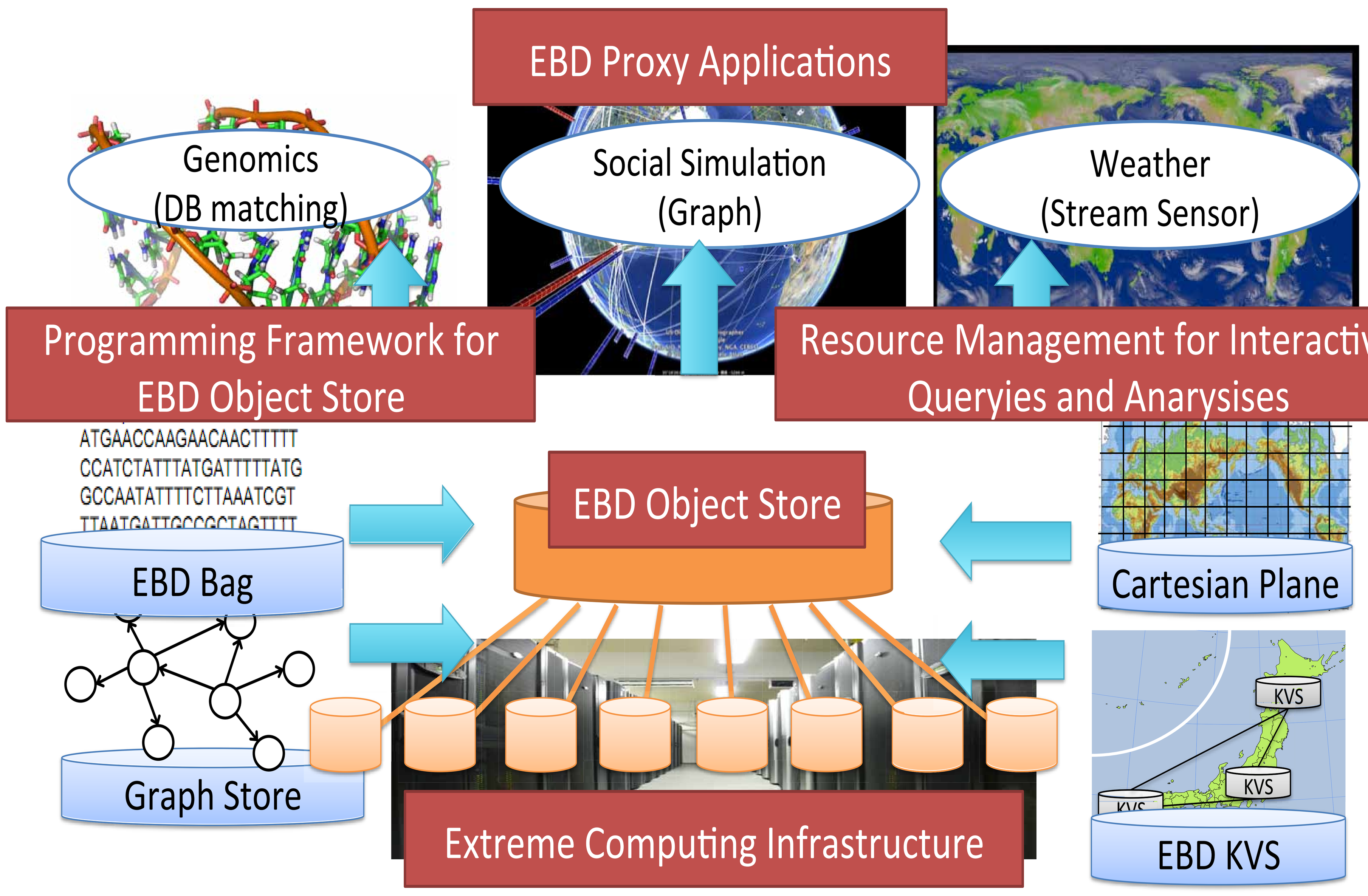
Extreme Big Data

Next Generation Big Data Infrastructure Technologies Towards Yottabyte/Year

PI: Satoshi Matsuoka (matsu@is.titech.ac.jp), supported by JST-CREST

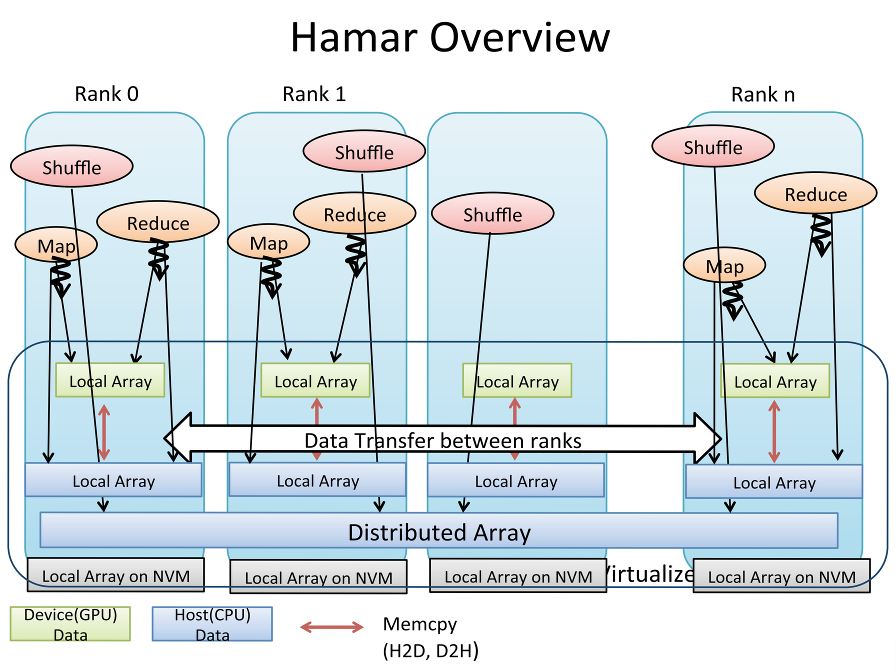
Extreme Big Data (EBD) Overview

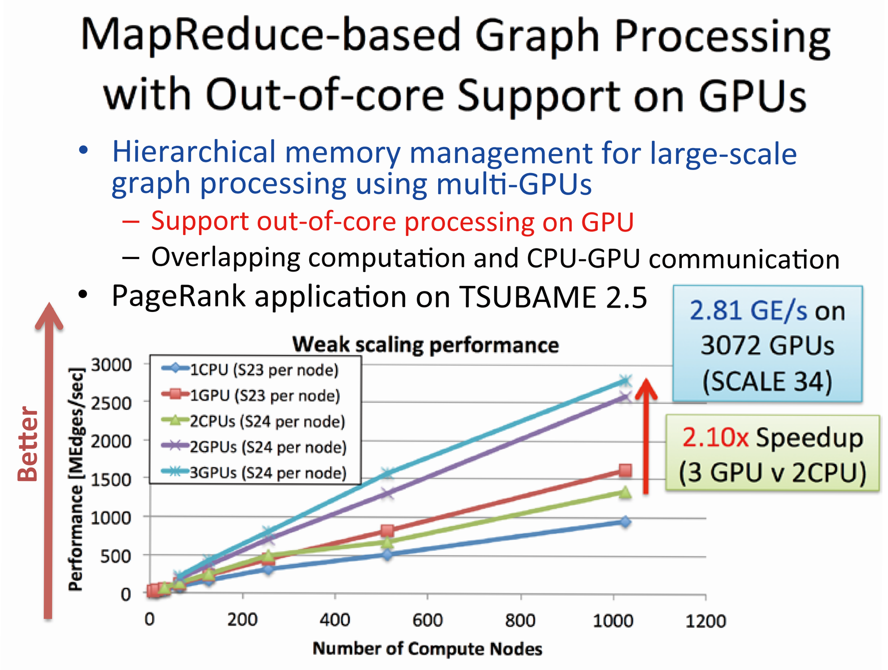
Our project, called EBD, aims to achieve the convergence of extreme supercomputing and big data in order to cope with explosion of data from multiple sources such as massive numbers of sensors whose resolution is increasing exponentially, high resolution simulations generating huge data results, as well as evolution of social infrastructures that allow for "opening up of data silos", i.e., data sources being confined within an institution, much as how scientific data are being handled in the modern era as common asset openly accessible within and across disciplines. Our primary target proxy applications include metagenomics, social simulation, and climate simulation with real-time data assimilation. Based on these EBD co-design applications, we define future EBD convergent SW/HW architecture and system. We have several on-going collaboration work with RIKEN AICS, ORNL, LLNL, ETH and JST Graph CREST / Univ. Kyushu.



GPU-based MapReduce

HAMAR is a MapReduce-style programming for next-gen supercomputers with many-core accelerators and non-volatile memory devices. Our framework handles memory overflow from GPUs by dividing data into multiple chunks and overlaps CPU-GPU data transfer and computation on GPUs as much as possible.[1]



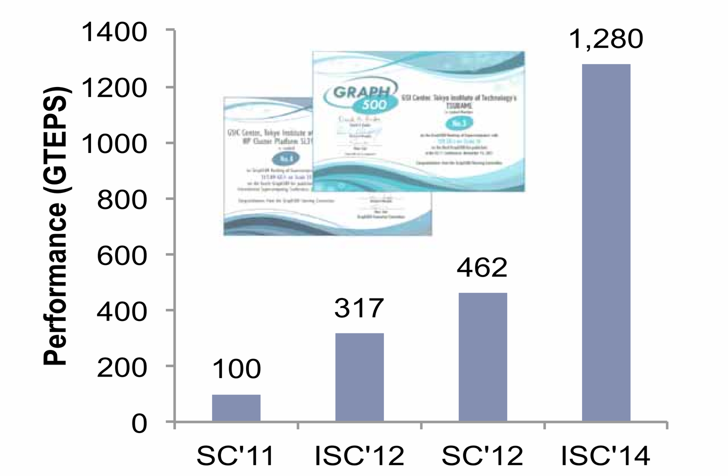


[1] K. Shirahata, H.Sato, S. Matsuoka, "Out-of-core GPU Memory Management for MapReduce-based Large-scale Graph Processing", IEEE Cluster2014.

Graph500

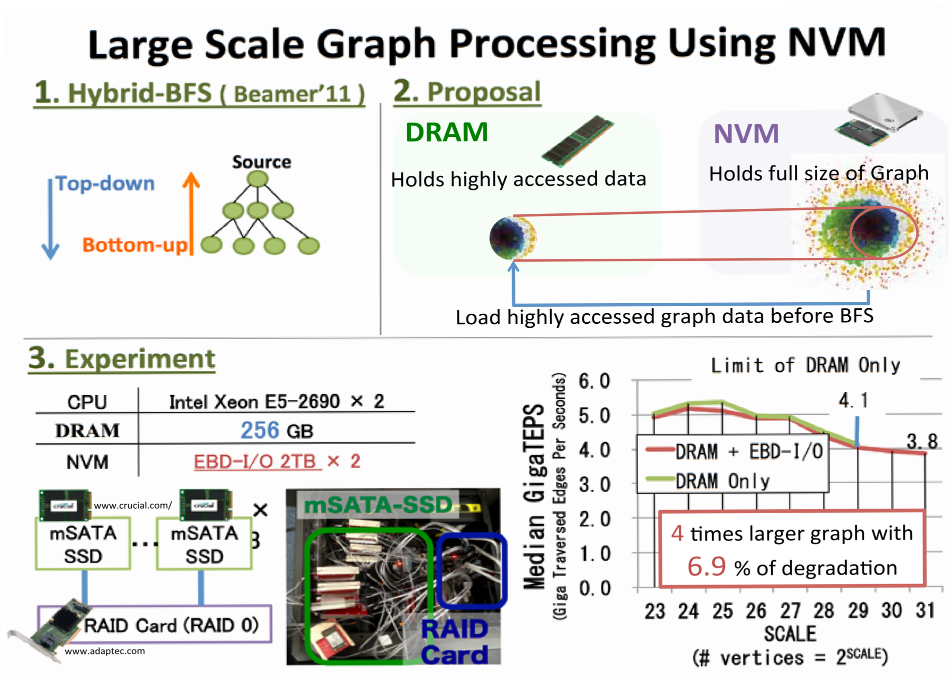
We have developed extremely fast breadth first search (BFS) implementations for large-scale distributed environments and NVM-based hierarchical memory machines[2]. We have achieved several notable results on the Green500 and the Green Graph500, including becoming world #1 on the Graph500 (June 2014)[3] on K Computer and #1 on the Green Graph500 (November 2013)[4] on TSUBAME-KFC, based on our implementations.

History of Graph500 scores on TSUBAME2



Optimization techniques

Optimizations	SC11	ISC12	SC12	ISC14
2D decomposition	✓	✓	✓	✓
vertex sorting	✓			
direction optimization				✓
data compression	✓	✓	✓	
sparse vector with pop counting				✓
adaptive data representation				✓
overlapped communication	✓	✓	✓	✓
shared memory				✓
GPGPU		✓	✓	

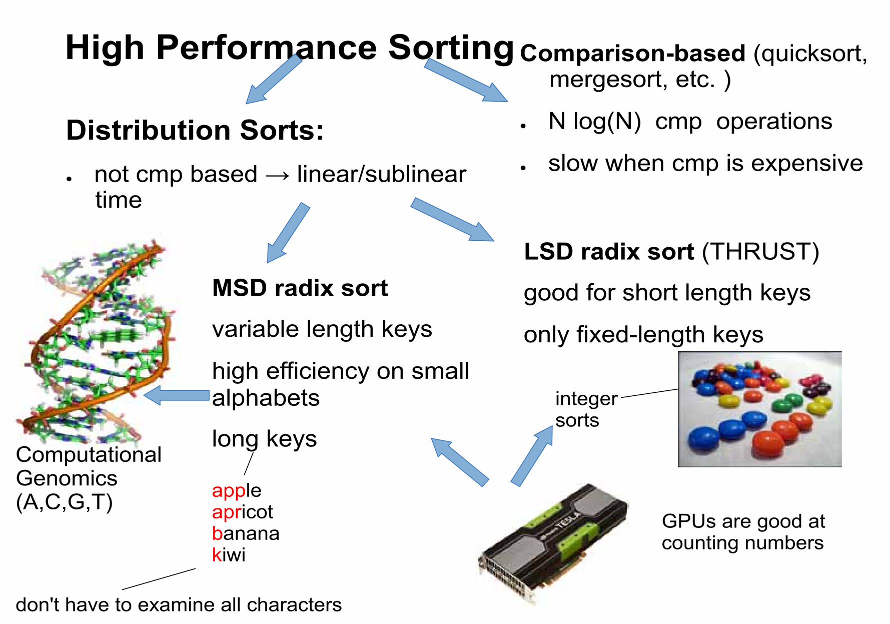


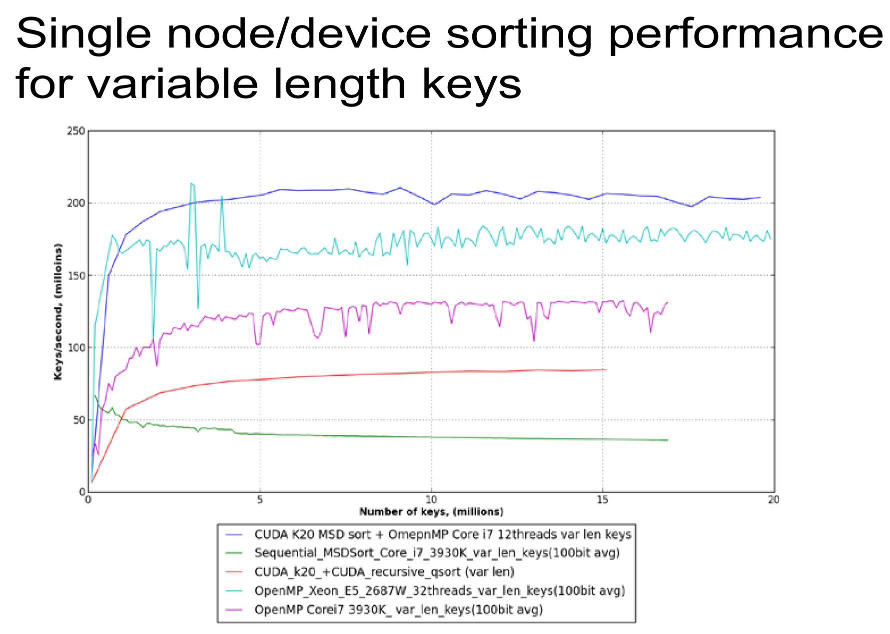
[2] K. Iwabuchi, H. Sato, Y. Yasui, K. Fujisawa, S. Matsuoka,"NVM-based Hybrid BFS with Memory Efficient Data Structure", IEEE BigData 2014.
[3] Graph500, http://www.graph500.org.
[4] Green Graph500, http://green.graph500.org.

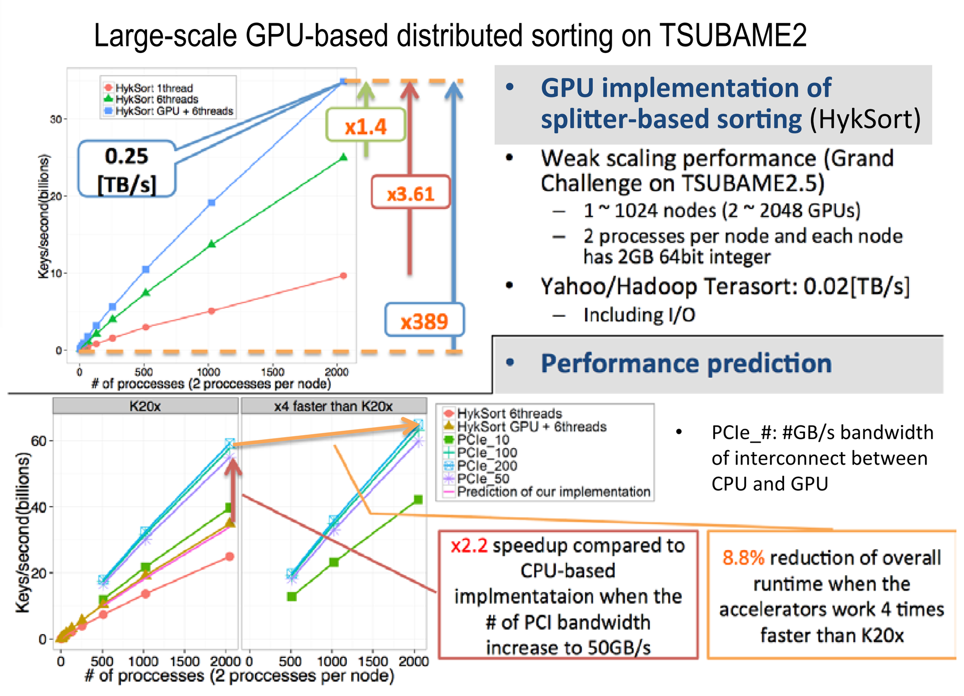
Collaboration work with JST Graph CREST project and RIKEN AICS

GPU Sorting

We have developed several GPU-based sorting implementations, including support for variable length keys[5], large-scale distributed environments[6], and out-of-core GPU memory management[1].







[5] A. Drozd, M. Pericas, S. Matsuoka, "Efficient String Sorting on Multi- and Many-Core Architectures", BigData Congress 2014.
[6] H. Shamoto, K. Shirahata, A. Drozd, H. Sato, S. Matsuoka,"Large-scale Distributed Sorting for GPU-based Heterogeneous Supercomputers", IEEE BigData 2014.