The design of the SKA is developing through design studies based on the science requirements, Pathfinder telescopes that provide experience of design options, and technology capability considerations.

*Figure 1 Possible configuration of SKA receptors and artist's impression of SKA core(from [21])*




The SKA provides a fabulous information technology challenge with a typical data rate from each dish antenna on the order of 100Gbs$^{-1}$ aggregating to over 100Tbs$^{-1}$ [21] and need for Exaflop computation [23] for post-processing. The infrastructure required to support the various science cases will need to range from real-time capability to transport and analyse the data at these high-data rates and the capacity to store and "publish" the data for later analysis and interpretation by the global astrophysics community. The computational systems will likely range from specifically designed FPGA-like units to exascale computing and Cloud-like data centres. The communications infrastructure will range from intra-chip and inter-chip with optical fibre to the correlator and on to a high-performance computer, to trans-oceanographic with the latter having data rates of at least 100Gbs$^{-1}$ over the general network providers. The SKA will succeed or not depending on both the physical implementation of the telescope design and the software infrastructure that will enable it. The software infrastructure required to realise this information technology challenge is itself has been identified as > 2000 person year task [22] but even this may not take full account of the complexity of the task.

*Figure 2: SKA computational requirements in context of Top500 [23]*


The fundamental algorithms in the SKA data pipeline include beamforming, gridding, convolution and filtering algorithms. Cornwell et al [23] estimate the computational requirements in the context of the Top500 and show them to be beyond the scale of projected performance in the timescale required (figure 2).

Alongside the challenges of the computational infrastructure are the related challenges of powering the infrastructure. This includes the power

required for the core antennas (~30MW) and the remote stations ~0.5MW) and the various computational components including high-performance computing (~40MW) data transmission (~?MW) and Cloud (or equivalent) provision (~?MW). Energy provision is a major design factor in the delivery of the telescope – plans to mitigate the energy constraints include renewable energy sources (sun) providing the station power and it is easy to see that the SKA could possibly be the largest Green IT project ever to be considered.

## Counting Joules

As indicated above the ecosystem of computational resources required to enable the SKA provides a plethora of challenges and opportunities where energy efficiency is concerned.

Moving data takes energy whether the data is moving from L2 cache on chip or within a transatlantic Data Cloud. Effective management of that data communication is a major component of any optimal energy model. There has been a great deal of research on wireless network communications and sensor networks, where devices are most often low-energy devices with battery constraints. Indeed a lot of research has been done in general on low-energy devices including computational algorithms from which we might learn. The existing communications network across the globe relies on hundreds of thousands of switches and routers and these, unlike wireless or mobile infrastructure, do not power down when idle. The network is a complex system of different technologies and it is difficult to predict the energy consumption under different circumstances so sending data 10 times as fast might use interfaces that use 100 times the amount energy or in some cases (e.g. newer optical ones) 1000 times less [24]. A good overview of network energy costs for realistic configurations can be found in [25]. There are a number of research efforts considering the issues around Green network communications including INTERNET [26] and others can be seen at recent conferences [27,28]. Of course as energy-aware systems are developed there are still issues of what they optimise – usage or cost? Qureshi et al [29] illustrate effective ways in which energy-aware data centres can optimise cost by moving computation to nearby states where electricity costs are less.

Optimizing energy usage in large-scale data centres and Clouds is almost a science in itself. McKinsey and the UpTime Insitute [1] indicate that the energy used by data centres in the US is becoming a significant percentage and is likely to overtake airlines in terms of carbon emissions. The report states that the average data centre uses as much power as 25,000 households, but that estimate is probably somewhat out of date as in the last few years Cloud provisioners have built very –large scale data centres across the US [30]. On a somewhat smaller scale at the Oxford Supercomputing Centre (OSC) we have developed software that intelligently powers down components of the systems at times of under utilisation. The indications are that this will provide significant savings.

The McKinsey report identifies a number of issues around the effectiveness of data centres including siloed organisations and limited transparency that match very well with the findings of the HPC/NA for that community. McKinsey make the recommendation that metrics be defined that are not only measuring the facility but are linked to the applications using it and the processes integral to it. A consortium called the Green Grid is now in place with the aim to develop standards and best

practice for data centres. The recommendation regarding metrics is one that we, as a community, should also take on board as we develop exascale technology.

At the computing system level there are many approaches to energy efficiency. Again there is a large knowledge base in the low-energy systems community that we should consult. Indeed an approach that might be considered for some application areas builds directly on the low-energy embedded microprocessor technology to build petascale systems. This approach is under development for the Green Flash system [31, 32] at Berkeley National Laboratory where researchers are collaborating directly with Tensilica, Inc. to explore the use of Tensilica's Xtensa processor cores in building a computer to model clouds (that is clouds in the weather system not as a computer system!). They believe that a specially designed core could get 10–100 times better performance per watt. The Green Flash design will have on the order of 20 million processors consuming less than 4 MW whereas the equivalent in conventional microprocessors would require around 200MW.

With appropriate systems support it is possible to provide hooks into the operating system to allow spindown policies to be enacted, adaptive placement of memory blocks, agile use of component devices and of course energy-aware routing [33].To make use of this level of support it would be helpful to have tools that can provide a mapping from the higher-level action to the underlying activity.

The power cost for a device is proportional to the frequency cubed. This means that a multi-core device that is running at a lower clock rate is bound to provide the potential for higher performance at a lower energy cost. Interesting issues arise when there are multiple such devices to hand and the algorithm designer can chose to optimise the energy usage for a given computational performance. At this point in time the SKA community are considering the use of GPUs, FPGAs and multi-core chips [34, 35]. Nieuwpoort et al [34] show some interesting results in this area where they have used a variety of chipsets to underpin the computation to correlate signals. The results are particularly interesting in respect to the achieved efficiency vs Gflops/Watt where although a given device might underperform in terms of percentage efficiency it may still perform better in Gflops/Watt. The Green500 provide good reference indicators for feasible platforms at this time [36].

Within the Oskar project [37] we tackled the Digital beamforming for the aperture array components of the SKA that pose considerable computational challenges [38]. The proposed algorithm provides a hierarchical algorithm for beamforming using a simplified and flexible computational approach of direct matrix-vector multiplies rather than FFTs that provides a reduced data rate and a computational cost of forming beams of 1TFlop as opposed to 20TFlops for the FFT approach.

Within PrepSKA [39] Savlini et al [40] have developed a Pipeline for Extensible Lightweight Imaging and Calibration (PELICAN). This framework for parallel quasi-real time data processing offers two deployment options either with the server supplying multiple pipelines or the pipeline connecting to the data stream directly. PELICAN allows reuse of modular components and will be deployed on LOFAR [41] interferometer stations to allow all sky calibration and imaging, and pre-processing for pulsar searching. The modular framework allows appropriate computational components to be implemented on appropriate devices, in this case GPGPUs.

## Tools, Benchmarks, metrics

While benchmarks for datacentres and systems are now relatively mature [42, 43] there is still work to be done at the numerical algorithms level. On the exascale roadmap [8] it is suggested that standards to support energy-aware algorithms should be agreed in 2014/2015 with energy-aware libraries available in 2016. We have our work cut out!

Although for the SKA specific algorithms of interest we can model and implement across different platforms there is a lack of agreed metrics, tools and benchmarks to provide the community with clear evidence to support design features. This is true across the board for computational algorithms although there are some efforts in this direction.

There are a number of tools for a variety of platforms that will allow measurements of device and sub-device power usage. Kirk Cameron and his group have created a profiling tool that is an open source software system together with hardware power measurement devices, PowerPack [44]. Using the profiler they have analysed the HPC Challenge Benchmarks [45] and been able to provide very clear analysis of the behaviour of the algorithms in terms of energy used.

It may be time to consider again what benchmarks should stress. As we consider adaptive algorithms that are able to autotune to heterogeneous sets of processors we may need to consider a different set of characteristics and metrics that are captured by existing suites. There may be instances where replacing data movement by computation provides a saving in energy while maintaining computational speed.

## Conclusions

Energy constraints provide one of the major challenges for extreme computing in the future. There are opportunities at every level of the computational infrastructure to address the challenges through both efficient physical hardware but also through more sophisticated use of the physical infrastructure. An intelligent software infrastructure could reduce the energy use on chip, within a processor, between processors, between computers, across Cloud platforms and indeed over international boundaries.

Ultimately our own interest lies in the development of energy-aware algorithms for extreme computing. We believe that to achieve such algorithms and to see the benefits there will need to be significant investment in the support of tools and low-level mechanism to allow the profiling of energy use across platforms. Equally important are standard metrics and computational benchmarks that will allow agreement on measures of “success”. Just as the linpack benchmark has come to be seen as the measure of the performance of a computer system we need the equivalent application/algorithm benchmarks that capture the energy characteristics of any given system.

The future holds continuing complexity in computing systems from the combinations of computing within the whole ecosystem to the heterogeneity of chips on our laptops. The most effective algorithms at any given point in time for any specific application are going to depend upon the choices that can be made given the hardware configurations. In general, at present we cannot even articulate those choices let alone provision the most appropriate algorithm. This challenge of

articulation and optimal algorithm design can only be addressed through a co-design approach with hardware, software and application scientists working together.

While we have focussed here on the energy aspects of the problem, and specifically as driven by SKA, there are any one of several other key issues that could and should be addressed including the complexity of the software development for such an infrastructure, the usability of the software systems, the data management and related semantic issues for such colossal data systems or the provision of imaging and data analysis in the Cloud. Each of these is a talk of its own – for next time.

## Acknowledgements

My thanks to colleagues on the prepSKA project Steve Rawlings, Aris Karastergiou, Stef Salvini, Ben Mort, Chris Williams, Fred Dulwich and Andy Faulkner. The HPC/NA project was in collaboration with Nick Higham, Iain Duff, Peter Coveney, Mark Hylton and Stef Salvini. I am grateful to Jeyan Thiyagalingam, Simon McIntosh-Smith, Jon Crowcroft and Jaafar Elmirghani for their input and suggestions.

## References

1. Revolutionizing Data Center Efficiency, McKinsey Company and the Uptime Institute, *www.**mckinsey**.com/.../Revolutionizing_Data_Center_Efficiency.pdf*
2. www.exascale.org
3. The SKA project: www.skatelescope.org/
4. The DARPA Exascale Study Report: http://users.ece.gatech.edu/mrichard/ExascaleComputingStudyReports/exascale_final_report_100208.pdf
5. The Department of Energy workshop on extreme computing: http://extremecomputing.labworks.org/index.stm
6. The HPC/NA Roadmap: www.oerc.ox.ac.uk/research/hpc-na/roadmap
7. The International Exascale Software Project: www.exascale.org
8. The IESP roadmap: http://www.exascale.org/mediawiki/images/4/42/IESP-roadmap-1.0.pdf
9. The Top500 machines: www.top500.org
10. Intel http://www.eetimes.com/electronics-news/4204959/Intel-Silicon-Optics
11. 3D DRAM Design and Application to 3D Multicore Systems, Hongbin Sun, Jibang Liu, Rakesh S. Anigundi, Nanning Zheng, Jian-Qiang Lu, Kenneth Rose, Tong Zhang, IEEE Design and Test of Computers**,** vol. 26, no. 5, pp. 36-47, September/October, 2009.
12. The Fourth Paradigm: Data Intensive Research,http://research.microsoft.com/en-us/collaboration/fourthparadigm/
13. The Data Deluge: An e-Science Perspective, A.J.G. Hey and A.E. Trefethen, in *Grid Computing - Making the Global Infrastructure a Reality*, Berman, Fox, Hey, eds., pp 809-824, Wiley and Sons, 2003.
14. DOE Report on Modeling and **Simulation** at the Exascale for Energy and the Environment, June 2007
15. Above the Clouds: A Berkeley View of Cloud Computing, Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson,

Ariel Rabkin, Ion Stoica, and Matei Zaharia, UC Berkeley Reliable Adaptive Distributed Systems Laboratory . http://radlab.cs.berkeley.edu/ February 10, 2009

16. The QCDOC project: *ukqcd.epcc.ed.ac.uk/community/qcdoc/*
17. The MD-Grape3 machine: http://www.peta.co.jp/index-en.html
18. The D.E.Shaw Machine*:* www.deshawresearch.com/
19. The Square Kilometre Array, http://www.skatelescope.org/
20. The Square Kilometre Array, Peter E. Dewdney, Peter J. Hall, Richard T. Schilizzi, and T. Joseph L. W. Lazio, Proceedings of the IEEE | Vol. 97,No. 8, August 2009
21. The Square Kilometer Array (SKA) Radio Telescope: Progress and Technical Directions, P.J. Hall, Richard T. Schilizzi, and T. Joseph L. W. Lazio, U.R.S.I, The Radio Telescope Bulletin, No 326, September 2008
22. SKA Memo 100: Preliminary Specifications for the Square Kilometre Array, R. T. Schilizzi, P. Alexander, J. M. Cordes, P. E.Dewdney, R. D. Ekers, A. J. Faulkner, B. M. Gaensler, P. J. Hall, J. L. Jonas, K. I. Kellermann, *http://www.skatelescope.org/PDF/memos/100_Memo_Schilizzi.pdf*
23. Scaling Mount Exaflop: from the pathfinders to the Square Kilometre Array, T.J. Cornwell, *Ger van Diepen, www.atnf.csiro.au/people/tim.cornwell/publications/MountExaflop.pdf*
24. Green Optical Communications—Part II: Energy Limitations in Networks, Rodney S. Tucker*,* IEEE JOURNAL OF SELECTED TOPICS IN QUANTUM ELECTRONICS
25. Private communication with Jon Crowcroft
26. First ACM SIGCOMM Workshop on Green Networking, August 2010, *http://conferences.sigcomm.org/sigcomm/2010/gncfp.php*
27. INTERNET: Intelligent Energy Aware Networks, *http://www.internet-project.org.uk/*
28. E-energy: First international conference on energy-efficient computing and networking, April 2010, *http://www.e-energy.uni-passau.de/*
29. Cutting the Electric Bill for Internet-Scale Systems, A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag and B. Maggs. SIGCOMM 2009.
30. Data Centres in the News: *http://www.datacenterknowledge.com/archives/2010/02/05/virginia-nc-battling-for-microsoft-data-center/*
31. The Green Flash computer project, *http://www.lbl.gov/cs/html/greenflash.html*
32. Towards Ultra-High Resolution Models of Climate and Weather , Michael Wehner, Leonid Oliker, and John Shalf, *International Journal of High Performance Computing Applications May 2008 22: 149-165, doi:10.1177/1094342007085023*
33. Every Joule is Precious: The Case for Revisiting Operating System Design for Energy Efficiency, Amin Vahdat, Alvin Lebeck, and Carla Schlatter Ellis, Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system Pages: 31 – 36, 2000 ISBN:1-23456-789-0
34. Using many-core hardware to correlate radio astronomy signals, Rob V. van Nieuwpoort, John W. Romein, Proceedings of the 23rd international conference on Supercomputing, pages: 440-449, 2009, ISBN:978-1-60558-498-01
35. Evaluating Multi-Core Platforms for HPC Data-Intensive Kernels, Alexander S. van Amesfoort, Ana L. Varbanescu, Henk J. Sips, Delft University of Technology, The Netherlands
36. The Top Green 500 Computers, *www.green500.org*
37. The OSKAR project: http://www.oerc.ox.ac.uk/research/oskar

38. OSKAR: Simulating Digital Beamforming for the SKA Aperture Array, Fred Dulwich, Benjamin J. Mort, Stefano Salvini, Kristian Zarb Adami, and Mike E. Jones, Widefield Science and Technology for the SKA, SKADS Conference 2009
39. Preparation for SKA: *www-astro.physics.ox.ac.uk/~sr/prepska.html*
40. The Pelican framework: *https://wiki.oerc.ox.ac.uk/svn/pelican/slides/pelican-slides2.ppt*
41. The Lofar Pathfinder: *www.lofar.org/*
42. The SPEC power benchmark, *http://www.spec.org/benchmarks.html#power*
43. JouleSort: A Balanced Energy-Efficiency Benchmark, Suzanne Rivoire, Mehul A. Shah, Parthasarathy, Ranganathan, Christos Kozyrakis, SIGMOD'07, June 11–14, 2007, Beijing, China
44. PowerPack: Energy Profiling and Analysis of High-Performance Systems and Applications , Rong Ge, Xizhou Feng, Shuaiwen Song, Hung-Ching Chang, Dong Li, Kirk W. Cameron, IEEE Transactions on Parallel and Distributed Systems, 23 Apr. 2009. IEEE computer Society Digital Library. IEEE Computer Society, <http://doi.ieeecomputersociety.org/10.1109/TPDS.2009.76>
45. Energy Profiling and Analysis of the HPC Challenge Benchmarks Source, Shuaiwen Song, Rong Ge, Xizhou Feng, Kirk W. Cameron International Journal of High Performance Computing Applications Volume 23 , Issue 3 (August 2009), Pages: 265-276, 2009

# 10PetaFLOPS は
# 生命科学で何を実現できるか

理化学研究所
姫野 龍太郎

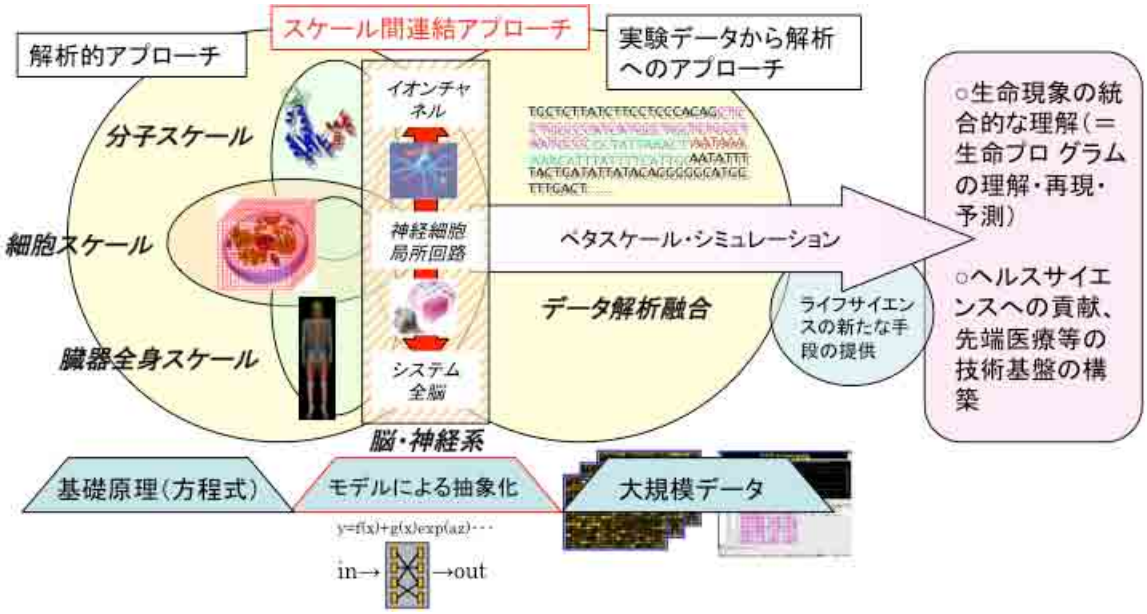
# 10PetaFLOPSは生命科学で何を実現できるか

理化学研究所　次世代生命体統合シミュレーション研究グループ
グループディレクタ　姫野龍太郎

Keywords
スーパーコンピュータ、生命科学、シミュレーション、分子、細胞、臓器、全身、脳神経系、マルチスケール

## 1. はじめに

次世代スーパーコンピュータは来年度から部分的に稼働が始まり、再来年度には全システムの稼働が予定されている。この稼働に合わせて、生命科学分野でのグランドチャレンジとも呼ばれる次世代生命体統合シミュレーション研究開発プロジェクト（ISLiM）が、2006 年秋から理研を中核拠点として行われている。このプロジェクトはペタスケールのシミュレーション技術によって、ライフサイエンスに仮説-検証型の新たな研究手段を提供し、生命現象を定量的かつ統合的に理解・予測・解明することを目指すと共に、創薬・ヘルスサイエンスサイエンスへの貢献、新規医療技術の実用化を図るものである。



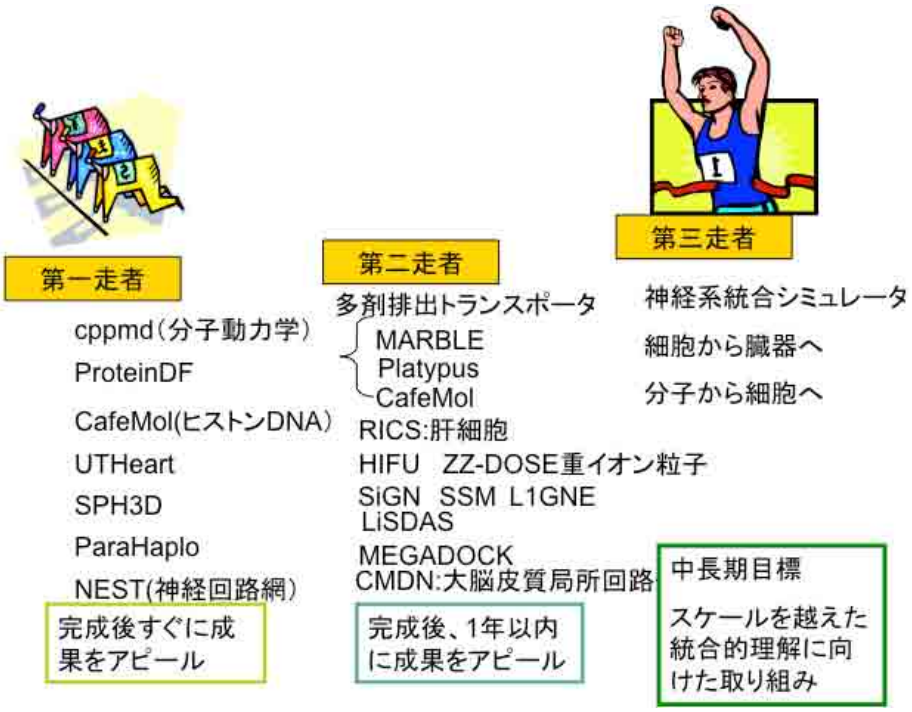

## 2. 推進体制とソフトウェア開発の状況

このような目的の実現のため、分子・細胞・臓器全身という３つの階層に分かれた研究チームと、大量の実験データから法則に迫るデータ解析融合チーム、更に脳神経系チーム、全てのチームを支える HPC チームの６チームを設置して取り組んでいる。

これらのチームで現在開発中のソフトウェアは量子化学や分子動力学から構造流体連成計算、脳の局所回路モデル、可視化ソフトウェア、並列化のためのミドルウェアなど、合計３４本にのぼる。これまでの開発で、並列化に関して 1,024 を超えるところまでテストが終わったものが 18 本、8,000 並列までのテストが終わったものは９本という状況である。今後は次世代スーパーコンピュータでの実効性能と実行時間の予測を行い、どの程度のモデルでどのくらいの計算をするかを検討してゆく予定である。

## 3. 現在の見通し

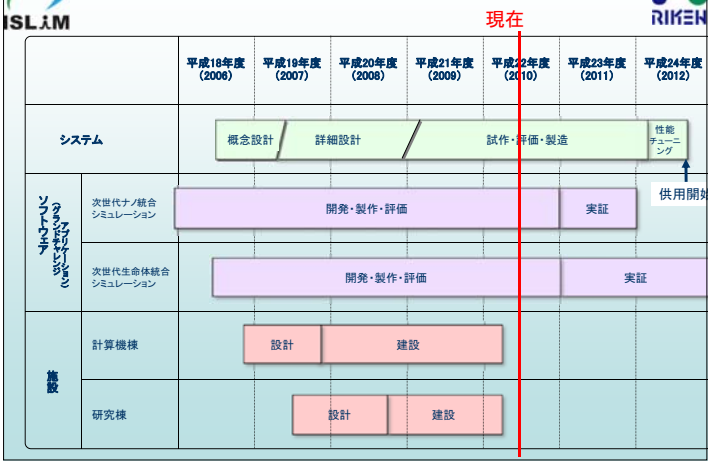
昨年から開発中のソフトウェアの並列性能などの開発状況を睨みながら、10PetaFLOPS の次世代スーパーコンピュータを使ってどんな問題を解くかを検討してきた。また、どの問題をどの時期に解いてゆくか、問題間の優先順位をどのようにつけるかを検討した。その結果、それぞれのソフトウェアを３つに分類、計算を行う時期に合わせて、第一走者、第二走者、第三走者として、波状的に成果を出してゆくこととした。開発中のソフトウェアの数が多いので、第一走者と第二走者の一部について、そのソフトウェアの概要と解こうとしている問題を紹介する。



# 10PetaFLOPSは生命科学で何を実現できるか

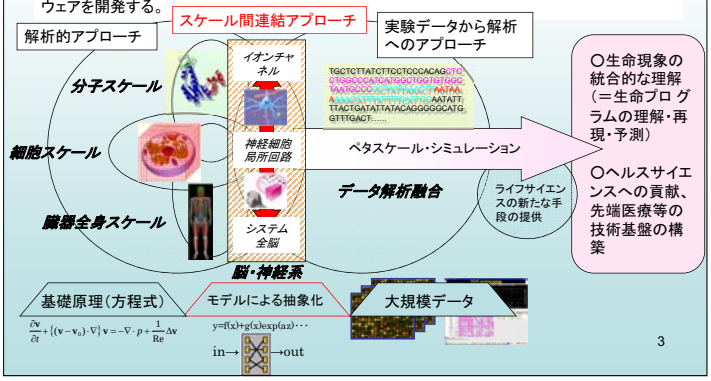
次世代生命体統合シミュレーション研究開発グループ
グループディレクタ
姫野龍太郎

## この研究開発の位置づけ
次世代スパコン開発プロジェクトの一部




## 研究開発の概要と達成目標（2008年10月以降）

基礎方程式に基づく解析的アプローチと、大量の実験データから未知の法則に迫る実験データから解析へのアプローチ、さらには多階層を連結するアプローチにより、異なるスケールの研究と実験データを統合的かつ有機的に結びつけ、ペタスケールという桁違いの性能を持つスーパーコンピュータの性能をフルに発揮し、生体で起こる種々の現象を理解し医療に貢献するためのソフトウェアを開発する。

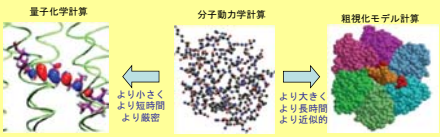





## 分子スケール研究開発

**開発目標**：量子化学計算（QM）・分子動力学計算（MM）・粗視化モデル計算（CG）を総合化する技術を開発、タンパク質や細胞の機能発現過程のシミュレーションにつなげ、細胞スケールとの有機的な連携を図る。ターゲット：多剤排出トランスポータ、脂肪酸代謝酵素反応

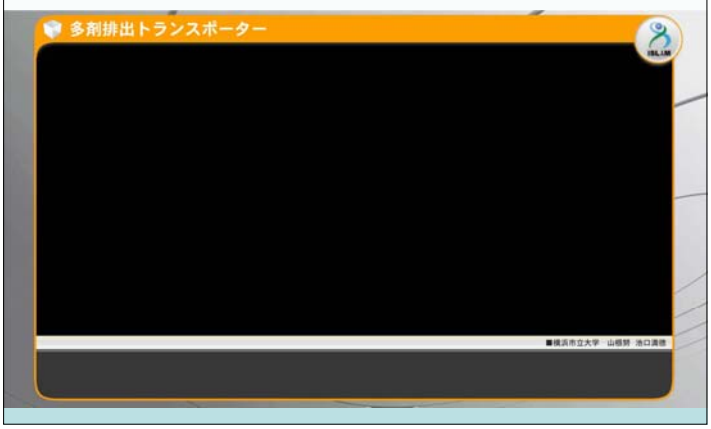
**開発計画**：QM、MM、CGの各スケールでプログラムの開発を進めるとともに、それらの手法を結合したQM/MM、MM/CG法によってマルチスケールシミュレーションを実現するためのプログラムを開発する。




**進捗状況**：QM/MM/CGの3階層それぞれで計算方法をプログラム化し、テストを続けながら開発、基本的手法のプログラム化が終わり、128-512並列程度まで性能が出るまでになっている。



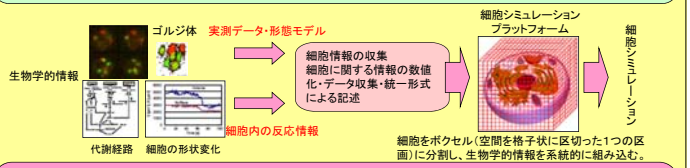
## 分子シミュレーションの可視化ムービー




## 細胞スケール研究開発

**開発目標**：ペタフロップス級の計算力を活かして細胞を100万のボクセルに区画、そこに包括的な実証データを組み込んだ肝細胞・肝小葉のシミュレーションの実現を目指す。薬剤等の作用や副作用の予測を行うシミュレーションの開発につなげる。

**開発計画**：生物の空間的情報を入れ込むことが出来るプラットフォームの開発と平行して、プラットフォームに実装するためのデータ収集とモデルの構築を進める。また、プラットフォームと各種データ・モデルとの有機的連携を図る。

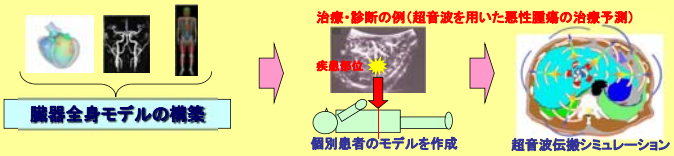



**進捗状況**：細胞を約3万個のボクセルに分割、複数のオルガネラを表現し、移流拡散方程式による物質移動をシミュレーションできる細胞シミュレーションプラットフォームを開発、E-CELL3をベースにした代謝反応がオルガネラ間も含めてシミュレーションできるようになった

## 臓器全身スケール研究開発

**開発目標:** ペタフロップス級の計算量を用いて、数時間で実行可能な臓器全身シミュレーションの構築を目指す。これにより、病態予測や治療予測等の医療支援ツールの開発につなげる。

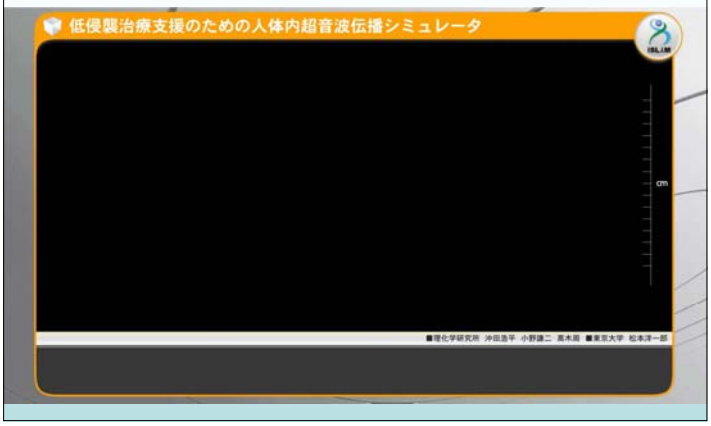
**開発計画:** 血管網、各種臓器、全身を3次元的に再現した臓器全身モデルを構築し、病態予測や治療予測を行う臓器全身モデルを次世代スパコンに実装、循環器系疾患を始めとする種々の病態のシミュレーションと、ガンなどの治療支援シミュレーション（放射線・集束超音波など）につなげる。




**進捗状況:** 1mm分解能の全身ボクセルモデルを開発すると同時に新しい構造流体連成手法と多媒質内の超音波伝搬解析手法を開発した。これらにより、外科手術シミュレーションや収束超音波によるガンの治療等へつなげる基盤を構築した。

## 血栓形成と輸送のシミュレーション

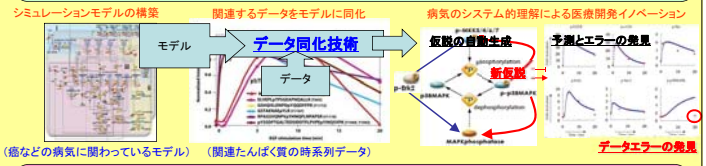

## 収束超音波による治療のシミュレーション




## データ解析融合研究開発

**達成目標:** ペタフロップス級の計算によって、創薬ターゲット探索や個人差を考慮した医療のための基盤情報技術の構築を目指す。これにより、ヒト全遺伝子を対象とした創薬ターゲット遺伝子探索の実現につなげる。

**開発計画:** 遺伝子ネットワーク推定・タンパク質ネットワーク推定・ゲノム多型解析技術等の研究、ならびにデータ同化技術の開発を進め、開発した解析法やデータ同化技術を次世代スパコンに実装し、データ解析とシミュレーションを融合した技術を開発する。



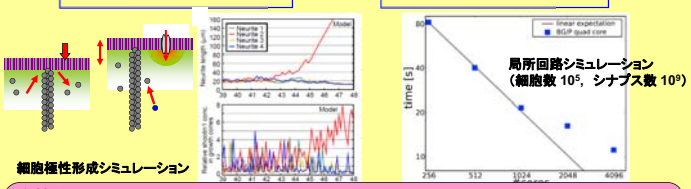

**進捗状況:** 分子のネットワークを地図として抽出するための方式として、大規模遺伝子ネットワーク探索及びタンパク質構造に基づく相互作用予測の研究を行い、新たな技術開発の成功と新たな並列プログラムの開発により、これらのプログラムが1024を超えるcoreで稼動するようになった。

## 脳神経系研究開発

**開発目標:** 脳全体レベルでの入出力関係を再現できるモデルとシミュレーションソフトウェアを開発、次世代スーパーコンピュータの性能を引き出して、脳の柔軟な環境適応能力である脳の発達と学習の再現を目指す。

**開発計画:** 神経細胞と局所回路のシミュレータ、および、脳のシミュレータとして昆虫の匂い情報処理回路モデル、ほ乳類の網膜モデルとを開発、次世代スーパーコンピュータに適したプログラムとすることで、昆虫の嗅覚系情報処理では実時間で、ほ乳類の視覚系情報処理は60倍の時間で計算可能とする。

神経マルチフィジクスシミュレータ　　局所回路シミュレータNEST

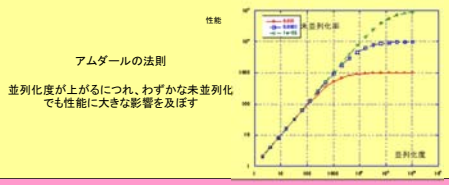



**進捗状況:** 本チームは2008年10月に発足、昨年度は神経マルチフィジックスシミュレータでは神経細胞の極性形成について、分子レベルからのシミュレーションができた。また、局所回路シミュレータNESTでは10^5個の神経細胞、10^9個のシナプスからなる大規模局所回路のシミュレーションができた。

## 生命体基盤ソフトウェア開発・高度化

**開発目標:** プロジェクトで開発される各種ソフトウェアの高性能化、特に次世代スーパーコンピュータ向けの最適化を行い、次世代スーパーコンピュータの生命科学への応用を促進すると同時にグランドチャレンジを実現する。

**開発計画:** コアソフトウェア・基盤ライブラリの開発と他チームアプリケーションの調査・高速化に向けた検討を進め、次世代スーパーコンピュータ実機でのアプリケーションの高速化、チューニングを行う。



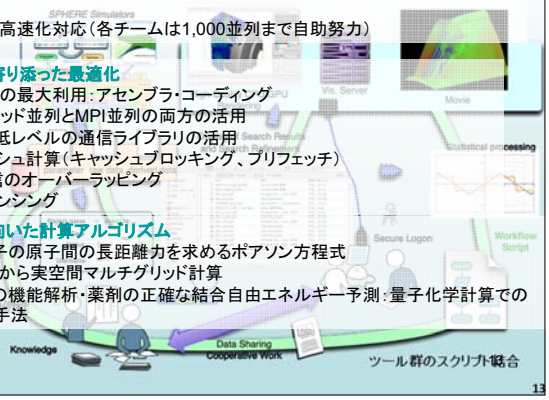

**進捗状況:** 本チームは2007年11月に発足、これまでに分子動力学計算ソフトウェアの解析・開発、量子化学計算ソフトウェアの高性能化についての調査、基盤ライブラリ・可視化ソフトウェアの調査・開発を進めた。

## 生命体基盤ソフトウェア開発・高度化チーム（HPCチーム）の詳細

役割
- 分子動力学や量子力学用ソフトの高速コアライブラリーの開発
- 並列ミドルウェア・超大規模データ処理可視化システム開発（領域型解法、粒子ベースの解法用。ライフサイエンス分野を主な対象としているが、汎用的に使える設計）
- 高並列化・高速化対応（各チームは1,000並列まで自助努力）

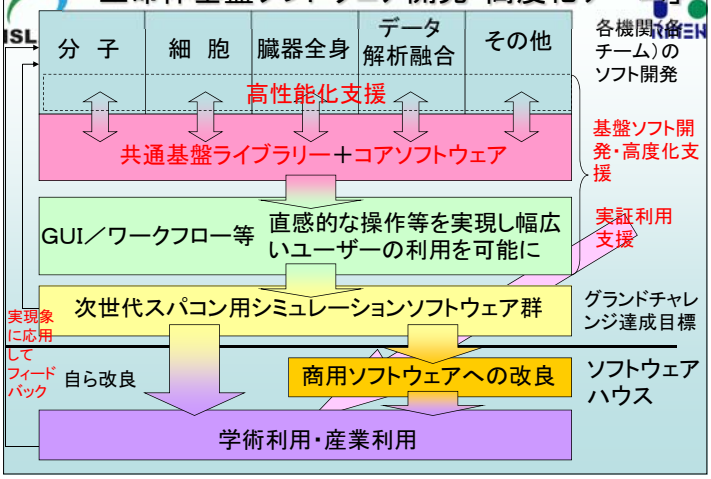
ハードウェアに寄り添った最適化
- SIMD機構の最大利用：アセンブラ・コーディング
- マルチスレッド並列とMPI並列の両方の活用
- MPIよりも低レベルの通信ライブラリの活用
- オンキャッシュ計算（キャッシュブロッキング、プリフェッチ）
- 計算と通信のオーバーラッピング
- ロードバランシング

ハードウェアに向いた計算アルゴリズム
- 生体高分子の原子間の長距離力を求めるポアソン方程式
  → FFTから実空間マルチグリッド計算
- 酵素反応の機能解析・薬剤の正確な結合自由エネルギー予測：量子化学計算でのオーダーN手法



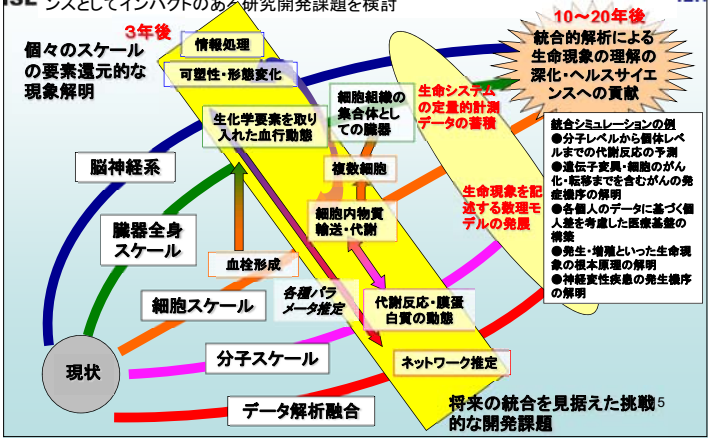



## 「生命体基盤ソフトウェア開発・高度化チーム」



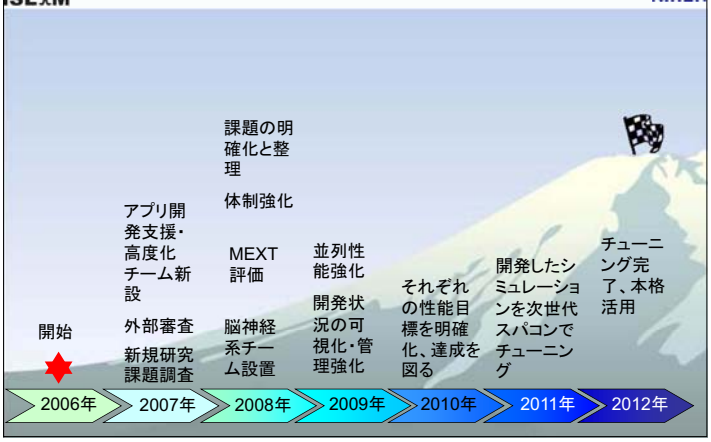

## チーム目標と共通目標（モデル化を目指す現象）

「生命現象の統合的解明」に向け、スケールを超えた連携を視野に入れつつサイエンスとしてインパクトのある研究開発課題を検討






## これまでの経緯と今後の予定




## 研究開発での二面性

- 次世代スパコン開発としての側面
  - 次世代スパコンで初めて可能になる革新的計算
  - 次世代スパコンの性能をフルに引き出すソフトを利用者に広く提供
- 生命科学としての側面
  - 生命現象の理解への貢献
  - 医療・医薬・ライフケアへの貢献



# 2. ライフサイエンス分野における計算科学研究開発の世界の現状

# 1）世界的な研究の中心は

●三極：アメリカと欧州（＋豪州、ニュージーランド）、日本

アメリカ：DoE、NIH、DoD、民間企業（ベンチャー、製薬企業）が主、DoEが基礎科学（生物学＋計算科学）、NIHが医療を担当するため、互いの協力は希薄。計算科学と医療をつなぐ動きがない。

欧州：EUがスポンサーとなり、IT技術を医療に生かす取り組みを推進。一例：VPH(Virtual Physiological Human)、個別患者の治療が目標。大学・医療機関・製薬企業・医療機器メーカーが参加、マッチングファンド、5年で数百億円規模。脳科学ではスイスがBlueBrainプロジェクトを推進。世界をリード。

SciDAC Scientific Discovery through Advanced Computing

Virtual Physiological Human network of excellence

Blue Brain Project



# 2）海外での研究開発の今後の方向

- 世界的に取り組まれているのは
  - タンパクの構造や変化などの解析
  - 大量の実験データに基づく多次元因子解析など


http://simbios.stanford.edu/

- 現在の研究開発でホットな話題はマルチスケール
  NIHもEUでも二つ以上のスケールをつなぐ研究開発にファンディング
  日本はグランドチャレンジとして既にマルチスケールに挑戦
  ・方程式に忠実にモデル化し、スケールを超える
  ・膨大な計算量は次世代スパコンで解消

欧米では医療機関で使うことを前提に簡略化したモデル化。
（脳科学では例外的にスパコンを指向）

戦略分野化で、日本が世界に貢献

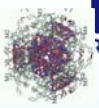

http://ec.europa.eu/information_society/newsroom/cf/itemdetail.cfm?item_id=3956


http://www.nibib.nih.gov/Research/ProgramAreas/MathModeling



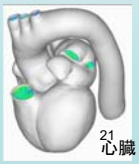
# 3）国際比較

- 分子スケール
  - アメリカは超並列計算で先行
  - 粗視化モデルによる計算、全電子を入れたタンパク質の量子化学計算では日本がリード
- 細胞スケール
  - システムバイオロジー（E-cell等）は日本発
  - 細胞群から臓器へ、分子から細胞へのアプローチはどこもこれから
- 臓器・全身スケール
  - 国際的な研究開発体制が構築されつつある
  - 日本は心臓モデル、健常者からの詳細人体モデルでリード
- バイオインフォマティクス
  - 個人の多様性と医療をつないでゆくところでは日本は優位
  - 次世代超高速シーケンサー等による巨大データに直面、世界的な課題
- 脳科学
  - スーパーコンピュータによるシミュレーションではBlueBrainなどが先行
  - しかし、計算論的神経科学では日本が世界をリード、BlueBrainではこの観点が今は抜けている



全電子計算では世界をリード（ProteinDF）

粗視化モデルによる分子モータ



心臓



# 3）国際比較（続き）

- 数万並列規模の計算能力をもったソフトウェア開発を推進

今後、次世代スパコンに向けた並列性能向上を加速

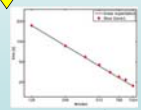
⇒分子・細胞・臓器全身の各スケールと実験データ解析、脳神経系を統合して研究開発

★世界的にも全くない試み

同時に行うことにより、手法やソフトを互いに利用し、開発を加速
⇒次世代スパコンで世界をリードする

チーム間の連携

全電子計算での並列性能（ProteinDF）2500コアまでの並列を確認



脳の局所回路シミュレーションでの並列性能（BlueGeneLで1000並列までの速度向上を確認）

| 受 / 提供 | データ解析融合 | 分子スケール | 細胞スケール | 臓器全身スケール | 脳神経系 |
|---|---|---|---|---|---|
| データ解析融合 | 肺ガンと薬 | ターゲットタンパクの特定 | 生体分子ネットワークモデル | 画像処理パラメータ推定 | パラメータ推定(モデル化) |
| 分子スケール | タンパク-リガンド | 代謝・酵 | 酵での輸送モデル | 血栓形成 | 神経可塑性・変性・イオンチャネル |
| 細胞スケール | 生体分子ネットワークデータ | 代謝反応・酵での物質輸送データ | 肝細胞 | 複数細胞から臓器へ・血栓形成 | 細胞プラットフォーム供給 |
| 臓器全身スケール | 医療画像データ | データ提供・ドラッグデリバリ | 血流・分泌物輸送 | 循環器系 | 全身モデル |
| 脳神経系 | 神経細胞データ・高所回路データ | 可塑性や神経変性に関するデータ | 成長・変形モデル | 運動制御 臓器制御 | 細胞から高次機能までの一貫 |



# 3. ソフトウェア開発状況



# 現在の開発状況

| | 項番 | アプリケーション名 | 略称 | 開発責任者 |
|---|---|---|---|---|
| 分子 | A1-1 | マルチコピー・マルチスケール分子シミュレーション法開発の基盤となるクラスライブラリ | Platypus-MM/CG | 木寺 |
| | A1-2 | レプリカ交換分子動力学計算インターフェイス | Platypus-REM | 村田 |
| | A2-1 | 全原子分子動力学計算 | | |
| | A3-1 | 粗視化モデル計算 | | |
| | A4-1 | 密度汎関数法に基づくタンパク質全電 | | |
| | A5-1 | ハ | | |
| | A6-1 | 量子 | | |
| | A6-2 | 量子 | | |
| | A6-3 | 粗視 | | |
| 細胞 | B1-1 | 細胞 | | |
| 臓器全身 | C1-1 | 全身ログ | | |
| | C1-2 | 微小（復 | | |
| | C1-3 | 重粒子線治療シミュレーション | | |
| | C2-1 | 低侵襲治療シミュレーション（ポケット超音波伝播プログラム） | | |
| | C3-1 | 肺呼吸・肺循環シミュレーション | | |
| | C6-1 | マルチスケール・マルチフィジックス心 | | |

| | 現Phase | 到達並列数 | 進み（月） | ハイブリッド並列化 | F)コンパイラ対応（RIOC） | F)コンパイラ対応（FX1） | 実行時間の測定 | 演算量の測定 | 到達並列数 |
|---|---|---|---|---|---|---|---|---|---|
| | | 2010年4月末 | | | | | | | 2010年6月14日 |
| | B-1 | 4000 | -7 | × | ○ | ○ | × | × | 8000 |
| | B-1 | 1024 | -5 | × | ○ | × | × | × | 1024 |
| | | | | | | | | | 2048 |
| | | | | | | | | | 8000 |
| | | | | | | | | | 8000 |
| | | | | | | | | | 128 |
| | | | | | | | | | 8000 |
| | | | | | | | | | 2000 |
| | | | | | | | | | 128 |
| | B-1 | 1024 | -5 | × | × | × | × | × | 1024 |
| | B-2 | 8000 | | | | | | | 8000 |
| | B-0 | | | | | | | | 512 |
| | B-1 | | | | | | | | 1024 |
| | B-2 | | | | | | | | 8000 |
| | B-0 | | | | | | | | 128 |
| | B-2 | 8000 | 4 | × | ○ | ○ | × | × | 8000 |

**管理：開発フェーズを4つに分類**
Phase Ⅰ：プロトタイプ開発と並列化のテスト（想定2006年～2008年）
Phase Ⅱ：大規模並列化(想定2009年～2010年)
Phase Ⅲ：超大規模並列＋実機チューニング(想定2011年)
Phase Ⅳ：次世代スパコン上での本格的な実行とPR普及(想定2012年)
各I-IIIのフェーズの中に、さらに3つのマイルストーンを設定、10段階に

開発ソフトウェア：全34本
18本：1024並列以上、**10本：8000並列以上**
可視化ソフトも100GPGPU並列

3ヶ月毎にアップデート、問題があれば機動的に対処

## 開発ソフトの優先順位とリソースの有効活用

**背景**

- 稼働初期（部分稼動～本格稼動の間）にはユーザーが使えるマシン時間は少ない
- 高度化チームが同時に取り組めるソフトも限られる

・全ソフトの開発進捗をうまくコントロール、フェーズをずらして、リソースを有効利用できるように図る。
・開発中の**アプリに優先順位**をつける。
・稼働後**短期間に成果を計画的かつ波状的に**出してゆく



## それぞれの成果アピール時期

第一走者

第二走者

第三走者

cppmd（分子動力学）
ProteinDF
CafeMol(ヒストンDNA)
UTHeart
SPH3D
ParaHaplo
NEST(神経回路網)

多剤排出トランスポータ
MARBLE
Platypus
CafeMol
RICS:肝細胞
HIFU　ZZ-DOSE重イオン粒子
SiGN　SSM　L1GNE
LiSDAS
MEGADOCK
CMDN:大脳皮質局所回路

神経系統合シミュレータ
細胞から臓器へ
分子から細胞へ

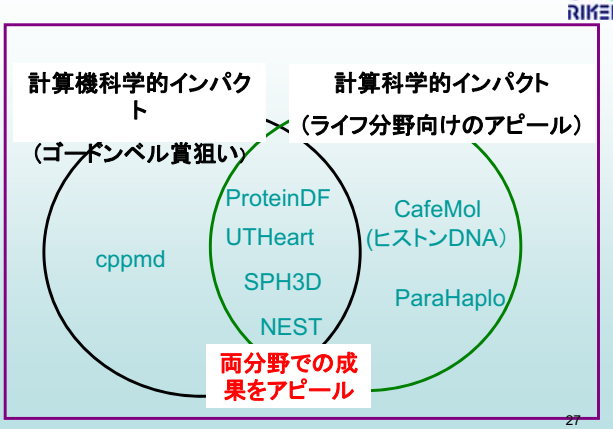
完成後すぐに成果をアピール

完成後、1年以内に成果をアピール

中長期目標
スケールを越えた統合的理解に向けた取り組み



## 第一走者の分類






## 大規模並列用MDコアプログラム:cppmd

開発責任者：泰地　真弘人（理化学研究所）

**世界最速の分子動力学計算でGordonBell賞を目指す**

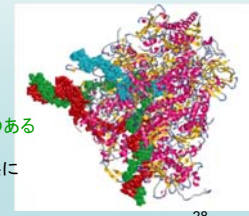
― タンパク質の機能を高速で長時間シミュレーションするための計算技術 ―

**ソフトウェアの特徴**

○分子動力学計算ソフトウェア：10万コア以上の超並列化（100万-1000万原子系で）
10Petaflopsで1億原子級のシミュレーションを実現し、実効性能で最速を目指す

○生体分子の運動の長時間シミュレーション
100万原子系のサブミリ秒シミュレーションを目指す
（対象例）細胞膜構造、核膜輸送、ウィルスなど
【従来の100倍以上の時間スケール】

○これまで不可能だった時間スケールで、生理的に意味のあるタンパク質の大きな構造変化を捉えることにより
生命システムの原子レベルでの構成原理を理解すると共に
創薬に必要なレベルの計算精度を実現する



大規模分子動力学シミュレーション



## 密度汎関数法に基づく タンパク質全電子波動関数計算（ProteinDF）

開発責任者：佐藤　文俊　（東京大学）

**世界最大のタンパク質全電子計算を目指す**

― タンパク質の電子レベルでの反応を正確に解明するための計算科学技術 ―

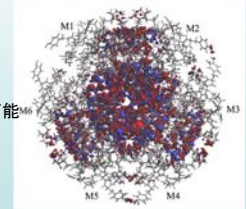
**ソフトウェアの特徴**

○密度汎関数法による全電子波動関数計算ソフト
タンパク質の励起状態を1万並列超で計算

○巨大なタンパク質の励起状態や電子移動の計算が可能
（1千残基、10万軌道、励起状態計算）
→ Protein Data Bankの99%以上が計算可能に

○ミネラルを含むタンパク質（酵素やヘムタンパク質）の働きを解明するのに有効

○電子が主役の高速反応が詳細に解明可能になり、生体模倣光電子素子、人工光合成などへの応用が期待される




（成果の例）
インスリンの凝集に伴う電子の再分布
6量体と単量体の全電子分布の差を描いたもの
10PetaではCytochrome P450などがターゲット



## マルチスケール・マルチフィジックス 心臓シミュレーション:UTHeart

開発責任者：久田　俊明　（東京大学）

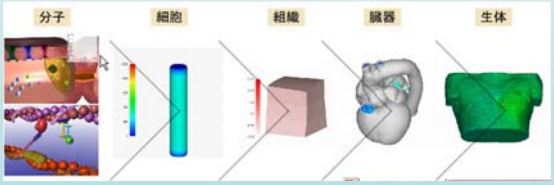
**世界最大規模の 心筋細胞レベルからの心臓全体シミュレーションを目指す**

― 世界でも前例のないバーチャル心臓を世界最大規模でコンピュータ上に再現 ―

**ソフトウェアの特徴**

○分子レベルの電気化学、力学現象を再現した、心筋細胞のモデルを使って
心臓を組み上げ、心臓の動きと血液の拍動を再現するシミュレーション・ソフト

○各種心臓病(不整脈や拡張型心筋症など）の基になっている因子を細胞内で起きる分子レベルの挙動から検討することが可能。医療・創薬への応用を検討。



## 全身ボクセルシミュレーション（ボクセル構造流体連成解析プログラム：SPH3D）

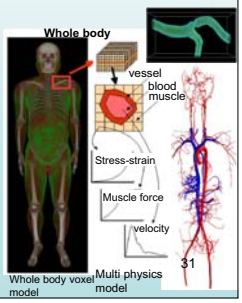
開発責任者：高木　周　（理化学研究所）

**世界最大規模の血栓の形成、輸送、梗塞の同時シミュレーションを目指す**

— 医療応用に向け、やわらかな人体の解析・予測に適した計算科学技術 —

**ソフトウェアの特徴**

- ○大きな血管から毛細血管までの赤血球、血小板を入れた計算ができる構造流体連成ソフトウェア
- ○医療データ（CTスキャン,MRI等）から直接シミュレーションを行うことができ、個人毎の診断に即応してシミュレーションが利用可能に
- ○血栓形成・閉塞による心筋梗塞，脳梗塞の予測と薬効評価
- ○動脈瘤・動脈硬化症等の循環器系疾患の予測と治療法の検討
- ○全身血管網シミュレータとの連成による血圧波形（血圧測定）からの循環器系疾患の予測






## 大脳皮質局所神経回路シミュレーター:NEST

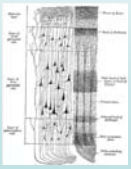
開発責任者：Marcus Diesmann　（理化学研究所）

**世界最大規模の大脳皮質神経回路の活動シミュレーションを目指す**

— 単純な神経細胞モデルから実際の脳の活動が説明できるかを解明 —

**ソフトウェアの特徴**

- ○大脳皮質の100コラムに相当する1千万神経細胞、神経細胞間を接続するシナプス1千億の信号処理を再現・予測（従来の世界トップの100倍の規模）
- ○超高並列プログラム（現時点で32,000並列）次世代スパコン上では数十万並列を達成、**世界最速を目指す**



大脳皮質の情報処理単位
１コラム（1mm³）に
１０万個の神経細胞



## 粗視化モデル計算（CafeMol）

開発責任者：高田　彰二　（京都大学）

**膨大な長さのDNAが核の中に収納され、必要なときに読み出せる謎に挑む**

— 細胞核内でのDNAのヒストンへの巻き付きと、ほどくメカニズムの解明 —

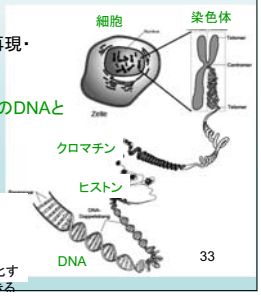
**ソフトウェアの特徴**

- ○規模・時間的に巨大な生体分子の現象をシミュレーションするための粗視化モデル計算
- ○世界最大1千万原子規模・世界最長のミリ秒の現象を再現・予測を目指す
- ○学術的なインパクトを与える、遺伝情報が転写される際のDNAとヒストンのダイナミックスを再現

細胞中で膨大な長さのDNAはヒストンに巻き付き、コンパクトに核に収納されている（クロマチン構造）

遺伝子が発現する際には、適切な場所のDNAをヒストンからほどき、遺伝情報を転写している。

iPS細胞の機能は、クロマチン構造による転写制御機構を初期化することによって達成されるといわれている。そのなぞにも挑戦できる






## ハプロタイプ関連解析に於ける統計検定ソフトウェア（ParaHaplo）

開発責任者：鎌谷　直之（理化学研究所）

**47疾患の原因遺伝子を一挙に発見することを目指す**

— 個人の遺伝情報と疾病、薬物反応性の関連を明らかに —

**ソフトウェアの特徴**

- ○複数の遺伝子や環境要因（身長、体重、喫煙、飲酒など）の組み合わせによって決まる疾病や薬物の効き方、副作用などの関連を網羅的に解析
- ○全ゲノム（30億塩基）の個人ごとの遺伝情報の違いの中から、疾病に関連する遺伝情報を、網羅的に精度よく探し出すことが可能に
- ○バイオバンクジャパンに登録されている約20万人、**47疾患を対象**に解析

**47疾患**

●悪性腫瘍（肺がん、食道がん、胃がん、大腸がん、肝がん、膵がん、胆嚢・胆管がん、前立腺がん、乳がん、子宮がん、卵巣がん、造血器腫瘍）、●バセドウ病、●高脂血症、●糖尿病、●関節リウマチ、●骨粗しょう症、●閉塞性動脈硬化症（ＡＳＯ）、●狭心症、●心筋梗塞、●心筋症、●心不全、●不整脈、●脳血管障害（脳梗塞、脳動脈瘤）、●慢性肝疾患（Ｂ型・Ｃ型慢性肝炎、肝硬変）、●子宮筋腫、●子宮内膜症、●尿路結石、●ネフローゼ症候群、●過敏性症候群（薬疹）、●ケロイド、●アトピー性皮膚炎、●花粉症、●ぜんそく、●結核、●間質性肺炎・肺線維症、●肺気腫（ＣＯＰＤ）、●白内障、●緑内障、●熱性けいれん、●てんかん、●歯周病、●筋萎縮性側索硬化症（ＡＬＳ）

SNP遺伝子型と疾患との関連を示すアーミテージ検定のP値の-log P値




# 4. 各ソフトウェアの概要



## アプリのリスト（1）

| | アプリケーション名 | 略称 | 開発責任者 | 言語 | 並列化の方法 |
|---|---|---|---|---|---|
| 分子 | 密度汎関数法に基づくタンパク質全電子波動関数計算 | ProteinDF | 佐藤(東大) | C++ | MPI, OpenMP |
| | 粗視化モデル計算 | CafeMol | 高田(京大) | Fortran90 | MPI, OpenMP |
| | 全原子分子動力学計算 | MARBLE | 池口(横浜市大) | C | MPI |
| | マルチコピー・マルチスケール分子シミュレーション法開発の基盤となるクラスライブラリ | Platypus-MM/CG | 木寺(横浜市大・理研) | C++ | MPI |
| | ハイブリッドQM/MM反応自由エネルギー計算 | Platypus-QM/MM-FE | 林(京大) | FORTRAN77 | GAMESSのsocksライブラリ |
| | レプリカ交換分子動力学計算インターフェイス | Platypus-REIN | 杉田(理研) | Fortrna90 | MPI |
| | 粗視化モデル計算/分子動力学計算 | Platypus-CGM/MM | 中村(阪大) | C++, Fortrna90 | MPI |
| | 量子化学計算/分子動力学計算 | Platypus-QM/MM | 中村(阪大) | FORTRAN77, Fortran90 | MPI |
| | 量子化学計算 | Platypus-QM | 中村(阪大) | FORTRAN77, Fortran90 | MPI |
| 細胞 | 細胞シミュレーションプラットフォーム | RICS | 横田(理研) | Fortran, C, C++ | Sphere(MPI) |
| 臓器全身 | 全身ボクセルシミュレーション(ボクセル構造流体連成解析プログラム) | SPH3D | 高木(東大・理研) | Fortran, C, C++ | Sphere(MPI) |
| | マルチスケール・マルチフィジックス心臓シミュレーション | UTHeart | 久田(東大) | Fortran | MPI&OpenMP |
| | 低侵襲治療シミュレーション(ボクセル超音波伝播プログラム) | HIFU | 高木・松本(東大) | Fortran90 | MPI&OpenMP |
| | 微小循環シミュレータ(埋め込み境界法による微小循環プログラム) | ZZ-RBC | 高木(東大・理研) | Fortran90 | Sphere(MPI) |
| | 重粒子線治療シミュレーション | ZZ-DOSE | 高木(東大・理研) | Fortran90 | MPI |
| | 肺呼吸・肺循環シミュレーション | ZZ-LUNG | 和田(阪大) | Fortran, C, C++ | Sphere(MPI) |

## アプリのリスト（2）

| | | | | | |
|---|---|---|---|---|---|
| データ解析融合 | ハプロタイプ関連解析に於ける統計検定を行うためのソフトウェア | ParaHaplo | 鎌谷（理研） | C | MPI, OpenMP |
| | 大規模遺伝子制御ネットワーク推定プログラム | SiGN | 宮野（東大） | C99 | MPI, EP (SGE) |
| | 再帰的正則化法による生体内分子の大規模ネットワーク推定プログラム | L1GN | 宮野（東大） | R | EP (SGE) |
| | 状態空間モデルによる時系列データからの遺伝子ネットワーク推定プログラム | SSM | 宮野（東大） | C, C++, JAVA | MPI, OpenMP |
| | データ解析融合プラットフォームの開発 | SBiP | 宮野（東大） | | プラットフォーム |
| | 網羅的タンパク質ドッキング解析プログラム | MEGADOCK | 秋山（東工大） | C++ | MPI |
| | 生命体データ同化プログラム | LiSDAS | 樋口（統数研） | Fortran90, C, C++ | MPI, OpenMP |
| | 次世代シークエンス解析プログラム | NGS analyzer | 鎌谷（理研） | perl, shell | EP |
| | 拡張RAT法による2SNP組合せの全ゲノム関連解析ソフトウェア | ExRAT | 鎌谷（理研） | C++ | MPI |
| 脳神経 | Neural Simulation Tool | NEST | Diesman（理研） | | |
| | Cortical Microcircuit Developed on NEST – Interneuron | CMDN | 深井（理研） | SLI, C++ | MPI |
| | 神経細胞形態シミュレーションキット | NeuroMorphoKit | 石井（京大） | MATLAB, C/C++ | 逐次 |
| | 視覚系シミュレーションのための共有プラットフォーム | VSM | 臼井（理研） | | プラットフォーム |
| | 昆虫嗅覚系全脳シミュレータ | IO3S3M | 神崎（東大） | | 実験系アプリ |
| 基盤 | 大規模並列用MDコアプログラム | cppmd | 泰地（理研） | C++ | MPI |
| | 大規模仮想化合物ライブラリー | VLSVL | 船津（東大） | | EP |
| | 分散並列大規模データ可視化システム | LSV | 小野（理研） | C++, C | MPI |
| | アプリケーションミドルウェア | SPHERE | 小野（理研） | C++,C | MPI |

## 多剤排出トランスポーターにおける薬剤排出シミュレーション（MARBLE／Platypus／CafeMol ）

研究代表者：木寺 詔紀（理化学研究所）

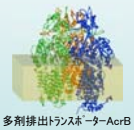
**抗生物質が効かなくなる**
**メカニズムの理論的解明を目指す**

**ソフトウェアの特徴**

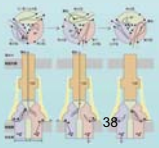
○電子スケール～原子・分子スケール～タンパク質全体スケールの多階層（量子化学・分子動力学・粗視化モデル）の連成計算によって巨大タンパク質の動作機序を明らかにする。

○計算対象：細胞膜にある多剤排出トランスポーター
3つのタンパク質が結合した巨大タンパク質。
各々、取り込み型 → 結合型 → 排出型
と順繰りに構造を変えながら、薬剤化合物を排出すると考えられている。そのメカニズムを探る

○細菌やがん細胞の薬剤耐性化の主要な原因の1つであるメカニズムの解明につながり、薬剤耐性化しにくい抗生物質や、排出を妨げる新薬の設計等への波及が期待できる。



多剤排出トランスポーターAcrB





## 低侵襲治療シミュレーション:HIFU（ボクセル超音波伝播プログラム）

開発責任者：松本 洋一郎 （東京大学）

**超音波治療機器の設計開発・安全性の確認に貢献**

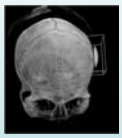
**― 患部だけを焼く集束超音波治療の安全な適用に向けた計算科学技術 ―**

**ソフトウェアの特徴**

○人体内（骨や臓器など音響特性の異なる多媒質体内）の超音波伝搬をシミュレーションするソフトウェア

○医療画像データから開発されたボクセル人体モデルを使ってシミュレーション。HIFU（強力集束超音波）を使った治療機器の超音波の集束状況の高精度な制御をシミュレーション可能

○現在の超音波治療では対応困難な肝臓などの体内深部にある臓器や脳など骨で囲まれた部位に対して，集束超音波で腫瘍を焼灼する医療機器の開発に利用。シミュレーション結果をもとに超音波の照射方法を検討し，手術計画を立案可能



頭蓋骨越しのHIFU照射シミュレーション



## 細胞シミュレータ：RICS

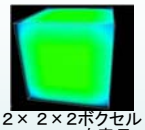
**反応ー拡散ー膜透過の連成計算を100万Voxel空間で実現！**
**($100^3$)**

小規模モデルの計算時間（赤血球）
2× 2×2（＝8）ボクセルを逐次（1CPU）で計算
1000steps/197秒 Δt=1ms （実時間1秒）

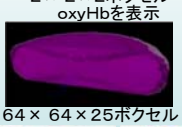
赤血球形状モデル
64× 64×25（＝102400）ボクセルを逐次（1CPU）で計算
512並列：1.36h Δt= 0.05ms
実時間1分：600h 10万並列：3hr

肝細胞詳細モデル
100×100×100（百万）ボクセルを10万並列）で計算
フル代謝モデルを想定 Δt= 0.05ms
実時間1分 ：10万並列：30hr

肝小葉モデル（代謝骨格モデル）
50×50×500（百万）ボクセルを10万並列）で計算
フル代謝モデルを想定 Δt= 0.5ms
実時間10分 ：10万並列：40hr



2× 2×2ボクセル
oxyHbを表示



64× 64×25ボクセル



# 5. ソフトウェアの高性能化





## ソフトウェアの高性能化

- MDコア
  大規模並列化された分子動力学計算コア
- MOコア
  Platypus QM/MMのためのMOの並列化
- 第一走者アプリケーションの高性能化
  - 他チームのアプリケーションを含めたチューニング
  - 性能評価を完了、結果をフィードバック
  - 今後共同で高性能化を進める
- 高度化チームとしての本番の活動は次世代スーパーコンピュータが稼働を始める今年度以降
  - 低レベル通信API
  - SIMD命令の直接的利用
- 準備状況としては、ほぼ予定通り

## 世界の状況

- 並列化の状況に関しては、アプリケーションによってまちまち。
- MDの場合：

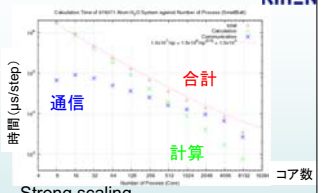
  Desmond (D. E. Shaw Research), NAMD(UIUC), Blue Matter (IBM) などが数千並列以上の並列を実現。

  何れのコードも、高性能計算の専門家が中心に参加して開発を進めている。
- ライフアプリケーションでの$10^4$以上の並列度については、世界的に見てもこれまでの蓄積は非常に少ない。

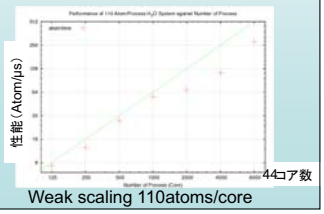


## MDコア:cppmd（1）開発のねらいと状況

- 大規模並列に向けて、様々な並列化手法を盛り込んだ基盤ソフトウェア
- 大規模並列化の「トレーニング」
- 開発時点から次世代スーパーコンピュータを強く意識
- 多機能版
  - 再利用性・手法比較
- 高速版
  - 二体相互作用カットオフ計算の高速化を優先
  - 高並列時の通信を考慮
  - 水 976371原子（110原子/コア）RICC 8000並列での並列化確認
  - その他の機能は多機能版から移植中




Strong scaling
976,371 atoms, smallball, XYZ通信



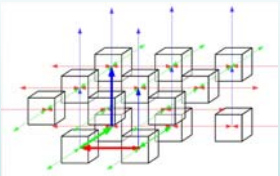

Weak scaling 110atoms/core

## MDコア（2）次世代スーパーコンピュータのネットワークトポロジを考慮した通信パターンの改良

通常の実装

- 全方向直接
- 通信相手が多い
- 競合が激しくなる
- 次世代スーパーコンピュータの特徴を生かせない

今回の実装



- x,y,z軸に逐次通信
- 通信相手が少ない
- 次世代スーパーコンピュータのネットワークトポロジ（トーラス構造）に一致

**8000コアでの性能向上を実現**



## MDコア（3）次世代スパコンでの展望

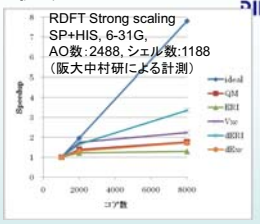
- 二体相互作用のカットオフ計算
  B/Fの違いも考慮すると
  90原子/ノード程度までスケール（11原子/コア）
- 次世代スパコン全体（80,000ノード）で
  7,200,000原子
- $2\times10^9$ step / day（効率10%= 1Pflops）
- 2fs/stepで4μsec/day
- 今後の課題
  他の計算の負荷の見つもり
  単体性能チューニング
  - ✓ Integer cache missの改善（FX1での実行結果から）
  - ✓ OpenMP・SIMDの活用
  - ✓ 命令実行密度(CPI)の向上

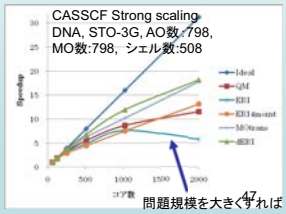


## MOコア 開発のねらいと状況

- Platypus QM/MMのための並列分子軌道法コアソフトウェア
- HF/RHF/CASSCF/CASCI
- ベクトル化（当初）、SIMD化についても調査
- 基本部分の開発を完了し、阪大中村研に引き渡し
- DFT/CASDFTについては阪大中村研で開発
- CASSCFのMO変換の性能向上などに特徴
- 今後、さらに並列化率の向上に協力。

  典型的な問題に対し、20,000コア以上までのスケーリング実現が目標
- その他、O(N)分子軌道法(NTChem, 有限要素基底)の高速化についても一部支援




問題規模を大きくすればさらに改善する見込み



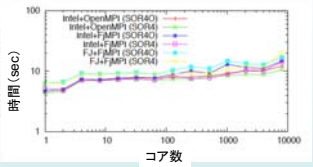
## 構造流体連成プログラムSPH3D（1）性能評価

臓器全身スケールチームで開発中のコード

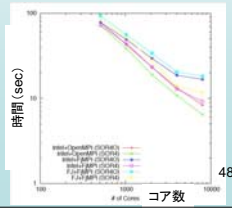
- 1メッシュ当りの演算数
  - A=7144+827+18$N_i$
  - $N_i$はSORのiteration回数
- Weak/Strong scaling共に良好

| | | |
|---|---|---|
| solphi | 827flop | 2.8GFLOPS@RICC(24% of peak) |
| solu | 7144flop | 1.5GFLOPS@RICC(13% of peak) |
| Solp_sor4 | 18flop/iteration | 0.71GFLOPS@RICC (6% of peak) |

Weak Scaling ($100^3$ grids/core)

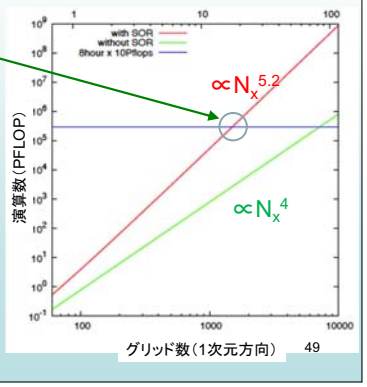



Strong Scaling ($960^3$ grids/core)



# 構造流体連成プログラムSPH3D（2）次世代スーパーコンピュータに向けた展望

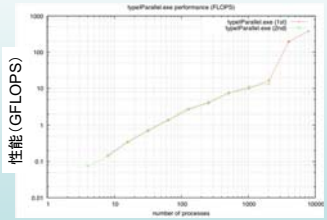
- 次世代スパコン(10Pflopsマシン)上でのペタスケール演算モデル
  - 現状コードでは$1500^3$ボクセルが限界
  - このときSORの反復数は48000回
    - 並列FFTベースソルバの1000倍遅い
  - 1Core当りのボクセル数が$17^3$：通信ボトルネック
  - Poisson solverの改良が必須
- 課題
  - Multigrid Poisson Solverの導入（MDコア等で開発）
  - Hybrid 並列化






# 全ゲノム関連解析ParaHaplo 性能評価と展望

- 基本的には、Embarrassingly Parallel
- RICC 8000コアでスケーリングを確認
- ターゲット規模：1万人集団×100万SNP 次世代スパコンで15時間
- ハードウェアカウンタでは383GFLOPS
  コア当たり0.05GFLOPS：整数演算中心
  → 整数演算の浮動小数点演算への置き換えを検討
- 連鎖不平衡ブロックあたりのハプロタイプデータ数のアンバランスが、ロードバランスに直結
  → 配分の見直しによるロードバランスの改善を検討






# NESTシミュレータ

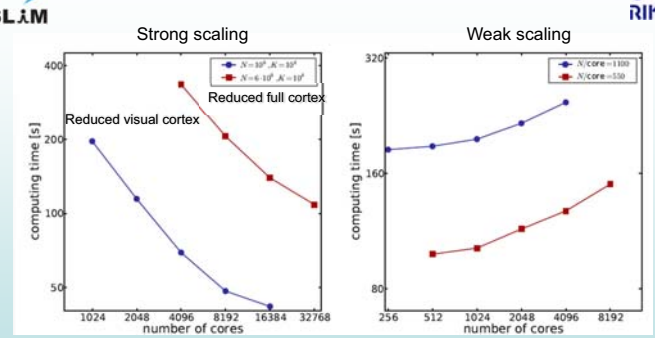
*http://www.nest-initiative.org*

- 大規模神経回路モデルの分散シミュレーション
  - 現時点で世界最高水準の集積度を可能としている
- *O*(100) プロセッサまで超線形、*O*(1000) プロセッサまで線形の性能向上
- CPU間のメッセージパッシングとマルチコアCPU内のマルチスレッディングのハイブリッドアーキテクチャ
- 連続値でのスパイクタイミングを用いた精度良い積分
- ユーザフレンドリーなPythonインターフェス***PyNEST***
- 多種のニューロンモデルとスパイクタイミング依存可塑性 (STDP) を含む多種のシナプスモデルを実装可能
- 短期可塑性 (STP)も扱うことができる

Eppler, Helias, Muller, Diesmann, and Gewaltig (2008)
Frontiers in Neuroinformatics 2:12. doi:10.3389/neuro.11.012.2008



# 大規模シミュレーションのテスト



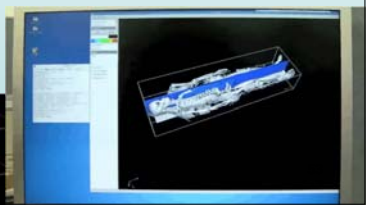

- 現在のシミュレーションは$10^6$ - $10^7$ ニューロンからなるモデルで実施中

Kunkel S, Potjans T, Morrison A, Plesser HE, Diesmann M



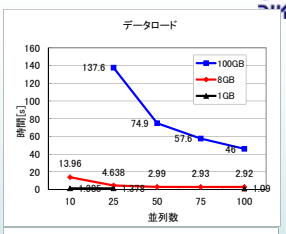
# 大規模可視化ソフトウェア（1）

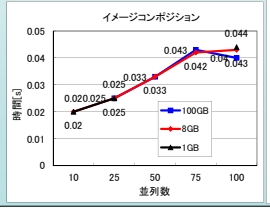
- 大規模並列化され、大規模データに対応した可視化ソフトウェア
- リモート可視化にも対応
- 人体ボクセルデータでのテスト
  （最高解像度：0.2mm pitch, 27.5 GVoxel, 107GB）
- RICC GPGPUクラスタ(upc)を使用
  - Nehalem+ TESLA C1060 100台

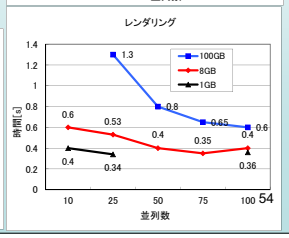


# 大規模可視化ソフトウェア（2）性能測定（人体ボクセル）

- データ分割によりデータロードは1〜3秒程度まで高速化を実現
- レンダリングは0.4秒程度
- イメージ重畳は一桁程短い時間
- 5FPS(0.2[sec/frame])の実現にはデータロードとレンダリングのチューニングが必要
  - データロード：ステージング＋高速化
  - レンダリング：ノード内並列化

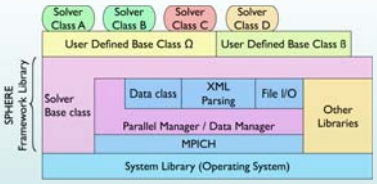


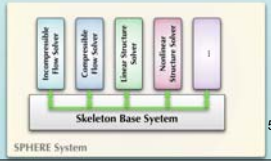



# アプリケーションミドルウェアSphere
## 並列ソフトウェア開発加速のためのプラットフォーム

- 開発者向け機能
  - ラピッドプロトタイピング
    - ライブラリ群の利用
    - ひな形制御構造の利用
    - コード部品の再利用
  - 並列処理機能の利用
  - ソースポータビリティ
  - 混合言語 C++/F90, GPGPU
  - 連成問題のスクリプト記述
- エンドユーザ向け機能
  - 統一されたUIと実行環境
  - XMLによる制御
  - マルチプラットホーム
    - 次世代スーパーコンピュータ, Linux cluster, Windows cluster, BG, SX, Mac, Altix, T2K…
- 現在、臓器全身スケール（SPH3D, HIFU）・細胞スケールチーム(RICS)で利用



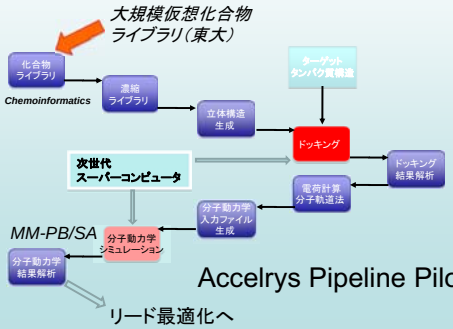

```
XML file :
<SphereConfig SolverType="Incompressible_flow_solver" >
```






# 産業応用に向けた整備

- 大規模な分子シミュレーションを用いた薬剤スクリーニング向けパイプライン




Accelrys Pipeline Pilot上に構築



13機関：東京大学、大阪大学、京都大学、東北大学、北陸先端大学、東海大学、千葉大学、慶応大学、横浜市大、東工大、広島大学、統計数理研究所、理化学研究所
約200名の研究者（ポスドク約80名）



# まとめ

- 開発ソフトウェア：全34本
  - そのうち18本：1024並列以上
  - **10本：8000並列以上**
  - 可視化ソフトも100GPGPU並列
- 今後、並列性能の改善
- 次世代スパコンにフィットした最適化
- 計算時間に合わせた適正な計算規模を設定し、利用計画を立案、フル稼働に備える



# Contributers

Workshop in 2009


Workshop in 2008

科学技術計算分科会 選出
2010 年度 科学技術計算分科会 会合 より

# 原子力研究開発を推進する スーパーコンピュータシステムと 原子力アプリケーション

日本原子力研究開発機構
平山 俊雄

サイエンティフィック・システム研究会
2010 年度科学技術計算分科会会合資料

2010 年 10 月 20 日

# 原子力研究開発を推進するスーパーコンピュータシステムと原子力アプリケーション

平山 俊雄 ・ 清水 大志 ・ 久米 悦雄
日本原子力研究開発機構・システム計算科学センター

[アブストラクト]

原子力機構では、計算科学を活用した原子力の研究開発を加速するために、本年、3 月には、Altix3700 Bx2（13TFlops）に替えて、国内最高性能となるピーク性能 200 TFlops の Linux クラスターシステム（BX900）とピーク性能 12 TFlops の次世代計算機プロトタイプ機（FX1）からなる 2 種類のスーパーコンピュータを導入し、それらの運用を開始した。BX900 は、旺盛な計算需要に応えるために、また FX1 は次世代計算機（京コンピュータ）の利用に向けた原子力アプリケーションのチューニングを目的としている。

BX900 については、その利用開始後 3 日目で 90%を超える利用率を達成し、その後も高い利用状況が続いている。これは、原子力計算科学研究者が数千コアを用いた大規模並列計算コードの開発能力と利用技術力を既に兼ね備えていることを表している。次世代計算機向け原子力コードのチューニングについては、FX1 を用いたプロファイリングデータに基づく性能予測を行い、1PFlops までの良好な並列性能を推定した。また、新旧 3 機種のスーパーコンピュータの基本性能を比較した結果についても言及する。

[キーワード]

原子力計算科学、次世代計算機、ペタフロップス計算、BX900、FX1、ハイブリッド並列

## 1. はじめに

計算科学技術は「理論」及び「実験」と並ぶ第 3 の研究手法として、21 世紀の先端的研究のフロンティアを切り開くための重要な基盤技術となっている。特に、原子力のような巨大技術においては、安全面や時間・空間の制約により実験が困難な場合が多く、計算科学技術は従来から重要な研究手段となっている。すなわち、計算科学技術は、原子力研究開発の効率化、原子力施設の安全評価、国際競争力強化のカギを握る技術となっている。原子力委員会においても、その原子力政策大綱の中で「シミュレーション技術の高度化による大規模な技術システム開発の効率化」を求めている。

原子力関連二法人の統合と独立行政法人化を経て、旧二法人が所有していたスーパーコンピュータの資源について、整理統合と合理化を果たし、膨大な計算需要に応えるための新たな設備を導入すると共に、京コンピュータの利用に向けた準備を整えた。

## 2. 新スーパーコンピュータの性能比較

BX900 の Linpack 性能は 191.40 TFlops、実行効率 95.66 %であり、2010 年 6 月時点の TOP500 によれば世界 22 位、実行効率世界 2 位であった。FX1 については、11.66 TFlops、実行効率 90.37 %である。

第 1 図に、新旧 3 機種の HPC Challenge Benchmark の性能を示す。ここで、Altix3700Bx2(Itanium2)は更新前の機種で、ピーク性能 13 TFlops である。Altix3700Bx2 に比較して、BX900 は演算性能、通信性能ともにコアあたりのピーク性能差（～2 倍）だけの改善がみられる。ただし、プロセス間通信のバンド幅および遅延については、ソケットあたりのコア数の増加やインターコネクトの構造の違いのため、単純に比較することができない。

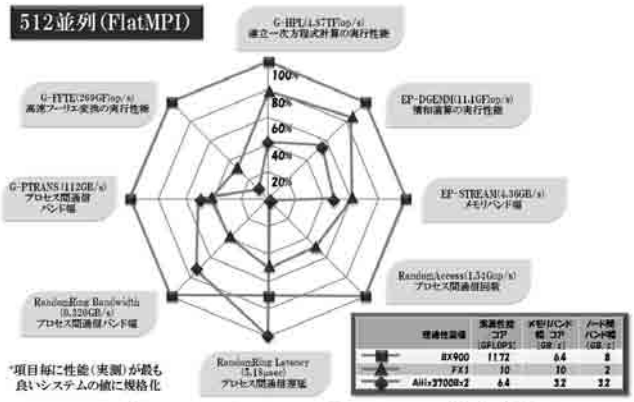



図1：新スーパーコンピュータの基本性能
HPC Challenge Benchmarkの結果

FX1 については、実効メモリバンド幅及びプ

ロセス間通信性能が BX900 と比較して低位にある。特に、実効メモリバンド幅については、BX900 がカタログ性能の 70%程度出ているのに対し、FX1 では 30%以下となっている。この問題は、メモリコントローラチップが CPU の外にあることに起因しているようだが、京コンピュータでは CPU 内蔵となっており、本問題は解決しているとのことである。

## 3. 京コンピュータの利用に向けたチューニングと性能予測

将来において膨大な計算需要を擁している核融合コードについて、 PFlops 級の計算を想定した並列性能および実行性能を BX900 および FX1 を用いて評価した。

BX900 の 16,384 コアを用いて並列性能を評価した結果、全てのコアに対して MPI プロセスを割り当てる FlatMPI では MPI プロセス間通信量の増大により、全体処理性能は大幅に低下する。したがって、大規模並列計算ではスレッド並列を併用する Hybrid 並列化を用いることで、MPI プロセス数を削減することが必須となる。なお、富士通 Fortran の自動並列化は核融合コードについてはかなりの部分で有効であった。また、メモリ使用量を軽減する上でも、Hybrid 並列化は有効である。

第 2 図は、京コンピュータ 4096 並列（～500TFlops）上での性能予測結果である。FX1 の 256CPU（1024 コア）を用いて性能予測をするため、4096 並列の 1 プロセスあたりの問題規模を保ったまま、weak scaling を仮定して 4096 プロセス並列から 256 プロセスにスケールダウンをする。次に、FX1 の 256CPU を用いて取得したプロファイリング情報に対して、FX1 から京コンピュータへの性能変換表を用いて演算時間を推定した。通信時間については、当該コードの通信の種類、データ長、通信回数を調査し、インターコネクトのトポロジー（仮定）をもとに机上計算により推定している。

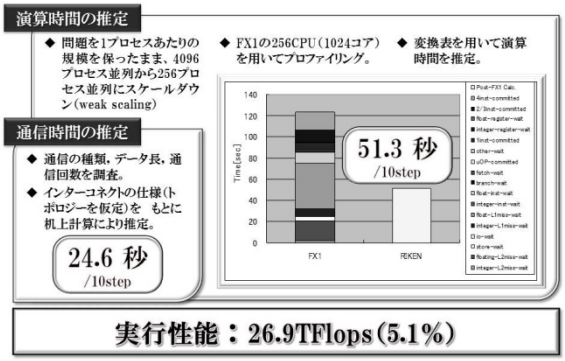
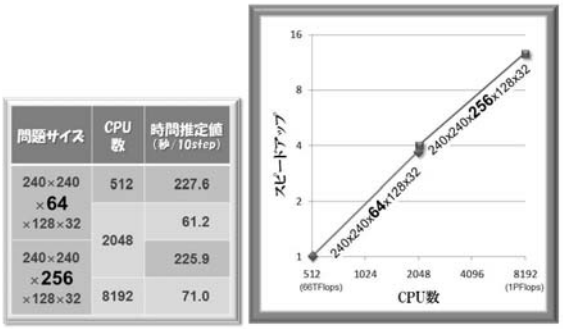



図2：性能予測；FX1から京へ
（4096並列、～500TFlops）

演算性能については、京コンピュータにおいてプロセスあたり 2.4 倍程度の高速化が期待できる。京コンピュータのコア数が、4 コア（FX1）から 8 コア（京）と 2 倍になっていることから、演算性能は十分にスケールしているといえる。しかし、1 コアあたりの性能はすでに飽和しつつあることに留意する必要がある。したがって、京コンピュータの利用にあたっては、コア数が増えた時のコードの並列化性能が特に重要になる。

第 3 図は、京コンピュータの 1PFlops について、上記推定手法に基づいた 2 段階の Strong Scaling である。1PFlops においても良好な並列化性能が期待できることがわかる。



| 問題サイズ | CPU数 | 時間推定値（秒/10step） |
|---|---|---|
| 240×240×64×128×32 | 512 | 227.6 |
| | 2048 | 61.2 |
| | | 225.9 |
| 240×240×256×128×32 | 8192 | 71.0 |



図3：1PFlopsまでのスケーラビィリティ

## 4. まとめ

日本原子力研究開発機構では、「シミュレーション技術の高度化による大規模な技術システム開発の効率化」を求めた原子力政策大綱に基づき、計算科学的手法を用いた原子力研究開発を推進してきた。その結果、年間の論文数の 20%以上が計算科学的手法を用いた研究に移行している。しかし、数 PFlops に達する膨大な潜在需要に応えるためには、計算機資源の不足は深刻な状況であり、京コンピュータの積極的利用が不可欠となっている。

京コンピュータのプロトタイプ・アーキテクチャの FX1 を用いたコード分析とチューニングは、京コンピュータの利用に向けた準備作業として極めて有効であることを確認した。また、原子力コードについて FX1 を用いたチューニングを行い、1PFlops までの良好な並列性能を確認した。

# 原子力研究開発を推進するスーパーコンピュータシステムと原子力アプリケーション

日本原子力研究開発機構
システム計算科学センター
平山 俊雄・久米 悦雄・清水 大志
平成22年10月20日



## 概　要





## Ⅰ. 原子力研究と計算科学





## 1 原子力機構研究開発拠点




## 2 原子力研究開発4大プロジェクト

**長期的エネルギー安全保障 地球環境問題の解決**

- 核燃料サイクルの確立
  - 高速増殖炉サイクル技術（国家基幹技術）
  - 高レベル放射性廃棄物処分技術
  - 軽水炉サイクル事業支援
- 原子力による水素社会への貢献

**国際競争力のある科学技術を生み出す基盤**

- 核融合研究開発
- 量子ビームテクノロジー

**原子力の安全と平和利用を確保するための活動**

- 安全研究
- 核不拡散技術開発

- 自らの施設の廃止措置 廃棄物の処理・処分
- 産学官との連携　国際協力 人材育成　原子力情報

**共通的科学技術基盤**

- 原子力基礎工学研究、先端基礎研究



## 3 原子力研究における計算科学の役割

◆ 原子力の研究開発において、計算科学技術は理論、実験と並ぶ第3の柱

◆ 研究開発の効率化の原動力
- 原子力分野においては、大規模な施設・設備を用いる実験または長期間を要する実験や、観測が困難な現象について、計算科学技術はその解明や予測を可能にする

◆ 原子力施設の安全評価に必要不可欠
- 計算科学技術は、原子炉材料・核燃料の経年変化、原子力施設の耐震性等の詳細を評価・予測可能にする
- 安全性・健全性評価に最新知見を取り入れるための必須技術

◆ 国際競争力強化のカギを握る技術
- 原子力分野への計算科学技術の応用により、「実験・設計の繰り返しによる開発」から「高精度シミュレーション予測に基づく開発の効率化」へと技術革新を可能にする
- 米DOE，欧EURATOMにおいても、原子炉の設計、安全、材料に係るシミュレーション技術開発等を実施

# 原子力委員会からの要請
－ 計算科学技術を活用した原子力研究開発の必要性 －

◆シミュレーション技術の高度化等による大規模な技術システム開発の効率化
- 原子力政策大綱

◆最高水準の計算機シミュレーション能力を整備・維持・発展

◆開発システムの適合性を早い段階から計算機によるモデリング・シミュレーション技術等を駆使して多面的に評価
- 第5回原子力委員会資料第1号（独立行政法人日本原子力研究開発機構の次期中期目標の策定について（見解））

◆近年急速に進歩してきた情報技術、モデリングとシミュレーション技術を駆使して、性能目標を達成するシステム実現を、設計、製造・建設、運転、廃棄に至るすべてのプロセスについて、研究開発の可能な限り初期の段階から多面的かつ徹底的に検討するフロントローディング型の研究開発活動の導入を検討
- 原子力委員会　研究開発専門部会資料　原子力政策大綱に示している原子力研究開発に関する取組の基本的考え方の評価



# 4 原子力研究分野の紹介

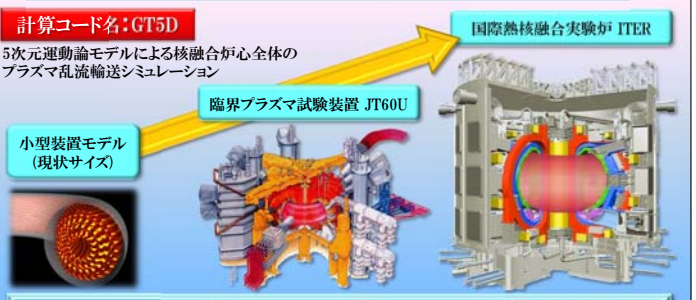
1. 核融合プラズマ乱流シミュレーション
2. 原子力機器内沸騰二相流解析
3. ナトリウム冷却高速炉における非定常流動現象
4. 第一原理計算による材料劣化メカニズム（粒界脆化）の研究
5. その他計算科学研究事例



# 1 核融合プラズマ乱流シミュレーション

計算コード名：GT5D

5次元運動論モデルによる核融合炉心全体のプラズマ乱流輸送シミュレーション




| | 小型装置 | JT60U | ITER |
|---|---|---|---|
| 炉心体積 | $6m^3$ | $60m^3$ | $800m^3$ |
| 格子数 | $5.24 \times 10^9$ | $1.38 \times 10^{11}$ | $1.23 \times 10^{12}$ |
| メモリ量 | 800GB | 14TB | 147TB |
| 計算量（実効性能・日） | 30TFlops・day | 750TFlops・day | 7PFlops・day |



# 2 原子力機器内沸騰二相流解析

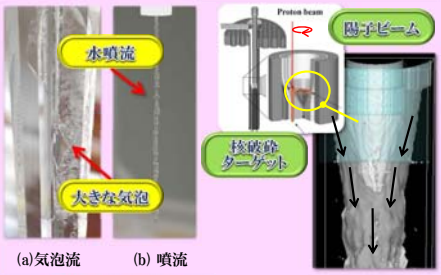
計算コード名：TPFIT

計算の目的　大規模数値解析に基づき大型実験に依存しない原子炉熱設計手法の開発

計算内容

気液二相流挙動を実験式を使用せず、気液界面の形状変化等を直接解析することで、原子炉内の詳細な二相流解析を行う

◆大規模（高額・長時間）な実験結果が不要
◆使用者の経験に依存しない（経験に依存する解析パラメータがない）
◆単一のコードで様々な解析が可能




(a)気泡流　(b) 噴流

二相流解析結果：様々な形態の二相流が解析可能

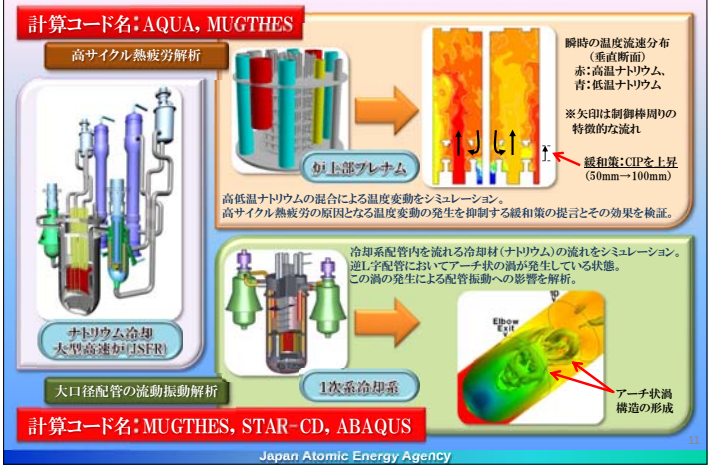
原子炉内二相流解析結果：加速器駆動炉の核破砕ターゲット



# 3 ナトリウム冷却高速炉における非定常流動現象

計算コード名：AQUA, MUGTHES

高サイクル熱疲労解析




高低温ナトリウムの混合による温度変動をシミュレーション。
高サイクル熱疲労の原因となる温度変動の発生を抑制する緩和策の提言とその効果を検証。

大口径配管の流動振動解析

計算コード名：MUGTHES, STAR-CD, ABAQUS



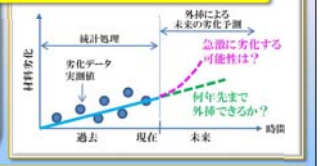
# 4 第一原理計算による材料劣化メカニズム（粒界脆化）の研究

計算コード名：VASP

劣化メカニズムを把握しなければ、科学的・合理的根拠に基づく劣化予測ができない

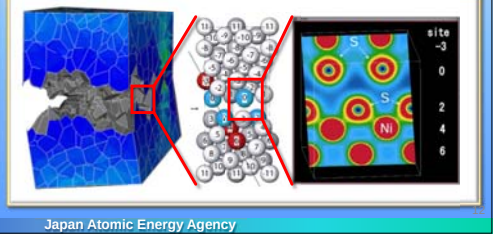
計算の目的、概要

◆原子炉構造材料の劣化メカニズム解明を通して、原子炉の高経年化対策に科学的・合理的根拠を提供すること。
◆本研究では「粒界脆化」という劣化について、原子・電子レベルから、そのメカニズムを明らかにする。

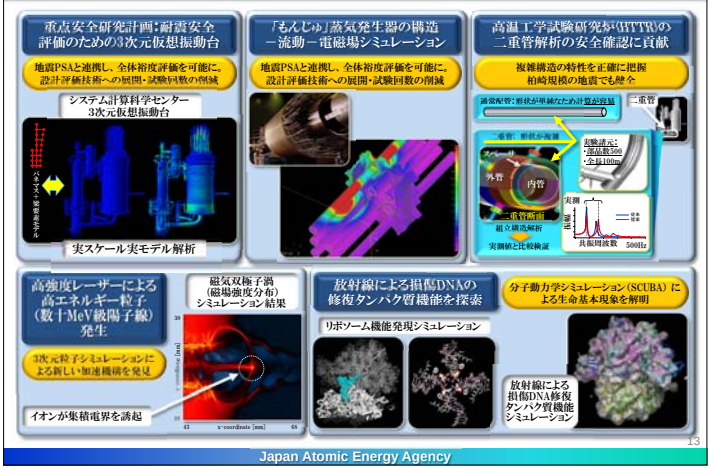



計　算　内　容

◆結晶粒界の原子モデルを作り、第一原理計算によって不純物元素の粒界への集まりやすさ、粒界強度の変化を調べる。

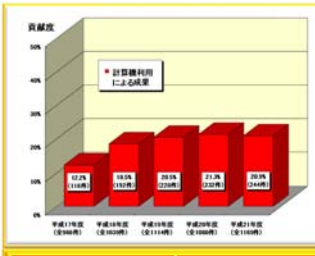


# 5 計算科学研究事例

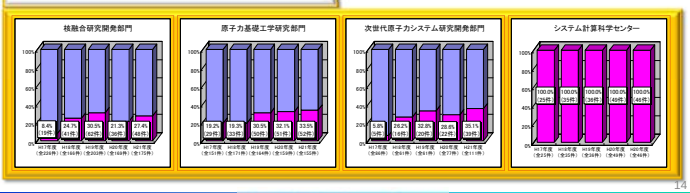





# －原子力機構（研究成果に対する計算科学の貢献）－




- ◆ 原子力機構成果（査読付論文）の 20%以上が計算科学技術的手法による
- ◆ 論文数では、5年間で約2倍に増加
- ◆ 原子力基礎、核融合、次世代部門では部門全体の成果の1/4～1/3 を占める






# 5 欧米における原子力研究開発と計算科学の現状 －米国における原子力計算科学研究の動向－

| 管轄 | 分野 | 機関名* | 順位 |
|---|---|---|---|
| DOE | 原子力 | ORNL | 1 |
| | | LANL | 3 |
| | | LLNL | 8 |
| | | ANL | 9 |
| | | SNL | 10 |
| NSF | 基礎・基盤 | テネシー大 | 4 |
| NASA | 航空宇宙 | ARC | 6 |

アルゴンヌ国立研究所のプレス発表
(2009年12月31日@ニューヨークタイムズ)

**"At Argonne Lab, a Shift From Radioactivity to Supercomputers"**

スーパーコンピュータTOP500（2010年6月）
補足:2位と7位は、National Supercomputing Centre in Shenzhen(中国)
5位は、Forschungszentrum Juelich（独国）

**1. ハードウェア**（スーパーコンピュータ等）
- ●連邦政府による研究・学術機関向け支援（市場45%）
- ●TOP500の50位以内中、18台は国立研究所で運用

**2. ソフトウェア**（シミュレーションとモデリング）
- ●先端科学コンピューター研究*プログラム等による拠点化（ORNL、ANL等エネルギー革新拠点）を創設・運営）
- ●総額366百万ドルの投資発表（122百万ドル/5年・拠点）

*ORNL: Oak Ridge National Laboratory, LANL: Los Alamos National Laboratory, LLNL: Lawrence Livermore National Laboratory, ANL: Argonne National Laboratory, SNL: Sandia National Laboratory, ARC: Ames Research Center

*Advanced Scientific Computing Research



# －欧州（フランス原子力庁）における原子力計算科学の動向－


EUプロジェクト:PRACE
（欧州における先端計算科学研究連携）

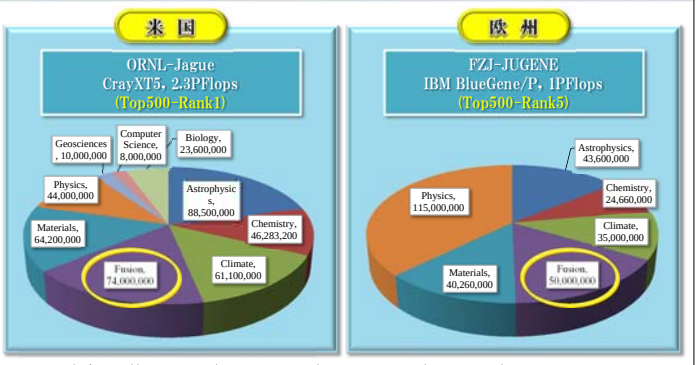
CEAにおける計算科学推進拠点設立
（フランス原子力庁における計算科学の活性化）

**1. ハードウェア**（スーパーコンピュータ等）
- ●目標:向こう5年間で6か所に数ペタ級計算機を導入
- ●投資:400百万ユーロ/5年

**2. ソフトウェア**（シミュレーションとモデリング）
- ●従来:CEA内7拠点で計算科学を推進
- ●新体制:計算科学による原子力研究の加速表明



# －欧米のフラッグシップマシンにおける計算機資源割り当て－




※ORNLデータ: http://www.nccs.gov/leadership-science/project-archive-list/fy09-projects/
※FZJデータ: http://www.prace-project.eu/hpc-access/page-11/

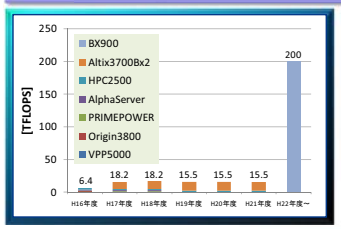
**世界的な動向として核融合はHPC開発を牽引する重要な分野となっている**



# Ⅱ. スーパーコンピュータ整備計画

1. 独立行政法人化ならびに二法人統合によるスーパーコンピュータ資源の整理・統合
2. 計算需要動向とスーパーコンピュータ整備方針
3. 新スーパーコンピュータの構成
4. 新スーパーコンピュータの利用状況
5. 増大する計算需要への対応

## 1 独立行政法人化ならびに二法人統合による スーパーコンピュータ資源の整理・統合

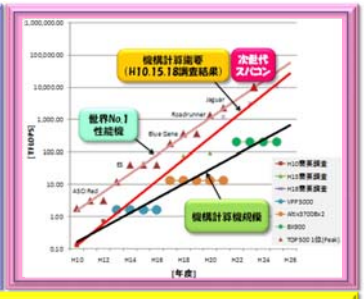




◆原子力機構では、原科研、那珂研、関西研及び大洗研の4地区でスパコンを運用

- 平成16年度末に原科研と那珂研を統合。
- 平成18年度末には法定耐用年数を経過した関西研のシステムを廃止。
- 平成21年度末に原科研と大洗研(旧JNC)を統合、原子力機構における唯一のシステム。
- 統合・合理化計画を完了。



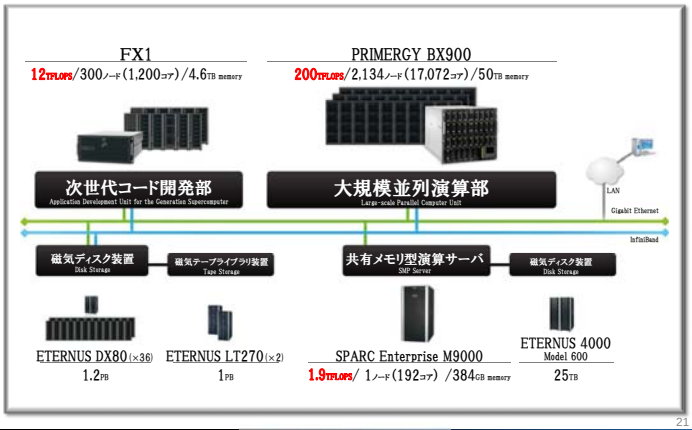
## 2 計算需要動向とスーパーコンピュータ整備方針


◆ 原子力の計算需要は拡大の一途
◆ 計算機資源の枯渇は深刻な状況
◆ スーパーコンピュータ整備方針
① 旧２法人のスーパーコンピュータを統合・合理化し、よりコストパフォーマンスの高い新システムへ更新 ➡ BX900
② 京コンピュータを活用するための環境整備 ➡ FX1



## 3 新スーパーコンピータの構成






## 大規模並列演算部
Large-scale Parallel Computation Unit

大規模並列演算部「PRIMERGY BX900」は、国内最高性能となる200TFLOPSの理論演算性能（ピーク）を有する大規模Linuxクラスタシステムです。本システムは、2,134ノードのサーバブレードから構成され、各ノードにはインテル®Xeon®プロセッサX5570（クアッドコア）を2プロセッサ、主記憶24GBを搭載しています。ノード間は、InfiniBand™ QDRの2ポート接続により片方向8GB/sの帯域を有し、高速なデータ転送を実現しています。本システムでは、主に原子力基礎工学、核融合、量子ビーム、計算科学分野などにおける大規模シミュレーション計算処理を行います。

PRIMERGY BX900

| | |
|---|---|
| ピーク性能 | 200TFLOPS |
| 主記憶容量 | 50TB |
| ノード数 | 2,134ノード |
| プロセッサ数(コア数) | 4,268プロセッサ(17,072コア) |
| プロセッサ名 | Intel®Xeon®Processor X5570 |
| コア数/プロセッサ | 4コア |
| 理論演算性能/コア | 11.72GFLOPS |
| メモリバンド幅/プロセッサ | 25.6GB/s |
| キャッシュ | 8MB |
| プロセッサ数/ノード | 2プロセッサ |
| 理論演算性能/ノード | 93.76GFLOPS |
| 主記憶容量/ノード | 24GB |
| ノード間ネットワーク帯域/ノード間最高通信性能 | 0.54GB/sec |
| ネットワークトポロジ | Fat Tree |
| 通信性能 | 8GB/s(片方向) |
| ノード間最高通信遅延(片方向)/ノード内最高通信性能 | 0.17μsec |




## 次世代コード開発部
Application Development Unit for the Next Generation Supercomputer

次世代コード開発部「FX1」は、12TFLOPSの理論演算性能（ピーク）を有するクラスタシステムです。本システムは、300ノードから構成され、各ノードには富士通クアッドコアプロセッサSPARC64™ VIIを1プロセッサ、主記憶16GBを搭載し、専用チップセットにより40GB/sという高メモリバンド幅を実現しています。また、並列プログラムにおけるプロセス間の同期や総和処理などを高速化する高機能インターコネクトスイッチ（ハードウェア）を有しています。本システムでは、本システムの技術が次世代スーパーコンピュータのベースとなっている特徴を生かし、次世代スーパーコンピュータの利用に向けたアプリケーション開発を行います。

FX1

| | |
|---|---|
| ピーク性能 | 12TFLOPS |
| 主記憶容量 | 4.6TB |
| ノード数 | 300ノード |
| プロセッサ数(コア数) | 300プロセッサ(1,200コア) |
| プロセッサ名 | SPARC64™ VII |
| コア数/プロセッサ | 4コア |
| 理論演算性能/コア | 10GFLOPS |
| メモリバンド幅/プロセッサ | 40GB/s |
| キャッシュ | 128MB(2nd 6MB/プロセッサ) |
| プロセッサ数/ノード | 1プロセッサ |
| 理論演算性能/ノード | 40GFLOPS |
| 主記憶容量/ノード | 16GB |
| ノード間ネットワーク帯域/ノード間最高通信性能 | 1.0GB/sec |
| ネットワークトポロジ | Fat Tree |
| 通信性能 | 2GB/s(片方向) |
| ノード間最高通信遅延(片方向)/ノード内最高通信性能 | 0.10μsec |




## 共有メモリ型演算サーバ
SMP Server

共有メモリ型演算サーバ「SPARC Enterprise M9000」は、1.9TFLOPSの理論演算性能（ピーク）を有するSMPサーバです。本システムは、1ノード構成で富士通クアッドコアプロセッサSPARC64™ VIIを48プロセッサ、主記憶384GBを搭載し、広大な共有メモリ空間を実現しています。本システムでは、主に次世代原子力システム開発のためのシミュレーション、安全評価などの計算処理を行います。

SMP（Symmetric Multiprocessing）：メモリ共有型並列コンピューティングやマルチプロセッシング方式。
SMPとは、複数のプロセッサによって処理を分散させるマルチプロセッサの方式のひとつ。
各プロセッサを等しく扱うことによって並列処理を行わせる方式のこと。全プロセッサが共通にアクセス出来るメモリ（空間）を持つ。

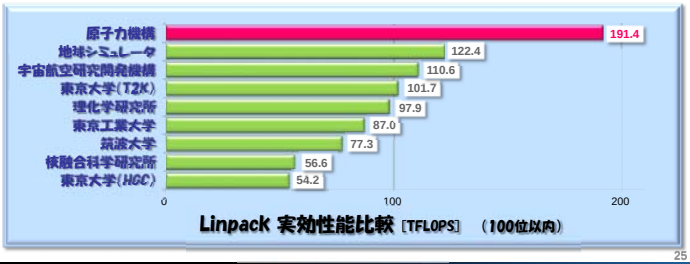
SPARC Enterprise M9000

| | |
|---|---|
| ピーク性能 | 1.9TFLOPS |
| 主記憶容量 | 384GB |
| ノード数 | 1ノード |
| プロセッサ数(コア数) | 48プロセッサ(192コア) |
| プロセッサ名 | SPARC64™ VII |
| コア数/プロセッサ | 4コア |
| 理論演算性能/コア | 10GFLOPS |
| キャッシュ | 128MB(2nd 6MB/プロセッサ) |

## ― Linpack ベンチマークテスト ―

| | Linpack性能 [TFLOPS] | ピーク性能 [TFLOPS] | 効率 [%] | 実行時間 [h] | 問題規模 [元] |
|---|---|---|---|---|---|
| BX900 | 191.40 | 200.1 | 95.66 | 13.40 | 2,401,344 |
| FX1 | 11.66 | 12.9 | 90.37 | 5.68 | 709,632 |

BX900のLinpack性能は、国内1位（実行効率1位）、世界22位（実行効率2位）
（Top500 2010.6）

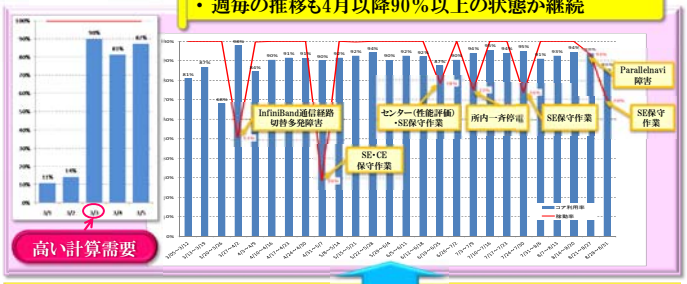





## 4 新スーパーコンピュータの利用状況

― BX900 ―

- 運用開始3日目（3月3日）にして、90%の利用率を達成
- 週毎の推移も4月以降90％以上の状態が継続



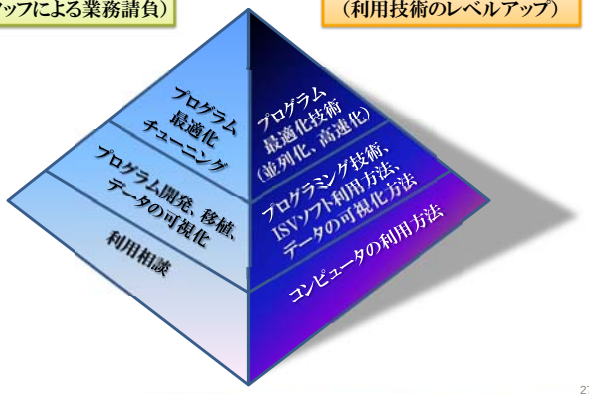

- システム利用に対する充実したサポート体制が確立されている（ユーザプログラムの最適化チューニング、プログラム移植、講習会開催、利用相談体制）
- 旧システム時代から主要コードの高並列化・最適化が進んでいる
- 高並列スーパーコンピュータを使う能力を有する研究者が数多く育っている



## ― 原子力機構のスーパーコンピュータ利用支援体制 ―

利用者支援
（専門スタッフによる業務請負）

講習会・セミナー
（利用技術のレベルアップ）






## ― 国内機関のスーパーコンピュータ利用支援体制 ―

原子力機構

プログラムの最適化（高速化）や利用相談対応など、充実した利用支援体制の下、研究者のサポートと計算機資源の有効利用

| 機関 | 利用相談 | プログラム最適化 | プログラム作成 | 備考 |
|---|---|---|---|---|
| JAEA | ●（5） | ●（7） | ●（6） | （ ）内は要員数 |
| A機関 | ● | | | |
| B機関 | ● | | | |
| C機関 | ● | ● | ● | プログラム最適化は相談・チューニング支援レベル |
| D機関 | ● | ● | | |
| E機関 | ● | | | |
| F機関 | ● | ● | | |
| G機関 | ● | | | |
| H機関 | ● | | | |



## 5 増大する計算需要への対応

### ― 大型計算機利用課題評価制度 ―

大型計算機資源の効率的・効果的利用を促進するために、平成17年度より、「大型計算機利用委員会」を設置し、大規模利用課題を評価、計算時間を配分

| 対象計算機 | | BX900（資源の約90%を承認課題に配分） |
|---|---|---|
| 課題申請条件 | | 年間 250,000コア時間超、または256並列超 |
| 評　価 | ① | 計算コードの実行効率、並列化効率 |
| | ② | 「研究成果の期待度」、「必要性及び緊急性の程度」、「計算時間設定の妥当性」の3項目について委員会による5段階評価 |
| | ③ | 年度末に利用結果「得られた成果の重要度、達成度及び費用（CPU時間）対効果」について委員会による5段階評価。<br>次年度への継続課題については、この結果を次年度評価に反映 |

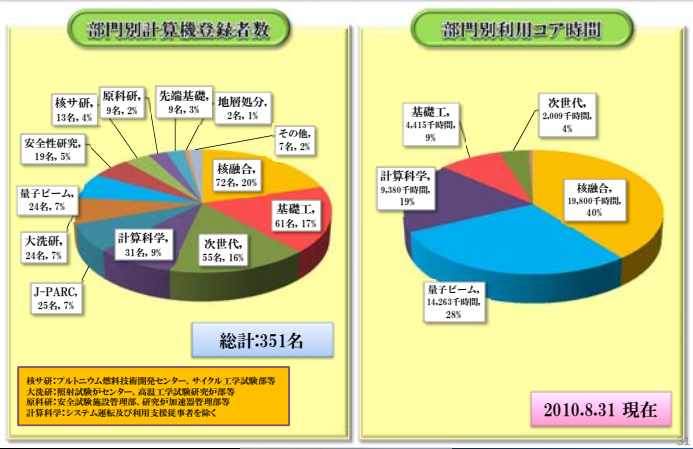


## ― 利用課題申請状況と配分 ―

高い潜在計算需要

- 計算機能力の1.5倍を超える課題申請
- 新システムの演算性能は、旧システムの約14倍と飛躍的に向上したが、各ユーザの計算規模も大幅に拡大（H22年度）

## －原子力機構における計算科学をリードする研究部門－




Japan Atomic Energy Agency

## Ⅲ. 新スーパーコンピュータの性能

1. 原子力プログラムの特徴
2. 新スーパーコンピュータの基本性能 ※Web掲載資料からは削除

Japan Atomic Energy Agency

## 1 原子力プログラムの特徴

| コード | 対象 | 手法 | 並列化 | 特徴 |
|---|---|---|---|---|
| GT5D | プラズマ乱流 | 差分法<br>有限要素法<br>FFT | Hybrid | 高演算効率<br>（メモリアクセスがボトルネック） |
| NOPIC | レーザー伝播 | 差分法 | FlatMPI | 1対1通信（exchange） |
| exact-diag | 固体中の電子 | 固有値問題<br>（CG） | Hybrid | 集団通信（scatter） |
| TPFIT | 圧縮性流体 | 行列解法<br>（ILUCGS） | FlatMPI | 低キャッシュヒット率<br>（メモリアクセスがボトルネック） |
| AQUA | 非圧縮性流体 | 差分法<br>（ICCG） | OpenMP | （共有メモリ型） |

演算性能、メモリバンド幅及び通信バンド幅のバランスのとれた計算機が必要

Japan Atomic Energy Agency

## Ⅳ. 京の利用に向けたプログラム開発

1. GT5Dコード
2. ハイブリッド並列化
3. FX1を用いた性能予測
4. 1PFLOPSまでのスケーラビリティ

Japan Atomic Energy Agency

## 1 GT5D: 核融合プラズマ乱流シミュレーションコード*

計算コード名：GT5D

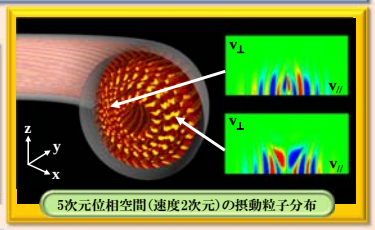
* Idomura et al.,Comput. Phys. Commun.(2008)

**概　要**

核融合プラズマ乱流の第一原理モデルである5次元ジャイロ運動論モデルの直接数値シミュレーションを行い、核融合炉心性能を評価する。

**アルゴリズム**

- 5次元位相空間の移流・拡散
  - 無散逸・保存型差分スキーム
- 時間積分
  - 半陰的ルンゲ・クッタ法
- ハイブリッド並列
  - MPI×(自動並列化+OpenMP)




5次元位相空間(速度2次元)の揺動粒子分布

**今後の計画**

- 臨界プラズマ試験装置JT-60U等の実験データに対する実証解析を実施し、炉心性能評価の確度を向上。 → 1ペタフロップス計算 （問題サイズ：240×240×256×128×32）
- 国際熱核融合実験炉ITERの炉心性能の予測や新たな運転手法の開発に活用。
- 核融合原型炉の開発において、実機試験が困難な核融合炉心の設計支援に活用。

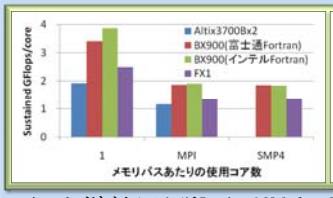
Japan Atomic Energy Agency

## －GT5Dコードの差分演算カーネルの演算性能比較－

**コア演算性能とメモリバンド幅のGT5Dカーネル処理性能への影響**

| 4次元移流項の4次精度森西差分カーネル、配列サイズ：48x80x64x100=200MB | | | | |
|---|---|---|---|---|
| ・ | Altix3700Bx2 | Itanium2 (6.4GF/core,L3=6MB,B/F=0.5) | 1 or 2 コア使用 | as is版 |
| ・ | BX900 | Nehalem-EP (11.72GF/core,L3=8MB,B/F=0.5) | 1 or 8 コア使用 | as is版 |
| ・ | FX1 | SPARC64VII (10GF/core,L2=6MB,B/F=1) | 1 or 4 コア使用 | FX1用最適化版 |

インテルFortran：-O3 (-openmp)，完全OpenMP化
富士通Fortran：-Kfast -Kocl -x200 [-Kmfunc=2, -Kprefetch_cache_level=1] (-Kparallel,OMP)，自動並列化+OpenMP化、　[ ]:FX1用オプション、():SMPオプション
※MPIは通信なしのembarrassingly parallelでノード（メモリバス）を埋めた計算





- ノードあたり1コアではItanium2,Nehalem-EPが30%程度の実効性能
- ノードが埋まった状態では15%程度まで実行性能が低下

Japan Atomic Energy Agency

## －GT5Dコードの差分演算カーネルの演算性能比較－

### GT5Dによるハイブリッド処理性能ベンチマーク

**問題サイズ：240x240x32x128x32**

BX900:Nehalem-EP(11.72GF/core,L3=8MB,B/F=0.5),4096コア使用

インテルFortran：-O3, フラットMPI(コード全体の完全OpenMP化は未対応)

富士通Fortran：-Kfast -Kocl -x200 -Kparallel,OMP, ハイブリッド並列(MPI,自動並列化,OpenMP)

| | BX900 Intel-SMP1 | BX900 Fujitsu-SMP1 | BX900 Fujitsu-SMP4 | BX900 Fujitsu-SMP8 |
|---|---|---|---|---|
| 全体/step (msec) | 10231.35 | 10397.27 | 9286.57 | 10928.65 |
| 通信/step (msec) | 2414.72 | 2608.57 | 1516.63 | 1533.82 |
| メモリ使用量(GB) | 3641.14 | 3754.09 | 1545.49 | 1249.91 |
| 実効性能(Gflops) | 4979.06 | 4899.60 | 5485.61 | 4661.37 |

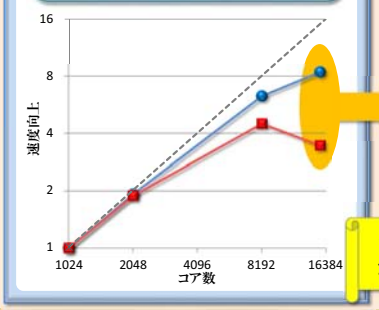
- ハイブリッド並列によるメモリコストの大幅な削減が可能
  - ベンチマークケースでは領域分割袖領域やフィールドソルバ演算子レプリカの削減によって8SMPで約1/3までメモリ使用量を削減
- MPI通信コストの減少による全体処理性能の高速化
- BX900では4SMPで最大性能が期待できる
- 富士通Fortranの自動並列化はかなりの部分で有効



## － Hybrid 並列 －

- スレッド並列によるメモリ使用量の大幅な削減
- MPIプロセス間通信量の減少による全体処理性能の高速化

問題サイズ: 240×240×64×128×32



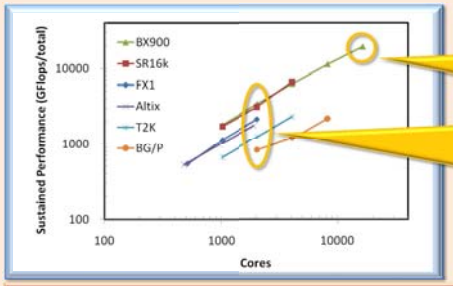

BX900における計算コスト

| | FlatMPI (15360) | Hybrid (3840×4) |
|---|---|---|
| 経過時間 /10step (秒) | 188.3 | 75.0 |
| 内、通信時間 (秒) | 145.0 | 34.7 |
| メモリ使用量 (TB) | 15.0 | 4.7 |

富士通Fortranの自動並列化はかなりの部分で有効



## －GT5Dコードの並列処理性能のstrongスケーリング－




**BX900/16384コア**

実効性能~19.4TF (10.1%)

**2000コアの性能比較**

BX900: ~3.42TF (14.2%)
SR16k: ~3.12TF (8.1%)
FX1 : ~2.14TF (10.5%)
Altix: ~1.75TF (14.3%)
T2K : ~1.23TF (6.6%)
BG/P : ~0.81TF (11.7%)

**格子数：240x240x64x100x32(Altix,T2K), 240x240x64x128x32(BX900,FX1,SR16k,BG/P)**

| | | | | | |
|---|---|---|---|---|---|
| • | BX900 | Nehalem-EP | (11.72GF/core,L3-8MB) | InfinibandQDR (8GB/s) | ハイブリッド並列4SMP |
| • | SR16000 | POWER6 | (18.8GF/core,L3-32MB) | InfinibandDDR (32GB/s) | ハイブリッド並列4SMP(SMT) |
| • | FX1 | SPARC64VII | (10GF/core,L2-6MB) | InfinibandD (2GB/s) | ハイブリッド並列版4SMP |
| • | Altix3700Bx2 | Itanium2 | (6.4GF/core,L3-6MB) | NUMAlink (3.2GB/s) | フラットMPI |
| • | T2K(HA8000) | Opteron | (9.2GF/core,L3-2MB) | Myrinet-10G (5GB/s) | ハイブリッド並列4SMP |
| • | BlueGene/P | PowerPC450 | (3.4GF/core,L3-8MB) | BG/P Network (5.1GB/s) | フラットMPI |



## 3 性能予測－FX1から京へ:4096並列(500TFlops)の場合－

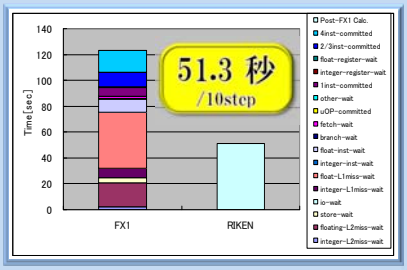
### 演算時間の推定

- 問題を1プロセスあたりの規模を保ったまま、4096プロセス並列から256プロセス並列にスケールダウン(weak scaling)
- FX1の256CPU(1024コア)を用いてプロファイリング。
- 変換表を用いて演算時間を推定。

### 通信時間の推定

- 通信の種類，データ長，通信回数を調査。
- インターコネクトの仕様(トポロジーを仮定)を もとに机上計算により推定。

24.6 秒 /10step



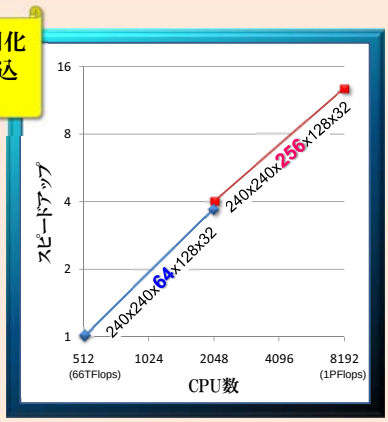

**実行性能：26.9TFlops(5.1%)**



## 4 1PFLOPSまでのスケーラビリティ

1PFlopsにおいても良い並列化性能(スケーラビリティ)が見込まれる

| 問題サイズ | CPU数 | 時間推定値 (秒/10step) |
|---|---|---|
| 240×240 ×64 ×128×32 | 512 | 227.6 |
| | 2048 | 61.2 |
| 240×240 ×256 ×128×32 | | 225.9 |
| | 8192 | 71.0 |

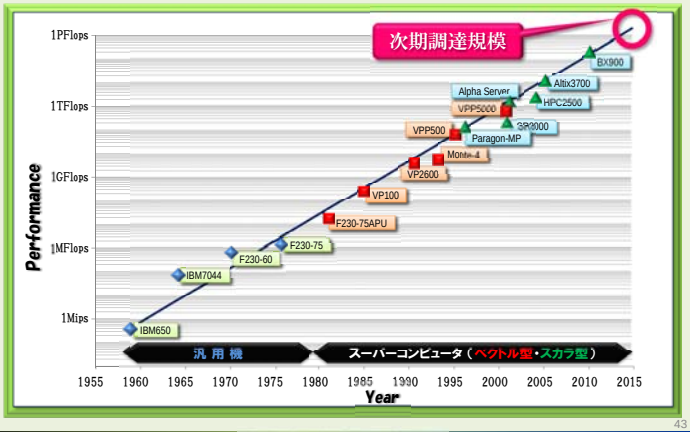





## VI. ま と め

- 原子力研究開発を進める上で、計算科学技術の積極的活用は必須
- 原子力機構では、既に年間の論文数の20%以上が計算科学技術を用いた研究にシフト
- 計算機資源の不足は深刻であり、一研究所で準備できる資源量を超過
- 京コンピュータの積極的利用が不可欠
- FX1を用いた京コンピュータ用チューニングは極めて有効
- 原子力コードについて、1PFlopsまでの良好なスケーラビリティを確認

## ― JAERI・JAEA 主要導入計算機 ―






## ―世界のスーパーコンピュータ整備動向（導入計画）―

| 国 内 (500Tflops以上) | | |
|---|---|---|
| 1) | 九州大学 | スーパーコンピュータ システム |
| | 総演算性能:500Tflops | 稼働開始:2011年/5月以降 |
| 2) | 東京大学 | 大規模超並列 スーパーコンピューター システム |
| | 総演算性能:1Pflops | 稼働開始:2011年/10月以降 |
| 3) | 東京工業大学 | クラウド型グリーン スーパーコンピュータ |
| | 総演算性能:2.4Pflops | 稼働開始:2010年 |
| 4) | 理化学研究所 | 京 |
| | 総演算性能:10Pflops | 稼働開始:2012年/6 |

| 国 外 | | |
|---|---|---|
| 1) | イリノイ大 | BlueWaters |
| | 総演算性能:1～10Pflops | 稼働開始:2011年 |
| 2) | ローレンス・リバモア | Sequoia |
| | 総演算性能:20Pflops | 稼働開始:2012年 |
| 3) | NASA | Pleiade |
| | 総演算性能:10Pflops | 稼働開始:2012年 |
| 4) | ロスアラモス | Cielo |
| | 総演算性能:1Pflops | 稼働開始:2010年 |
| 5) | ロスアラモス | Roadrunner ? |
| | 総演算性能:20Pflops | 稼働開始:2012年 |
| 6) | EU | PRACE |
| | 総演算性能:？Pflops | 稼働開始:2010年 |
| 7) | EU | ITER-BA |
| | 総演算性能:1Pflops | 稼働開始:2012年 |
| 8) | 中国 | Nebulae |
| | 総演算性能:3Pflops ? | 稼働開始:2010年 |



## ―世界のスーパーコンピュータ動向（Top500）―



| 2010.6 | 台数 | 割合 [%] | Rmax合計 [TFLOPS] | 割合 [%] | 世界ランク (国内1位) | Rmax* [TFLOPS] |
|---|---|---|---|---|---|---|
| 米国 | 282 | 56.4 | 17,969.3 | 55.4 | 1 | 1,759 |
| 英国 | 38 | 7.6 | 1,785.4 | 5.5 | 16 | 275 |
| フランス | 27 | 5.4 | 1,706.9 | 5.3 | 18 | 238 |
| 中国 | 24 | 4.8 | 2,992.6 | 9.2 | 2 | 1,271 |
| ドイツ | 24 | 4.8 | 2,248.5 | 6.9 | 5 | 826 |
| ロシア | 11 | 2.2 | 814.7 | 2.5 | 13 | 350 |
| インド | 5 | 1.0 | 283.4 | 0.9 | 33 | 133 |
| 韓国 | 1 | 0.2 | 274.8 | 0.8 | 14 | 275 |
| 日本 | 18 | 3.6 | 1,252.7 | 3.9 | 22 | 191 |

※ Rmax:Linpack性能値

◆日本の計算機資源の割合は、原子力機構のスパコン導入により前回（2009.11）よりは僅かに回復したものの、欧米や中国に比べ急減している。

◆中国はもはや米国に次ぐスパコン大国に。日本との差は急拡大中。

◆日本の国内1位（原子力機構）の計算機は、ロシア、韓国の1位にも及ばない。



Center for computational science & e-systems

科学技術計算分科会 選出
2010 年度 科学技術計算分科会 会合 より

# 3 次元 MHD コードによる FX1 の性能評価

名古屋大学太陽地球環境研究所
荻野 竜樹

サイエンティフィック・システム研究会　2010 年 10 月 20 日
SS 研科学技術計算分科会

# 3次元MHDコードによる FX1の性能評価

荻野竜樹
名古屋大学太陽地球環境研究所

［アブストラクト］

並列型スパコン FX1 や HX600 の利用により、地球磁気圏のグローバル構造とメソスケールの境界層渦乱流の自己無撞着な大規模高精度シミュレーションができる時代が訪れてきたが、その実現にはさらに高効率の並列計算コードの開発が不可欠である。ここに、キャッシュヒット率向上と多次元領域分割／一括転送による通信量最小化を可能にした高効率の 3 次元 MHD コード開発に成功し、FX1 の 3072 コア数まで 15-20%以上の実行効率を得ることができた。Flat MPI と Hybrid (Impact)の比較では、Flat MPI が最速だったが、コア数によらず Hybrid もよい実行効率を得た。通信速度と計算速度の比は 3072 コアでも 4%以下であり、通信速度は 10 万コア程度までボトルネックにならずにスケーラビリティが伸びることが期待できる。もちろん、実際にテストして実証する必要がある。

［キーワード］

地球磁気圏の MHD シミュレーション、高効率の並列コード開発、多次元領域分割と通信量最小化、Flat MPI と Hybrid の比較、スーパーコンピューティング

## １．はじめに

スペースプラズマや核融合プラズマの最も特異な性質はその異常輸送現象にあるといえる。特に顕著な例として温度の輸送や磁場の拡散は粒子間衝突から得られる古典的な輸送係数の千倍から 1 万倍に達することもあることが観測や実験から知られている。しかし、その基本的な物理過程の説明は今日でも一致した考えに達していないといえる。プラズマの異常輸送の有力な説明には二つある。一つは、マクローメソーミクロスケールが必ず結合していてメソスケール現象が決定的な役割を果たしているとする説である。もう一つは、マクロとミクロスケールの直接結合が本質的でメソスケールは重要な役割を果たしていないとする説である。ここに、最近の並列型スパコンの急速な進歩により MHD（電磁流体力学的）モデルからマクローメソスケール現象を自己無撞着に解くことができるような時代が実際に訪れてきた。即ち、メソスケール現象がプラズマの異常輸送現象に本質的な役割を果たしているかどうかに決着をつけるような大規模な 3 次元 MHD シミュレーションを実行できる段階に達してきたといえる。もちろん、ミクロスケール現象は MHD モデルで扱うことができないので、超粒子モデルやブラソフモデルなどの運動論的モデルによるシミュレーションが不可欠である。

そのマクローメソスケール結合を調べるには、太陽風と地球磁気圏相互作用のグローバル３次元 MHD シミュレーションが一つの典型的な対象となる。なぜなら、磁気圏のマクロ構造の中に物理量が急峻に空間変化する境界層があり、そこにメソスケールのプラズマ乱流が発生してプラズマの異常輸送を引き起こしている可能性が高く、グローバルモデルから境界層のプラズマ不安定とその非線形発展と結果としての渦乱流発生を自己矛盾なく解ける段階に達したからである。しかし、その大規模 MHD シミュレーションを実行するためには大規模並列型スパコンの効率を最大化するプログラムの開発が不可欠である。

## ２．MHD コードの並列化と超並列スパコンの有効利用

MHD モデルでは流体の方程式と Maxwell 方程式を高精度の数値計算法の一つである Modified Leap-Frog 法で初期値境界値問題として解く。その 3 次元 MHD コードを MPI Fortran を用いて並列化する。その際、CPU 数が 1 千個を越えて 1 万個、10 万個程度まで計算速度が劣化しないようなよいスケーラビリティが得られる並列計算コードの開発を目指す。

私達は、これまで富士通の VPP5000/64、PRIMEPOWER HPC2500 1536core などで MPI を利用してそれぞれ 60%、20%を超える実行効率を出す 3 次元 MHD コードを開発してきた。その知識と経験を活かし、高効率の並列計算 3 次元 MHD コードを開発して、平成 21 年度に名古屋大学情報基盤センターに導入された富士通の FX1 3072core と HX600 2048core を用いてテストした。作成した MHD コードは基本的に 4 種類に分類できる。それらは、1 次元領域分割法（1Da）、2 次元領域分割法（2Da）、3 次元領域分割法（3Da）およびキャッシュヒットを上げるために MHD 方程式の配列成分を第 1 変数に移動した 3 次元領域分割法（3Db）である。2、3 次元領域分割法を用いると分散した CPU 間の通信量が小さくなり、これまでのテスト計算からは、スカラー並列機 HPC2500 では（3Db）が最速、ベクトル並列機 Earth Simulator では（2Da）が最速であった。

## ３．FX1 と HX600 における３次元 MHD コードの計算速度

FX1 は計算速度とメモリバンド幅が同程度との特徴があるスパコンで、Flat MPI と Hybrid（Impact）両方の利用ができて、Hybrid は 1 ノード 4 コアを共有メモリとしてスレッド並列化し、ノード間をプロセス並列（MPI）で利用する。そのどちらが高速計算できるかが最初の注目点であったが、ほとんどの場合 Flat MPI が最速になっている。ただし、Hybrid の場合もそれほど劣化しない。また、かなりバラつきはあるが、3072 コア利用までスケーラビリティはよく保たれていた。前章の 4 種類の MHD コードを比較した場合、計算速度の速い順に並べると（3Db）＞（3Da）＞（2Da）＞（1Da）となり、（3Db）で 15-20%以上の実行効率を、（1Da）で 5-15%程度の実行効率を得ることができた。特に、（3Db）ではコア数が 3072 まで多くなってもほとんど計算速度の劣化は起らず、バランスが良い場合は若干ではあるがむしろ計算効率が向上する場合もあった。JAXA の FX1 でテストした場合は、多数コア（4096、5760 コア）で Hybrid（Impact）が Flat MPI よりも高速になる場合があった。これは大変興味ある結果であるが、MHD 変数の大きさや 3 次元方向の並列分割数がより適切に取られていたかの疑問もあり、継続的なテストが必要である。また、計算時間と通信時間を分離して計測したところ、通信時間と計算時間の比は 256-3072 コアの範囲で 3-4%であり、しかもその比は 3072 コアまであまり変らないので、スケーラビリティは 1 万コアを超えてもよく成り立つことが期待される。しかし、実際はより多くのコア数で実証確認する必要がある。

HX600 も Flat MPI が最速であり、スレッド並列数を増やして Hybrid にすると計算速度がかなり劣化する傾向が明確であった。私のテストでは、最適値を求めた場合 HX600 は FX1 より常に少し計算速度が低めであった。一方、共同研究者のほとんどは逆の結果を得ていて、HX600 が FX1 よりも並列効率がより良いと結論付けている。その結果の違いがどこからきているかはまだ明確ではないが、一つの可能性として 1 コア当りのメモリ使用量、即ち粒度が関係していると予想される。

## ４．おわりに

学際大規模情報基盤共同研究・共同拠点のプロジェクトで他の大学の情報基盤センターのスパコンを利用する機会を得た。それらは SX-9、HA8000、SR16000、TSUBAME である。4 種類の MHD コードでテストしたが、どのスパコンも期待したとおりのよい結果が得られ、スケーラビリティもテストしたコア数（3072 コア）の範囲でよく成り立っていた。気がついた点として、T2K のスパコン（HX600，HA8000）は（2Da）が最速で（3Da）がそれに続き、（3Db）は逆に遅かった。一方、SR16000 は FX1 と同様に（3Db）が最速で両者は似た傾向を示した。これらの結果は FX1 と SR16000 ではキャッシュのヒット率向上が重要であることを意味している。また、SR16000 は SMT が非常によく効き約 2 倍の速度がでた。一方、FX1 では SMT による計算速度向上の効果は僅かであった。T2K のスパコンで（2Da）と（3Da）が速いのは「擬似ベクトル機能が効くためと推測しているが、理由究明にはさらなる系統的なテストが必要である。

SS研科学技術計算分科会
「ハイパフォーマンス・コンピューティングの現在と未来」
2010年10月20日（水）、 ANAクラウンプラザホテル神戸

# 3 次元MHD コードによる FX1 の性能評価


名古屋大学太陽地球環境研究所
荻野竜樹

共同研究者
梅田隆行 （名古屋大学太陽地球環境研究所）
深沢圭一郎 （九州大学大学院理学研究院地球惑星科学部門）




## アウトライン

1．背景
2．地球磁気圏の３次元グローバルMHDモデル
3．PRIMEPOWER HPC2500からFX1とHX600への移行
4．FX1とHX600おける３次元MHDコードの計算速度
5．種々のスパコンの計算速度の比較と考察
6．地球磁気圏の３次元MHDシミュレーションの将来
7．まとめ

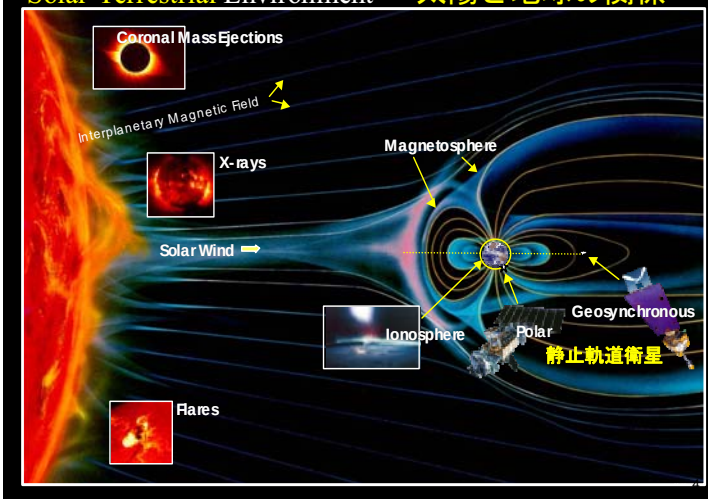


## 1．背景

- 太陽地球系科学：太陽活動とジオスペース（地球外圏）環境変化の因果関係を解明する研究
- 外部境界をできるだけ遠くに設定してかつ空間分解能をできるだけ上げる
- 計算方法の改良と最大級のスーパーコンピュータの効率的な利用
- 並列計算共通プログラム／手法の開発
  Message Passing Interface (MPI)
- 超並列スーパーコンピュータの効率的な利用
  富士通株式会社：PRIMEPOWER HPC2500
  からFX1とHX600へ

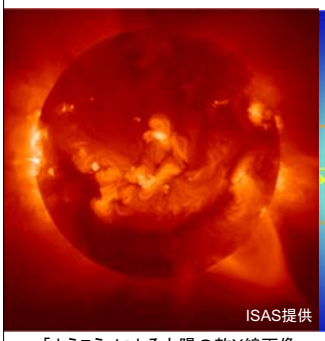


## Solar-Terrestrial Environment　太陽と地球の関係

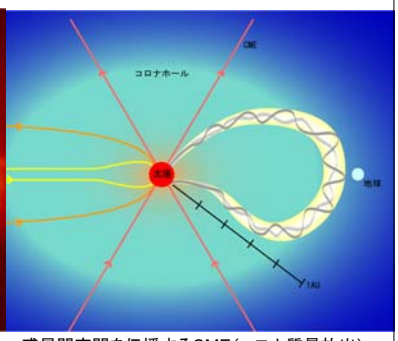



## 太陽表面爆発現象と惑星間擾乱の伝播

太陽フレアなどの太陽活動はジオスペース環境変化の源




「ようこう」による太陽の軟X線画像

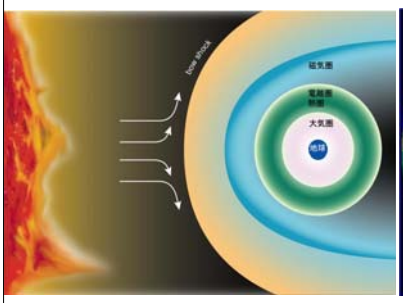



惑星間空間を伝播するCME（コロナ質量放出）

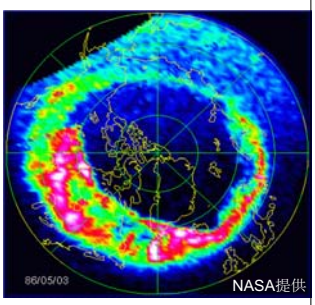


## 磁気圏・電離圏・熱圏結合：領域間結合

太陽風のエネルギーは磁気圏・電離圏・熱圏の領域間結合を介して地球内部に運ばれ、極域でオーロラを輝かせる。宇宙天気研究はその領域間結合を調べ、予報に結びつける。

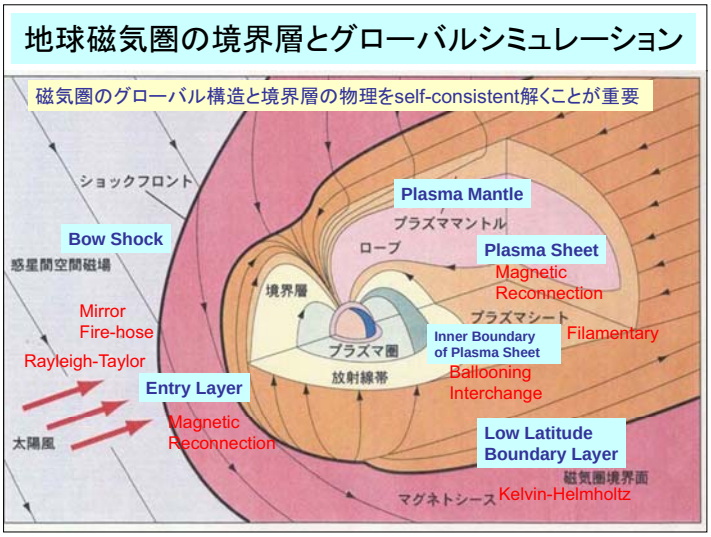


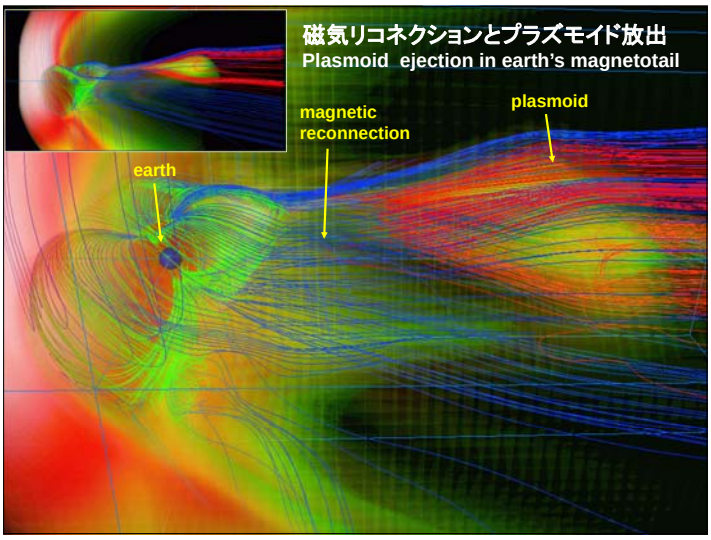
太陽風－磁気圏－電離圏・熱圏－大気圏結合システム

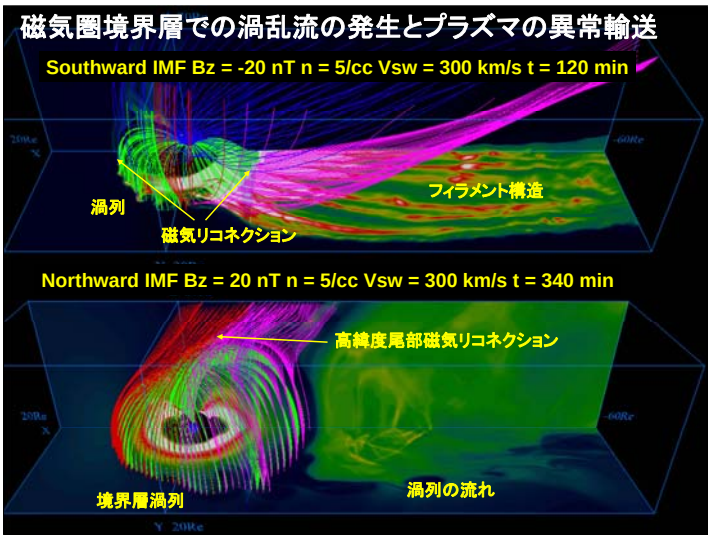


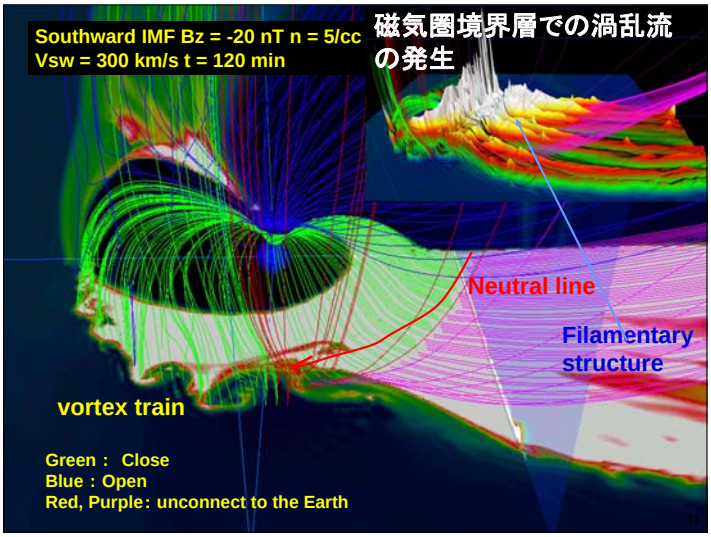

磁気嵐時の衛星オーロラ画像

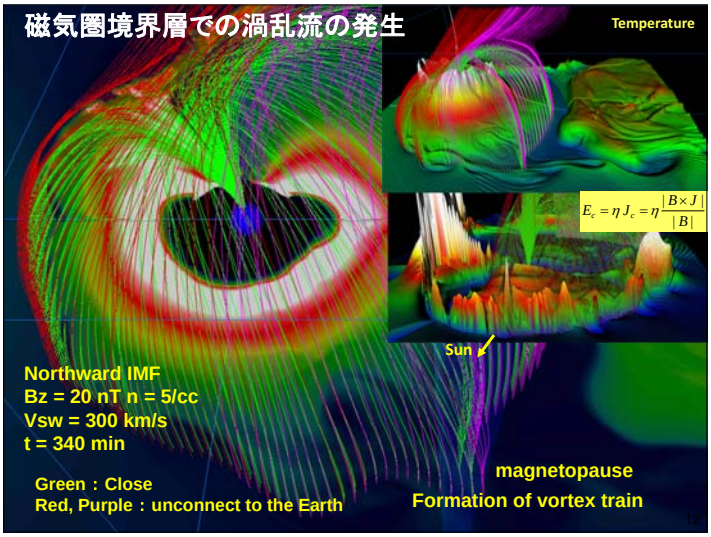

北極域のオーロラ


地球磁気圏の境界層とグローバルシミュレーション
磁気圏のグローバル構造と境界層の物理をself-consistent解くことが重要
ショックフロント
Bow Shock
惑星間空間磁場
Plasma Mantle
プラズママントル
ローブ
Plasma Sheet
Magnetic
Reconnection
境界層
プラズマシート
Mirror
Fire-hose
Rayleigh-Taylor
プラズマ圏
Inner Boundary
of Plasma Sheet
Filamentary
放射線帯
Ballooning
Interchange
Entry Layer
太陽風
Magnetic
Reconnection
Low Latitude
Boundary Layer
磁気圏境界面
マグネトシース
Kelvin-Helmholtz


磁気リコネクションとプラズモイド放出
Plasmoid ejection in earth's magnetotail
magnetic
reconnection
plasmoid
earth

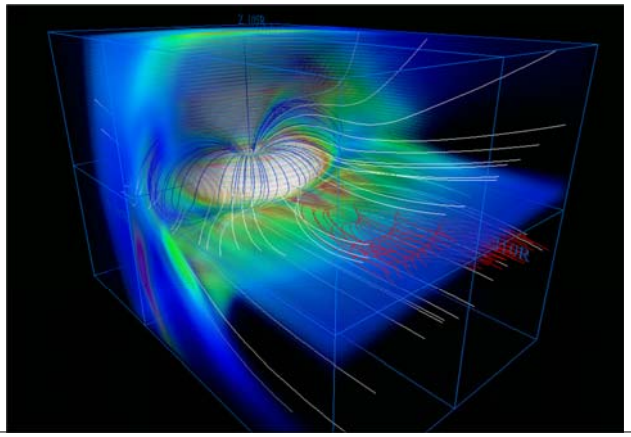

磁気圏境界層での渦乱流の発生とプラズマの異常輸送
Southward IMF Bz = -20 nT n = 5/cc Vsw = 300 km/s t = 120 min
渦列
磁気リコネクション
フィラメント構造
Northward IMF Bz = 20 nT n = 5/cc Vsw = 300 km/s t = 340 min
高緯度尾部磁気リコネクション
境界層渦列
渦列の流れ


Southward IMF Bz = -20 nT n = 5/cc
Vsw = 300 km/s t = 120 min
磁気圏境界層での渦乱流の発生
Neutral line
Filamentary
structure
vortex train
Green : Close
Blue : Open
Red, Purple: unconnect to the Earth


磁気圏境界層での渦乱流の発生
Temperature
$E_c = \eta J_c = \eta \frac{|B \times J|}{|B|}$
Sun
Northward IMF
Bz = 20 nT n = 5/cc
Vsw = 300 km/s
t = 340 min
Green : Close
Red, Purple : unconnect to the Earth
magnetopause
Formation of vortex train

## 高速回転する木星磁気圏でのプラズモイドの構造

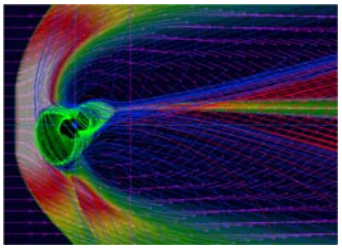
Patchy plasma sheet with IMF By and Bz



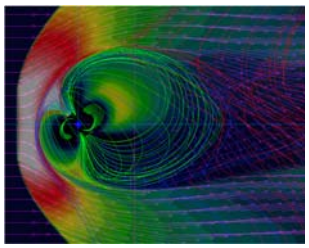


## 2. 地球磁気圏の3次元グローバルMHDシミュレーションモデル

南向き惑星間磁場



北向き惑星間磁場





## 基礎方程式：MHD方程式とMaxwell方程式

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\mathbf{v}\rho) + D\nabla^2 \rho \qquad (1)$$

$$\frac{\partial \mathbf{v}}{\partial t} = -(\mathbf{v}\cdot\nabla)\mathbf{v} - \frac{(\nabla p)}{\rho} + \frac{\mathbf{J}\times\mathbf{B}}{\rho} + g + \frac{\mathbf{\Phi}}{\rho} \qquad (2)$$

$$\frac{\partial p}{\partial t} = -(\mathbf{v}\cdot\nabla)p - \gamma p \nabla\cdot\mathbf{v} + D_p \nabla^2 p \qquad (3)$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla\times(\mathbf{v}\times\mathbf{B}) + \eta\nabla^2\mathbf{B} \qquad (4)$$

$$\mathbf{J} = \nabla\times(\mathbf{B}-\mathbf{B}_d) \qquad (5)$$



## スカラー並列機で高速計算を実現するために

並列計算のcpu数が大幅に増加
通信量を大幅に減らす
キャッシュのヒット率を上げる

## 3次元MHDコードの領域分割法による並列化

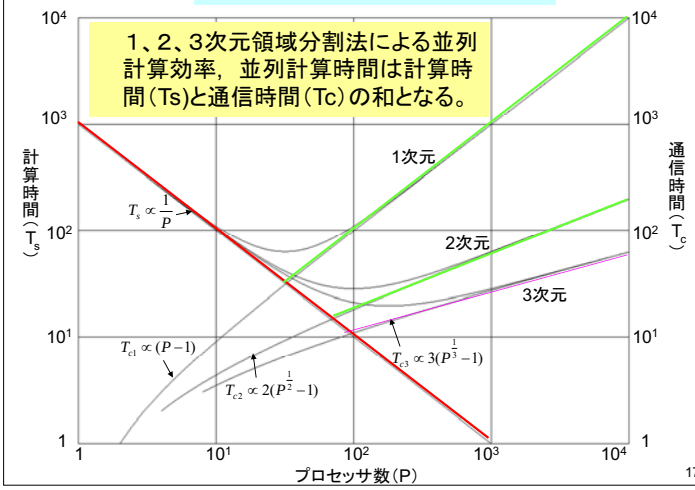
| | 計算時間 | 通信時間 |
|---|---|---|
| 1次元領域分割 | $Ts=k_1 N^3/P$ | $Tc=k_2 N^2 (P-1)$ |
| 2次元領域分割 | $Ts=k_1 N^3/P$ | $Tc=2k_2 N^2 (P^{1/2}-1)$ |
| 3次元領域分割 | $Ts=k_1 N^3/P$ | $Tc=3k_2 N^2 (P^{1/3}-1)$ |

k1とk2： 一定の係数
N： 3次元配列の1方向の変数量
P： 並列CPUの数



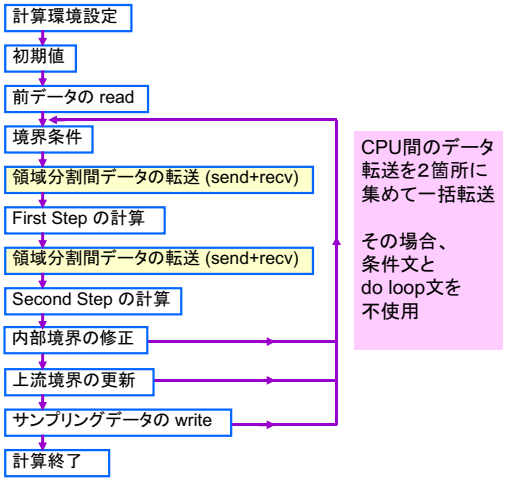
## 領域分割による並列計算効率

1、2、3次元領域分割法による並列計算効率，並列計算時間は計算時間(Ts)と通信時間(Tc)の和となる。






## MPIを用いたMHDコードの構造



## 3次元領域分割の方法

一括転送の2次元バッファを導入

```
CC MPI START
      parameter (npex=2,npey=2,npez=2)
      parameter (npe=npex*npey*npez,npexy=npex*npey)
      integer itable(-1:npex,-1:npey,-1:npez)
c
      parameter(nzz=(nz2-1)/npez+1)
      parameter(nyy=(ny2-1)/npey+1)
      parameter(nxx=(nx2-1)/npex+1)
      parameter(nxx3=nxx+2,nyy3=nyy+2,nzz3=nzz+2)
c
      dimension f(nb,0:nxx+1,0:nyy+1,0:nzz+1)
      dimension ftemp1x(nb,nyy3,nzz3),ftemp2x(nb,nyy3,nzz3)
      dimension ftemp1y(nb,nxx3,nzz3),ftemp2y(nb,nxx3,nzz3)
      dimension ftemp1z(nb,nxx3,nyy3),ftemp2z(nb,nxx3,nyy3)
c
```



## 領域分割されたCPU間のデータ転送1

```
c
      do k=-1,isizez
      do j=-1,isizey
      do i=-1,isizex
      itable(i,j,k)=MPI_PROC_NULL
      end do
c
      irank1=0
      do k=0,isizez-1
      do j=0,isizey-1
      do i=0,isizex-1
      itable(i,j,k)=irank1
      if(irank.eq.irank1) then
      irankx=i
      iranky=j
      irankz=k
      end if
      irank1=irank1+1
      end do
c
```

MPIで両端と中間のCPU間の転送を同じ形にするために、不実行を宣言する3次元のテーブルを作成

itable(-1:npex,-1:npey,-1:npez)

```
CC MPI START
      irightx = itable(irankx+1,iranky,irankz)
      ileftx  = itable(irankx-1,iranky,irankz)
      irighty = itable(irankx,iranky+1,irankz)
      ilefty  = itable(irankx,iranky-1,irankz)
      irightz = itable(irankx,iranky,irankz+1)
      ileftz  = itable(irankx,iranky,irankz-1)
C
CC MPI END
```



## 領域分割されたCPU間のデータ転送2

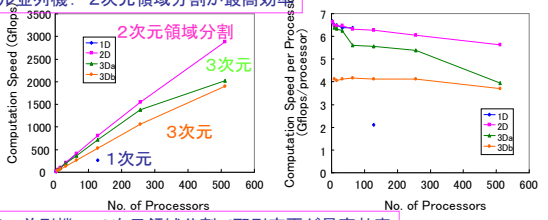
同期型の一括転送を利用

```
CC MPI START
      ftemp1x=f(:,is,:,:)
      ftemp1y=f(:,:,js,:)
      ftemp1z=f(:,:,:,ks)
      call mpi_sendrecv(ftemp1x,nwyz,mpi_real,ileftx,200,
     &                  ftemp2x,nwyz,mpi_real,irightx,200,
     &                  mpi_comm_world,istatus,ier)
      call mpi_sendrecv(ftemp1y,nwzx,mpi_real,ilefty,210,
     &                  ftemp2y,nwzx,mpi_real,irighty,210,
     &                  mpi_comm_world,istatus,ier)
      call mpi_sendrecv(ftemp1z,nwxy,mpi_real,ileftz,220,
     &                  ftemp2z,nwxy,mpi_real,irightz,220,
     &                  mpi_comm_world,istatus,ier)
C
      f(:,ie+1,:,:)=ftemp2x
      f(:,:,je+1,:)=ftemp2y
      f(:,:,:,ke+1)=ftemp2z
CC MPI END
```

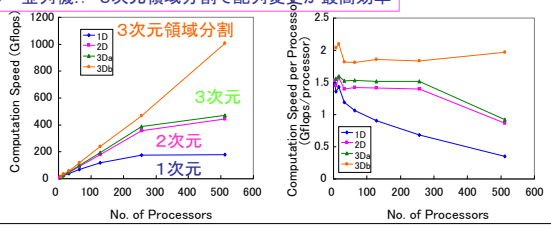


## ベクトル並列機（Earth Simulator）とスカラー並列機 PRIMEPOWER HPC2500での計算速度の比較

ベクトル並列機： 2次元領域分割が最高効率




スカラー並列機:： 3次元領域分割で配列変更が最高効率






## 3. PRIMEPOWER HPC2500から FX1とHX600への移行

（名古屋大学情報基盤センター）




## 名古屋大学情報基盤センターの新スーパーコンピュータ

**システムS1　M9000**
大規模な共有メモリ
3ノード x 128コア （1ノードはFX1のフロントエンド）
ノード当り：演算性能：1.28TFlops, メモリ：1TB
**シミュレーション結果の解析と画像処理に利用**

**システムS2　HX600**
クラスター型計算機（T2K型）ノード内は共有メモリ
160ノード x 16コア
ノード当り：演算性能：160GFlops, メモリ：64GB
**中規模シミュレーションに利用**

**システムS3　FX1**
大規模分散並列型、次世代スパコンとの連携
768ノード x 4コア
ノード当り：演算性能：40GFlops, メモリ：32GB
メモリバンド幅：40GB/s
**大規模シミュレーションに利用**

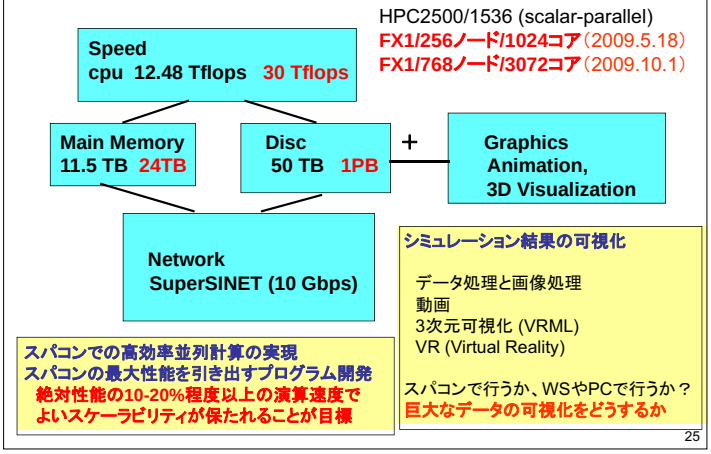
**大規模シミュレーションはシステムS3 (FX1) Hybrid (Impact)の利用を推奨**

総ノード／コア数：768／3072
総演算性能：30.72TFlops
総メモリ：24TB

**FX1利用の計算規模**

| | | | |
|---|---|---|---|
| コア数 | 256 | 512 | 1024 |
| メモリ (TB) | 2 | 4 | 8 |
| 演算性能 (Tflops) | 2.56 | 5.12 | 10.24 |

## Four Key Functions of Supercomputer






## Flat MPIとHybrid（user並列＋自動並列）はどちらが最速か？

Hybridでは

「Node内はベンダーが最高効率の自動並列化を提供する

**Node間はユーザーが最高効率のプロセス並列プログラムを作成する**

この結果、最高効率のプログラムが得られるはずである？？？」

この命題は正しいか？

それともFlat MPIが最速か？



**Hybrid**ではノード内の並列化とノード間の並列化に齟齬は生じないのか？

- ・メモリ上に変数がどの様に配置されるのか？
- ・ノード間でどの様に領域分割して、分散メモリに配分しているかは本質的に重要
- ・ノード内でどの様に領域分割して、メモリに割り付けるか、疑問？（ノード内自動並列化はどうしているのか？）
- ・**キャッシュのヒット率向上と通信量の最小化**

**Flat MPI**では全部の配列変数をどの様にメモリに割り付けるかを指定できる

通信量の最小化とキャッシュのヒット率をトータルで最適化

ただし、高度な技術が必要

そのテンプレート（プログラムの構造）を作れば誰でも利用できるはず



もちろん、**Flat MPIとHybridはどちらが最速か**はスパコンによる

- ・Fujitsu FX1は1node 4 coreで、1 node内では共有メモリを持つ
- ・従って、Flat MPIとHybrid（Impact）はどちらが最速か、検証が必要

完全な分散型スパコンであれば、Flat MPIが最速になると一般に信じてられている。

その他の全てのスパコンでFlat MPIが最速になると予想している。**これも現在までの経験からの予想**

**通信量最小化＋並列化オーバーヘッド**（Flat MPI）と**共有メモリ間衝突とキャッシュのヒット率変化**（Hybrid）はトレードオフの関係



## スーパースカラー並列スパコンで最高効率の並列プログラムを作成するには

1．先ず、プログラムの構造の確立が最も重要

・計算での変数参照を元にプログラム構造を決定

配列変数定義の最小化

計算の明確なブロック化

計算と通信の完全な分離

定数と変数は意識して区別（dependencyを解く）

a(i)=a(i+j)の形で j は定数で定義

・ 配列の大きさ、do loopの回数（初数と終数）、配列内の変数は定数で定義

定数定義はparameter文を利用



## スーパースカラー並列スパコンで最高効率の並列プログラムを作成するには（その2）

・領域分割法の導入（2、3次元領域分割）

通信では一括転送を実施

・Dependencyを可能な限り取り除く

計算のメイン部分ではIf文や条件文は絶対に使用不可

定数定義の利用

Mask、List vector、Gather and scatterの最適な使用

・変数の一括変換と一括転送

これを条件文やdo loopを導入せずに実行

If文や条件文が並列化（擬似ベクトル化）できたとの表示にだまされてはいけない。⟶ その効率は実際非常に悪い。（**必ず実測すること**）

## ４．FX1とHX600おける３次元MHDコードの計算速度

| コンピュータ | グリッド数 | 総CPU CORE数 | プロセス並列数 | 自動並列数 | CPU時間(sec) | 1グリッド当りの計算時間(nsec) | 1グリッド当りの計算時間×総CPU CORE数 |
|---|---|---|---|---|---|---|---|
| HPC2500 | 1024*1024*1024 | 512 | 512 | 1 | 2.487 | 2.316 | 1186 |
| HPC2500 | 2048*2048*2048 | 1024 | 512 | 2 | 10.763 | 1.253 | 1283 |
| VPP5000 | 1002*1002*1120 | 56 | 56 | 1 | 5.794 | 5.152 | 289 |
| SX6 | 512*256*256 | 16 | 16 | 1 | 0.693 | 20.662 | 331 |
| Earth Sim | 2048*1024*1024 | 1024 | 1024 | 1 | 0.62 | 0.289 | 296 |
| HX600 | 512*512*512 | 64 | 64 | 1 | 1.451 | 10.813 | 692 |
| HX600 | 1024*1024*1024 | 512 | 512 | 1 | 1.906 | 1.775 | 909 |
| HX600 | 2048*2048*2048 | 512 | 512 | 1 | 19.654 | 2.288 | 1171 |
| FX1 | 512*512*512 | 64 | 64 | 1 | 1.337 | 9.964 | 638 |
| FX1 | 1024*1024*1024 | 512 | 512 | 1 | 1.377 | 1.283 | 657 |
| FX1 | 1024*1024*1024 | 1024 | 1024 | 1 | 0.752 | 0.701 | 684 |
| FX1 | 2048*2048*2048 | 512 | 512 | 1 | 10.525 | 1.225 | 627 |
| FX1 | 2048*2048*2048 | 1024 | 1024 | 1 | 5.215 | 0.607 | 622 |
| SR16000 | 508*508*508 | 64 | 64 | 1 | 1.236 | 9.424 | 603 |
| SR16000SMT | 508*508*508 | 32 | 64 | 1 | 1.239 | 9.448 | 605 |
| SR16000 | 1018*1018*510 | 256 | 256 | 1 | 1.256 | 2.376 | 608 |
| SR16000SMT | 1018*1018*1018 | 256 | 512 | 1 | 1.29 | 1.223 | 626 |

6列目：Tc：1回時間ステップを進めるのに要するCPU時間（sec）
7列目：Tcg：1グリッドにつき1回時間ステップを進めるのに要するCPU時間（nsec）
8列目：Tcg*Ng：Tcgと総CPU/CORE数の積（nsec）、数値が小さいほど高効率計算を示す。



## PRIMEPOWER HPC2500とFX1での計算速度の比較

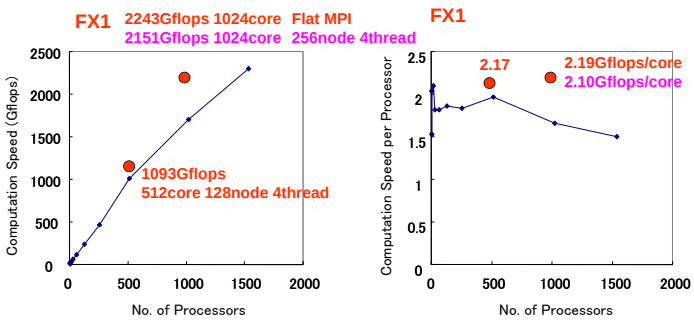



図6 PRIMEPOWER HPC2500でCPU数を増やした場合の計算速度（Gflops）と1CPU当たりの計算速度（Gflops/cpu）。・1536 cpuまでスケーラビリティは非常に良い

**Fujitsu FX1は、3次元領域分割とキャッシュヒットを改善した3次元MHDコードで128node, 512coreで22%が得られ、256node, 1024coreで21%の高効率計算.**

**大粒度計算での最高速は1024core Flat MPIで21%を達成。**



## FX1の計算速度 1

| コンピュータ | プログラム | グリッド数 | 総コア数 | プロセス並列数 | 自動並列数 | CPU時間(sec) | 計算速度(Gflops) | 効率(%) |
|---|---|---|---|---|---|---|---|---|
| FX1 | 1Da | 1024**3 | 256 | 64 | 4 | 152.0 | 9 | 0.4 |
| FX1 | 1Da | 1024**3 | 128 | 128 | 1 | 232.2 | 6 | 0.5 |
| FX1 | 1Da | 1124*1024**2 | 128 | 128 | 1 | 28.4 | 52 | 4.1 |
| FX1 | 2Da | 1024**3 | 128 | 128 | 1 | 17.1 | 79 | 6.2 |
| FX1 | 2Da | 1124*1024**2 | 128 | 128 | 1 | 8.23 | 180 | 14.1 |
| FX1 | 3Da | 1024**3 | 128 | 128 | 1 | 7.32 | 185 | 14.4 |
| FX1 | 3Db | 1024**3 | 128 | 128 | 1 | 5.22 | 259 | 20.2 |
| FX1 | 2Da | 3072*2048**2 | 3072 | 3072 | 1 | 17.2 | 1004 | 3.3 |
| FX1 | 2Da | 3072*2048**2 | 3072 | 768 | 4 | 19.7 | 822 | 2.7 |
| FX1 | 3Da | 3072*2048**2 | 3072 | 3072 | 1 | 3.91 | 4145 | 13.5 |
| FX1 | 3Da | 3072*2048**2 | 3072 | 768 | 4 | 4.36 | 2782 | 9.1 |
| FX1 | 3Db | 3072*2048**2 | 3072 | 3072 | 1 | 3.29 | 4932 | 16.1 |
| FX1 | 3Db | 3072*2048**2 | 3072 | 768 | 4 | 3.76 | 4321 | 14.1 |

1Da：1次元領域分割、2Da：2次元領域分割、3Da：3次元領域分割、3Db：3次元領域分割＋f(m,i,j,k)
グリッド数の選び方によっては極端に計算効率が劣化する場合があるが、少し調節すると改良できる。



## FX1の計算速度 2

| コンピュータ | プログラム | グリッド数 | 総コア数 | プロセス並列数 | 自動並列数 | CPU時間(sec) | 計算速度(Gflops) | 効率(%) |
|---|---|---|---|---|---|---|---|---|
| FX1 | 3Db | 1024**3 | 1024 | 1024 | 1 | 0.645 | 2096 | 20.5 |
| FX1 (SMT) | 3Db | 1024**3 | 1024 | 1024 | 1 | 0.661 | 2043 | 20.0 |
| FX1 | 3Db | 1024**3 | 1024 | 258 | 4 | 0.732 | 1846 | 18.0 |
| FX1 (SMT) | 3Db | 1024**3 | 1024 | 258 | 4 | 0.783 | 1810 | 17.7 |
| FX1 | 3Db | 1024**3 | 2048 | 2048 | 1 | 0.355 | 3813 | 18.6 |
| FX1 | 3Db | 1024**3 | 2048 | 512 | 4 | 0.380 | 3555 | 17.4 |
| FX1 (SMT) | 3Db | 2048**3 | 1024 | 1024 | 1 | 5.09 | 2127 | 20.8 |
| FX1 | 3Db | 2048**3 | 2048 | 2048 | 1 | 2.72 | 3979 | 19.4 |
| FX1 | 3Db | 2048**3 | 2048 | 512 | 4 | 2.88 | 3752 | 18.3 |
| FX1 | 3Db | 3072* 2048**2 | 3072 | 3072 | 1 | 2.70 | 5997 | 19.5 |
| FX1 | 3Db | 3072* 2048**2 | 3072 | 768 | 4 | 2.92 | 5556 | 18.1 |

2Da：2次元領域分割、3Da：3次元領域分割、3Db：3次元領域分割＋f(m,i,j,k)
3Dbは計算効率が高い、またFlat MPIがHybrid（自動並列数=4）より僅かに高効率



## FX1の計算速度 3

| コンピュータ JAXAJAXA JAXA | プログラム | グリッド数 | 総コア数 | プロセス並列数 | 自動並列数 | CPU時間(sec) | 計算速度(Gflops) | 効率(%) |
|---|---|---|---|---|---|---|---|---|
| FX1 (SMT) | 3Db | 2048**3 | 256 | 256 | 1 | 19.8 | 541 | 21.3 |
| FX1 (SMT) | 3Db | 2048**3 | 1024 | 1024 | 1 | 5.09 | 2127 | 20.8 |
| FX1 | 3Db | 2048**3 | 2048 | 2048 | 1 | 2.72 | 3979 | 19.4 |
| FX1 | 3Db | 3072*2048**2 | 3072 | 3072 | 1 | 2.70 | 5997 | 19.5 |
| FX1 | 3Db | 2048**3 | 512 | 512 | 1 | 10.53 | 1028 | 20.1 |
| | 3Db | 1024**3 | | | | | | |
| JAXA FX1 | 3Db | 1024**3 | 1024 | 1024 | 1 | 0.701 | 1929 | 18.8 |
| JAXA FX1 | 3Db | 1024**3 | 4096 | 4096 | 1 | 0.220 | 6160 | 15.0 |
| JAXA FX1 | 3Db | 1024**3 | 4096 | 1024 | 4 | 0.211 | 6406 | 15.6 |
| JAXA FX1 | 3Db | 1000*1200**2 | 5760 | 5760 | 1 | 0.247 | 7338 | 12.7 |
| JAXA FX1 | 3Db | 1000*1200**2 | 5760 | 1440 | 4 | 0.228 | 7966 | 13.8 |

1Da：1次元領域分割、2Da：2次元領域分割、3Da：3次元領域分割、3Db：3次元領域分割＋f(m,i,j,k)
多コア数の計算も高効率、JAXA FX1では多コアでHybridがFlat MPIよりも高効率の場合もあった。

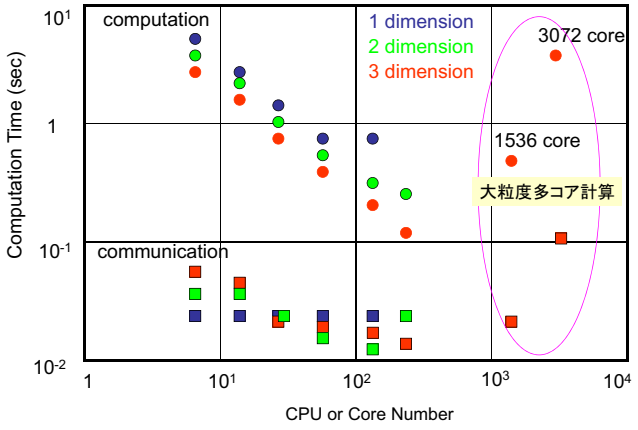


## HX600の計算速度

| コンピュータ | プログラム | グリッド数 | 総コア数 | プロセス並列数 | 自動並列数 | CPU時間(sec) | 計算速度(Gflops) | 効率(%) |
|---|---|---|---|---|---|---|---|---|
| HX600 | 2Da | 1024**3 | 256 | 256 | 1 | 4.95 | 273 | 10.7 |
| HX600 | 3Da | 1024**3 | 256 | 256 | 1 | 3.56 | 379 | 14.8 |
| HX600 | 3Db | 1024**3 | 256 | 256 | 1 | 5.22 | 259 | 10.1 |
| HX600 | 3Db | 1024**3 | 2048 | 2048 | 1 | 0.464 | 2912 | 15.9 |
| HX600 | 3Db | 1024**3 | 2048 | 512 | 4 | 0.685 | 1972 | 9.6 |
| HX600 | 3Db | 1024**3 | 2048 | 128 | 16 | 0.894 | 1511 | 7.4 |
| HX600 | 3Da | 1024**3 | 2048 | 2048 | 1 | 0.406 | 3327 | 16.3 |
| HX600 | 2Da | 1024**3 | 2048 | 2048 | 1 | 0.674 | 2005 | 9.8 |
| HX600 | 2Da | 2048**3 | 2048 | 2048 | 1 | 6.27 | 1724 | 8.4 |
| HX600 | 3Da | 2048**3 | 2048 | 2048 | 1 | 3.54 | 3047 | 14.9 |
| HX600 | 3Db | 2048**3 | 2048 | 2048 | 1 | 4.99 | 2217 | 10.6 |
| HX600 | 3Da | 2048**3 | 2048 | 128 | 16 | 3.54 | 151 | 1.5 |

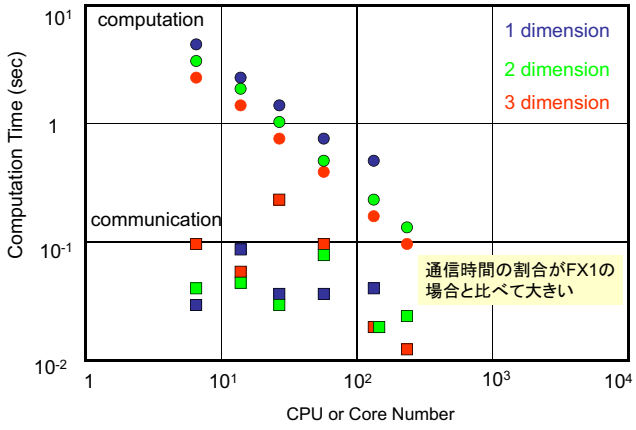
2Da：2次元領域分割、3Da：3次元領域分割、3Db：3次元領域分割＋f(m,i,j,k)
大粒度計算ではHX600はFX1より計算効率は低い、また、自動並列数を上げると計算効率がかなり低くなる。

# FX1 Flat MPI



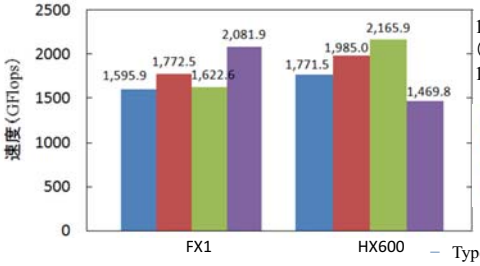



# HX600 Flat MPI



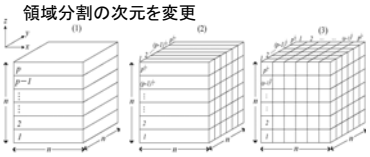



# MHDコードの性能評価




1024coreで計測
（理論性能10TFlops）
1coreあたり256MBに設定

領域分割の次元を変更



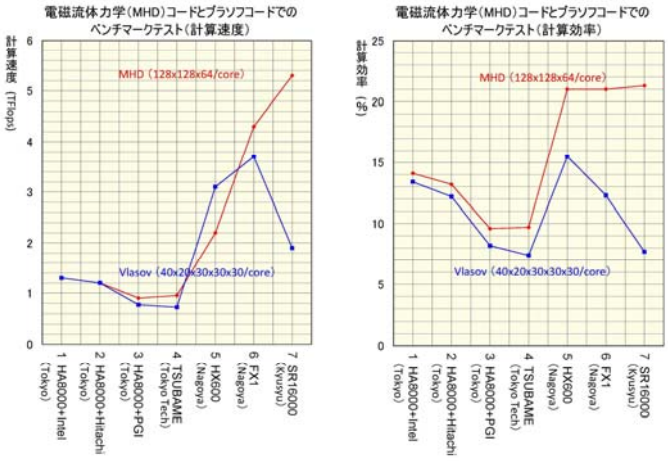
- Type A : f(i, j, k, m)
- Type B : f(m, i, j, k)

i, j, k が空間番号、m が変数番号
⇒同じ空間で変数に連続アクセス
HPC2500では有効だった

- FX1はHPC2500と同じ傾向
- HX600はSSEが有効
- 実効性能はどちらも約21%

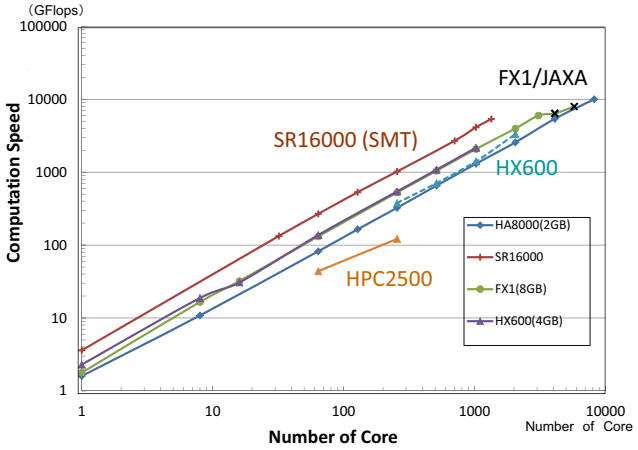


# MHDコードとVlasovコードでの性能評価






# 3D MHDコードによる計算速度の比較






# 名大スパコンのまとめ

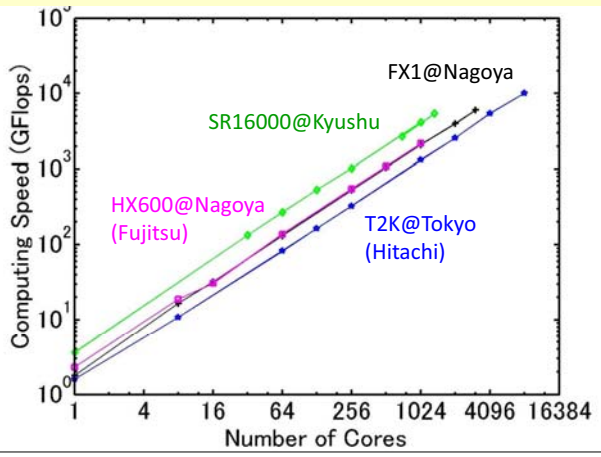
- 名古屋大学情報基盤センターの新スパコンシステムは実用の3次元MHDプログラムを用いて期待したとおりの性能がでている。

- 全国大学共同利用情報基盤センターで唯一のプリペタコンであるFX1はFlat MPIの2048 coreで最高速4.3 Tflops（絶対効率21%）を実現した。
  1024 coreでの最高速は2.2 Tflops（絶対効率22%）。
  mpifrt progmpi.f -o progmpi -Kprefetch_model=FX1 -Z mpilist
- FX1のハイブリッド並列（プロセス並列＋自動並列）は512 node 4 threadで4.1 Tflops（絶対効率20%）を得た。
  mpifrt progmpi.f -o progmpi -Kimpact -Kprefetch_model=FX1 -Z mpilist

- HX600のFlat MPIの1024 coreでは最高速2.2 Tflops（絶対効率21%）を得た。

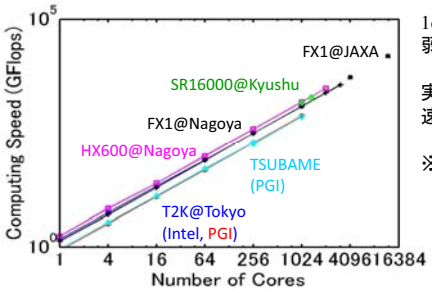
## 5．種々のスパコンの計算速度の比較と考察

**MHDコードのベンチマークテスト**






## ブラソフコード性能評価

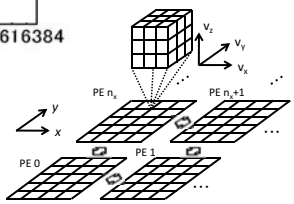



1coreあたり1Gに設定した弱いスケーリング

実空間(x,y,z)をMPI並列
速度空間をOpenMP並列

※4096と12032はJAXAで計測

- HX600で実効性能約15%
- FX1では実効性能約12%
  - T2K@東大とほぼ同速





## ◆ 種々のスーパーコンピュータの計算速度の比較

並列ベクトル計算機から超並列スカラ計算機まで
- ベクトル機：Fujitsu：VPP5000、NEC：SX-6、SX-8R、SX-9、ES
- スカラ機（x86除く）：Fujitsu：HPC2500、FX1、PRIMQUEST、Hitachi：SR16000
- X86機：Hitachi HA8000、Fujitsu HX600、Sun：TSUBAME、Fujitsu PRIMERGY

| | Core数/CPU数 | 実効性能(GFlops) | 理論性能(GFlops) | 実行効率(%) | 領域分割 | CPU種類 |
|---|---|---|---|---|---|---|
| SX-9 | 64/64 | 2058 | 6553 | 31 | 2次元 | ベクトル |
| HA8000 | 8192/1024 | 10038 | 75366 | 13 | 3次元A | Opteron |
| HX600 | 1024/256 | 2166 | 10240 | 21 | 3次元A | Opteron |
| FX1 | 1024/256 | 2081 | 10240 | 21 | 3次元B | SPARC64VII |
| SR16000 | 1344/672 | 5375 | 25267 | 21 | 3次元B | POWER6 |

| | SX-9 | HA8000 | HX600 | FX1 | SR16000 |
|---|---|---|---|---|---|
| @core | 1.0000 | 0.0400 | 0.0658 | 0.0632 | 0.1244 |
| @CPU | 1.0000 | 0.1595 | 0.2631 | 0.2528 | 0.2487 |



## 新しい技術について

SMT（Simultaneous Multithreading）は有効か？
Core内で複数の実行スレッドを同時に実行するプロセッサの機能
Node – Multi-Core – Multi-Thread

日立SR16000ではSMT効果が顕著に現れる
富士通FX1ではSMT効果は微少

SMTやコア内複数演算機がいつも機能するかは疑問



## 99．99%を超える並列化率と20%程度以上の絶対効率を得るためのヒント

1. プログラムのメイン部分で全てのdo loopを並列化（1個でも残したらダメ）
2. 全てのdo loopで並列化OKの表示が出ても安心してはいけない（if文や条件文は低効率の並列化をやっただけかも知れない）ほとんどの場合数倍程度
3. if文や条件文（**dependency**）を並列化するためには、**Mask, List vector, Gather and scatter**の中から最も適切なものを最も適切な方法で使う必要がある
4. 実測する場合には粒度（1個のCPUのメモリ占有率）を揃える必要がある
5. 最善のテストは**最大数のCPUを用いて、粒度を意識して絶対効率を実測**すること



## 今後解明すべきこと

1. Core数を1-10万個と更に増やした場合もscalabilityは保たれるか？（どうしたら予測できるか？）
2. 通信は本当にボトルネックになるのか？この限界を明確に示すにはどうすればよいのか？
3. Cacheのヒット率を更に上げるにはどうするのか？
4. Flat MPIとHybrid（プロセス並列＋自動並列）はCore数が増えた時、どちらが高効率なのか？
5. スパコンの種類によって高効率のプログラムは異なるのか（FX1型とT2K型）？

## 6. 地球磁気圏の３次元MHDとVlasovシミュレーションの将来

・どのシステムにおいてもほぼ線形なスケーラビリティ
・どのシステムにおいても、実効効率は10 - 20%以上

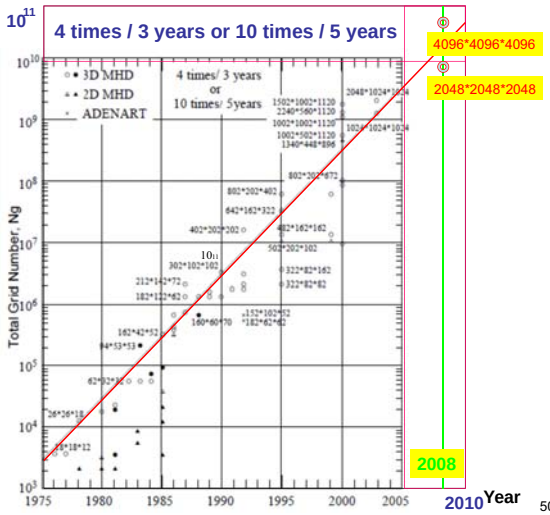
・今後は、128 – 1024 コアの並列計算により、スケール間結合の研究を推進

☆高解像度磁気圏グローバルMHDシミュレーション
☆中間スケール規模の第一原理ブラソフシミュレーション



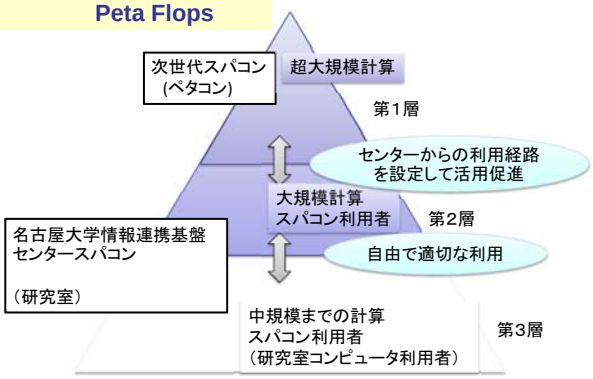
2D and 3D MHD Simulation

3年で4倍、5年で10倍程度の割合で計算規模が増大






ペタコン：約10万CPU以上
Peta Flops




名古屋大学情報連携基盤センターのスパコン利用者からみた計算機資源の階層的配置



### 数値計算・シミュレーションの計算規模

第1層　超大規模計算：情報連携基盤センターのスパコンを越えての利用
（cpu数1000程度を超えて）、ペタコン利用が必要
第2層　大規模計算：情報連携基盤センターのスパコンを大規模に利用
（cpu数1000程度まで）
第3層　中規模までの計算：各グループ所有のクラスター計算機でも計算可能
（cpu数100程度まで）

### ペタコン利用の研究計画の具体例（名古屋大学）

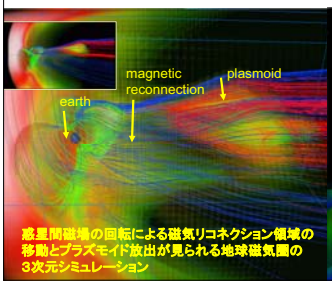
1：一様非等方MHD乱流の減衰法則
2：平行平板間乱流の超大規模直接数値計算
3：タンパク質のシミュレーション
4：ジオスペースにおける領域間結合
5：無衝突衝撃波による粒子加速
6：ブラソフシミュレーションで見る惑星磁気圏の3次元運動論的描像
7：分子動力学法による液膜破壊現象の直接計算
8：大規模非線形固有値問題の解法とその応用
9：生体分子の立体構造予測シミュレーション
10：強相関多体電子系シミュレーション
11：雲のバーチャルリアリティ
12：自己組織化マップを用いた気候変動の評価

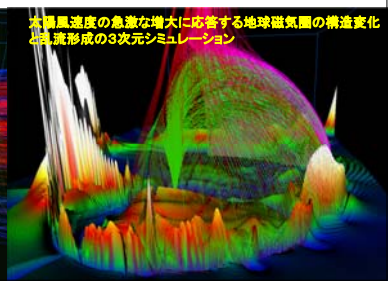
4－6：
太陽地球環境研究所の研究計画



### 大規模高精度計算

1. 研究課題：ジオスペースにおける領域間結合（磁気圏の境界層乱流と構造形成を解明）
2. 研究の概要：
・太陽地球系の境界層プラズマ不安定の成長による構造形成とエネルギー・物質輸送
・プラズマ乱流を介してのマクロとメソスケール結合の非線形物理の解明
3. 計算の概要：MHD、拡張MHD 及び流体モデルによるグローバルシミュレーション
4. 現在までの到達度：3次元MHD コードで20%以上の高計算効率（1024 cpu）
5. ペタコンで狙うものとその計算規模
・4096 の3乗の格子点を用いて現在のHPC2500 で10,000 時間のオーダーの計算量
・多流体、Hall MHD 、及びプラズマと中性流体の結合への拡張の必要性




## 7. まとめ

スーパースカラー並列機、スーパーコンピュータで最高速の並列計算プログラムを作成するために

1. プログラムの構造を確立することが最も重要（Dependencyのない構造）
2. キャッシュのヒット率を最大化
3. 通信量の最小化と一括転送

これらを実現して、種々のスパコンにおいて、実効効率は10 - 20%以上を達成

4. プログラムサイズの最小化は今後の目標
（1次元化で領域分割は使用不可、実現可能か？）

## FX1とHX600計算速度

| コンピュータ | プログラム | グリッド数 | 総コア数 | プロセス並列数 | 自動並列数 | CPU時間(sec) | 計算速度(Gflops) | 効率(%) |
|---|---|---|---|---|---|---|---|---|
| FX1 | 1Da | 1024**3 | 256 | 64 | 4 | 152.0 | 9 | 0.4 |
| FX1 | 1Da | 1024**3 | 128 | 128 | 1 | 232.2 | 6 | 0.5 |
| FX1 | 1Da | 1124*1024**2 | 128 | 128 | 1 | 28.4 | 52 | 4.1 |
| FX1 | 1Da | 602*402*402 | 256 | 64 | 4 | 0.417 | 294 | 11.5 |
| FX1 | 1Da | 902*402*402 | 256 | 64 | 4 | 0.704 | 261 | 10.2 |
| FX1 | 1Da | 1124*1024**2 | 128 | 128 | 1 | 8.23 | 180 | 14.1 |
| FX1 | 3Da | 1024**3 | 128 | 128 | 1 | 7.32 | 185 | 14.4 |
| HX600 | 1Da | 512**3 | 64 | 64 | 1 | 4.06 | 42 | 6.5 |
| HX600 | 1Da | 1024**3 | 256 | 64 | 4 | 9.81 | 138 | 5.4 |

1Da：1次元領域分割＋f(i,j,k,m)　m：MHD方程式の8個成分
2Da：2次元領域分割＋f(i,j,k,m)
3Da：3次元領域分割＋f(i,j,k,m) 3Db：３次元領域分割＋f(m,i,j,k)
**1次元領域分割も配列数を適切に選べばかなりの計算速度が得られる。**



## FX1の計算速度：１次元コード：f(i1)

・1次元配列f(i1)MHDコードはメモリを最小化したプログラム
・領域分割法を利用できないので新しいユーザー（プログラム）分割法の開発が必要

| コンピュータ | グリッド数 | 総コア数 | プロセス並列数 | 自動並列数 | CPU時間(sec) | 計算速度(Gflops) | 効率(%) |
|---|---|---|---|---|---|---|---|
| FX1 | 62*32*32 | 4 | 1 | 4 | 0.0227 | 3.5 | 8.8 |
| FX1 | 602*302*302 | 4 | 1 | 4 | 14.4 | 4.8 | 12.0 |
| FX1 | 802*402*402 | 4 | 1 | 4 | 32.6 | 5.0 | 12.5 |
| FX1 | 1002*502*502 | 4 | 1 | 4 | 65.7 | 4.8 | 12.1 |
| FX1 | 50*50*26 | 4 | 1 | 4 | 0.0238 | 3.4 | 8.6 |
| FX1 | 402*402*202 | 4 | 1 | 4 | 8.78 | 4.7 | 11.7 |
| FX1 | 602*602*302 | 4 | 1 | 4 | 28.9 | 4.8 | 11.9 |
| FX1 | 802*802*402 | 4 | 1 | 4 | 66.7 | 4.9 | 12.2 |

f(i1): i1=i+n1*(j-1)+n2*(k-1)+n3*(m-1), n1=nx2, n2=n1*ny2, n3=n2*nz2
f(i1)=f(i,j,k,m), f(n4)=f(nx2,ny2,nz2,nb), n4=n3*nb