MAESTRO
DATA ORCHESTRATION

https://www.maestro-data.eu/

Towards Enabling Memory- and Data-Aware HPC

Dirk Pleiter

Reading, 25.09.2018

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 801101

Outline

- Motivation
- Project
- Shortcomings and alternative concepts
- Applications
- Solution strategies
- Summary

Motivation

Memory Technology Trends

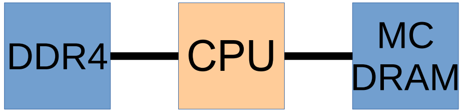
- Desirable memory performance features
  - Large memory capacity C_{mem}
  - High memory bandwidth B_{mem}
- Different memory technologies being integrated into HPC systems
  - DDR DIMMs: DDR3, DDR4, ...
  - High-bandwidth memory technologies: HBM, HMC/MCDRAM
  - Non-volatile memory technologies: NAND Flash, 3D-Xpoint
- Significant differences in terms o

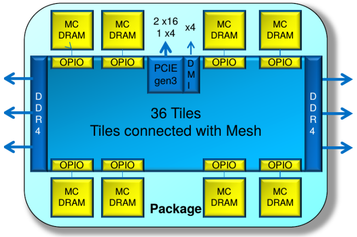
$$\Delta\tau = \frac{C_{\text{mem}}}{B_{\text{mem}}}$$

Hierarchical Memory Architectures

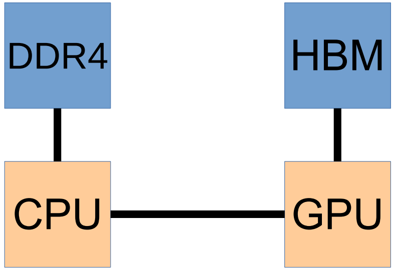
MAESTRO
DATA ORCHESTRATION

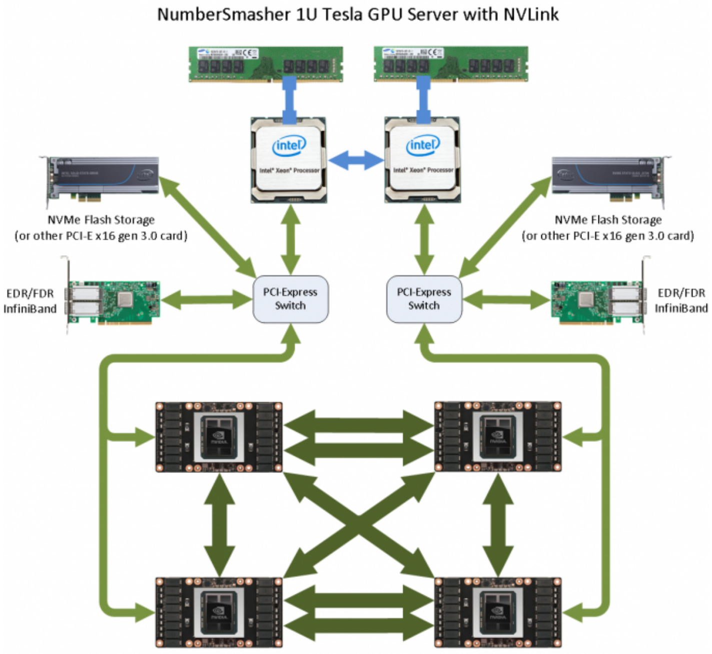
- Knights Landing





- GPU accelerators




NumberSmasher 1U Tesla GPU Server with NVLink
NVMe Flash Storage
(or other PCI-E x16 gen 3.0 card)
NVMe Flash Storage
(or other PCI-E x16 gen 3.0 card)
EDR/FDR
InfiniBand
PCI-Express
Switch
PCI-Express
Switch
EDR/FDR
InfiniBand

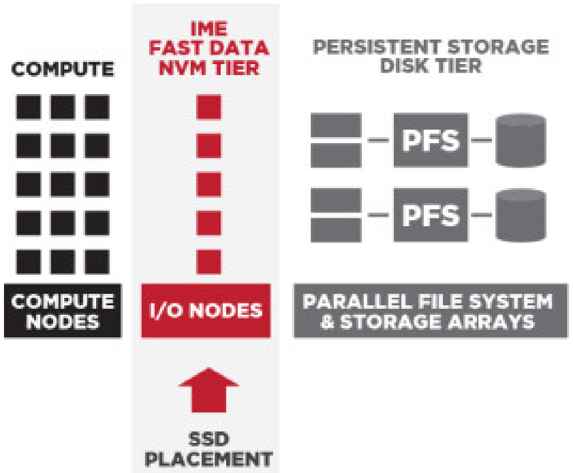
Hierarchical Storage Architectures

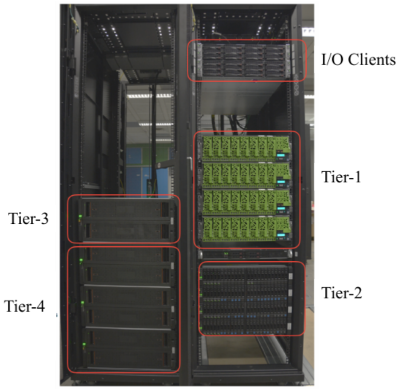
MAESTRO DATA ORCHESTRATION

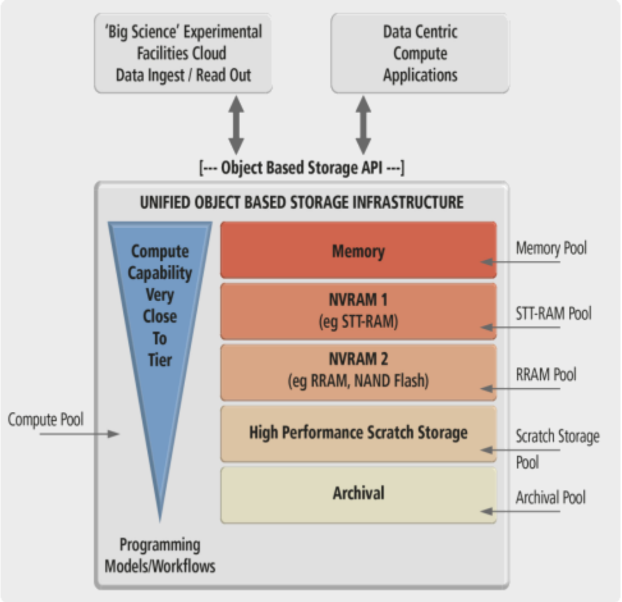
- Burst buffers
  - Example: DDN IME
- SAGE hierarchical object store

SAGE

http://www.sagestorage.eu/






'Big Science' Experimental Facilities Cloud Data Ingest / Read Out
Data Centric Compute Applications
[--- Object Based Storage API ---]
UNIFIED OBJECT BASED STORAGE INFRASTRUCTURE
Compute Capability Very Close To Tier
Memory
NVRAM 1 (eg STT-RAM)
NVRAM 2 (eg RRAM, NAND Flash)
High Performance Scratch Storage
Archival
Memory Pool
STT-RAM Pool
RRAM Pool
Scratch Storage Pool
Archival Pool
Compute Pool
Programming Models/Workflows

JÜLICH Forschungszentrum CRAY ECMWF SEAGATE CSCS cea appentra make code parallel

Application Requirements

- Increasingly complex workflows
  - Coupled applications
  - Human-in-the-loop
- Vast increase of data volumes
  - High data rate + large data volumes

Project

Consortium

- Industrial partners
  - Cray (Switzerland), Seagate (UK)
- Research organisations / supercomputing centres
  - CEA (France), CSCS (Switzerland), ECMWF (international), JSC (Germany)
- SME
  - appentra (Spain)

Goals

- Develop a middleware providing consistent data semantics to multiple layers of the stack
- Demonstrate progress for applications through memory- and data-aware (MADA) orchestration
- Enable and demonstrate next-generation systems software MADA features
- Improve the ease-of-use of complex memory and storage hierarchy

Shortcomings and Alternative Concepts

Lacking Data Awareness

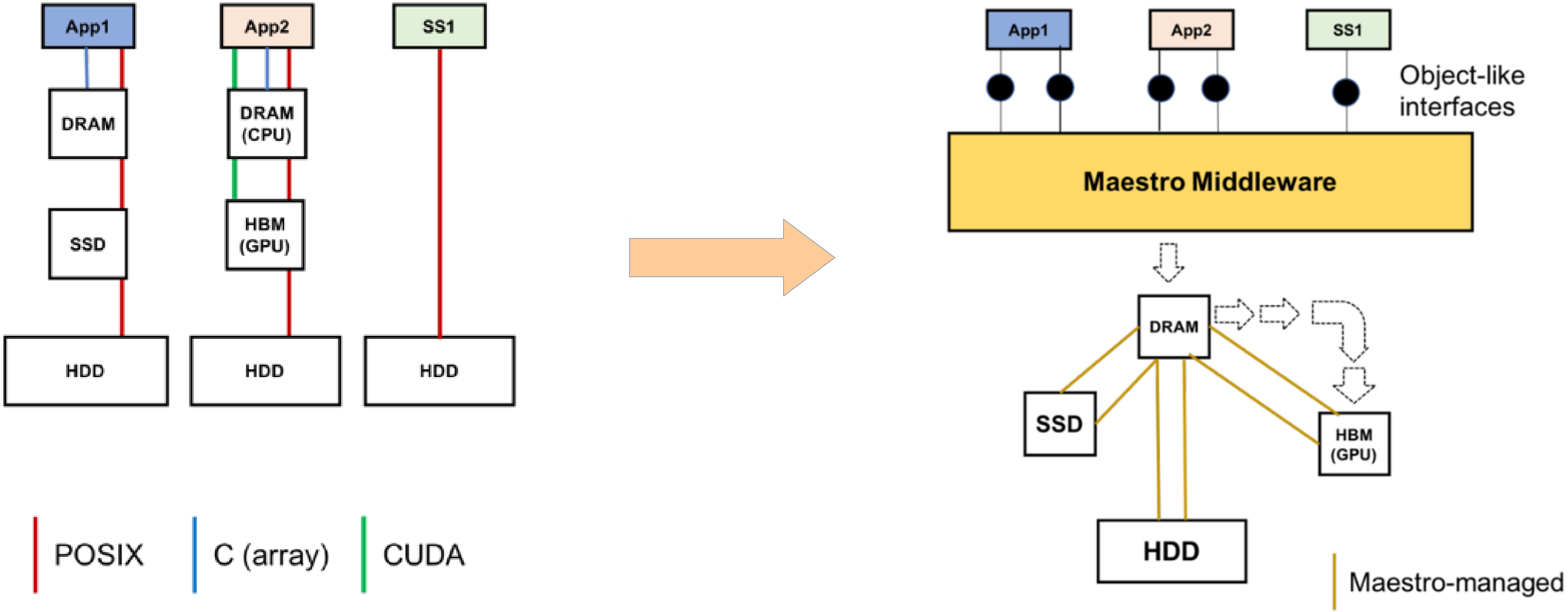
- Software stacks focussing on data processing
  - Optimised for filling of processing pipelines
  - Provide means for leveraging parallelism
- Lacking focus on basic data handling
  - Lacking functionality for controlling data handling
  - Lacking (unified) semantics for guiding data transport

Lacking Memory Awareness

- Missing information about available memory/storage hardware and its characteristics
  - Lacking ability for making data transport decisions
  - Missing information leads to hardware-neutral decisions
- Challenging variety of data access methods
  - Example storage class memory:
    Block store, file system, object storage

Maestro Vision

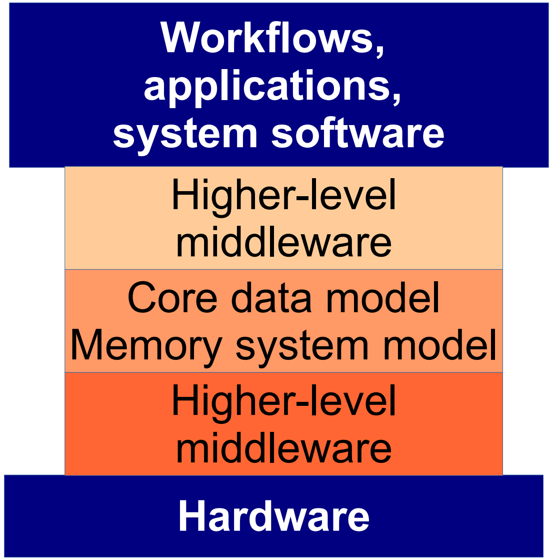
MAESTRO
DATA ORCHESTRATION



JÜLICH Forschungszentrum CRAY ECMWF SEAGATE CSCS cea appentra make code parallel

Concept

- Core data model
  - Provide data object semantics consistent at different layers
- Memory system model
  - Provide locality, access and performance information
- Higher-level middleware providing
  - Data access and adaptive transport capabilities
  - Workflow tools



Workflows, applications, system software

Higher-level middleware

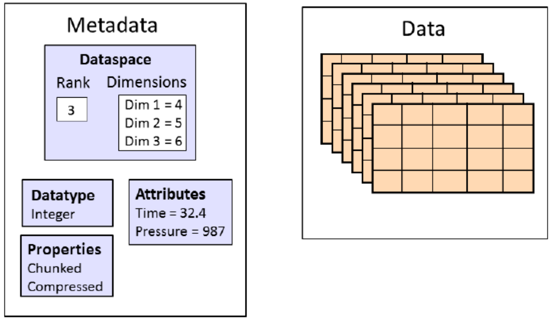
Core data model
Memory system model

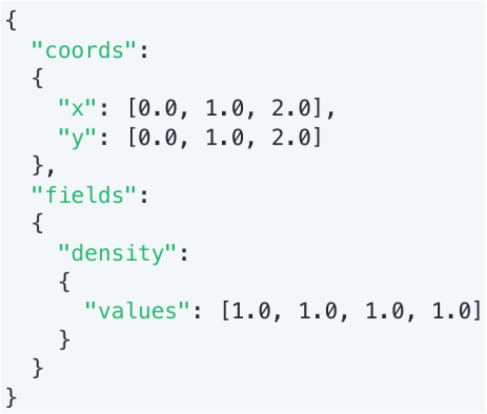
Higher-level middleware

Hardware

Existing Data Model Examples

- HDF5
  - Object oriented
  - Concept of data sets comprising data and metadata
- Conduit
  - Designed for exchanging data in HPC simulations (used, e.g., in VisIt)
  - API that allows to describe hierarchical data



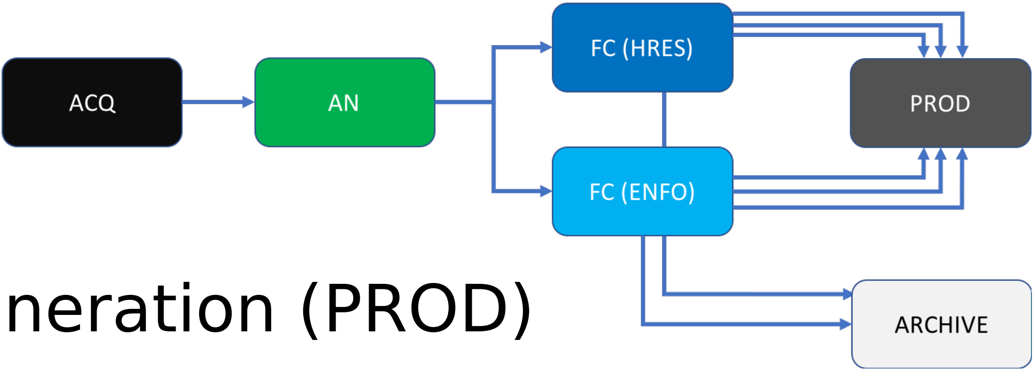


```
{
  "coords":
  {
    "x": [0.0, 1.0, 2.0],
    "y": [0.0, 1.0, 2.0]
  },
  "fields":
  {
    "density":
    {
      "values": [1.0, 1.0, 1.0, 1.0]
    }
  }
}
```

Applications

Numerical Weather Prediction

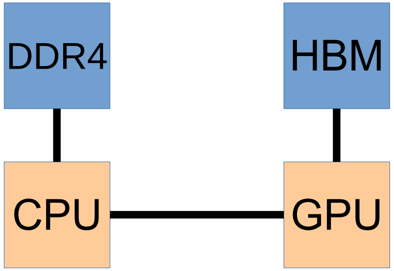
- Today’s bottlenecks
  - Data movement between forecast (FC) stages and product generation (PROD)
  - Irregular archiving of output from research workflows



- Solution strategy: Enable middleware avoiding multiple transfers based on suitable data object semantics

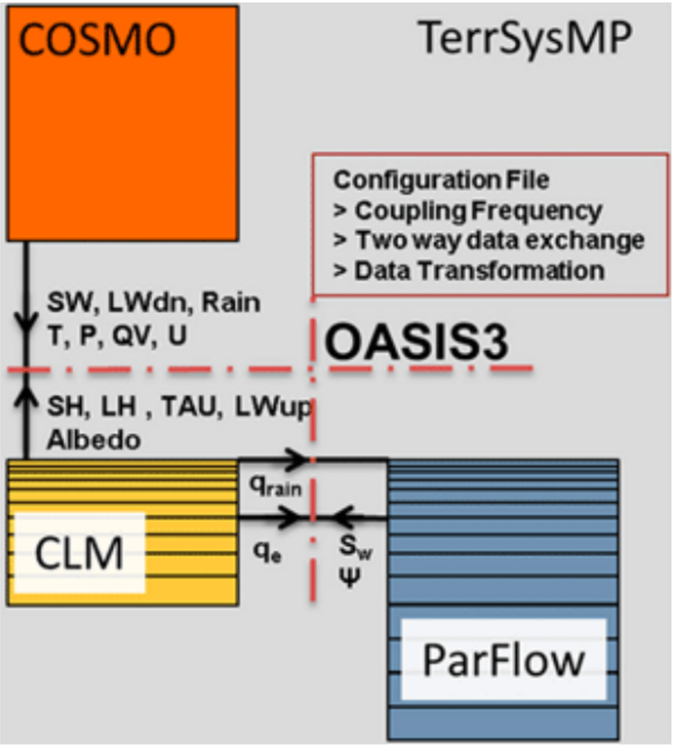
Materials Science

- Reference code: SIRIUS
  - Library providing building blocks for electronic structure applications
  - Written for GPU acceleration
- Aim for improved management of host and device memory resources
  - Awareness of data locality
  - Facilitate simpler data access and transport
    - Example: Processing of data objects that exceed device memory capacity



ESM Workflows

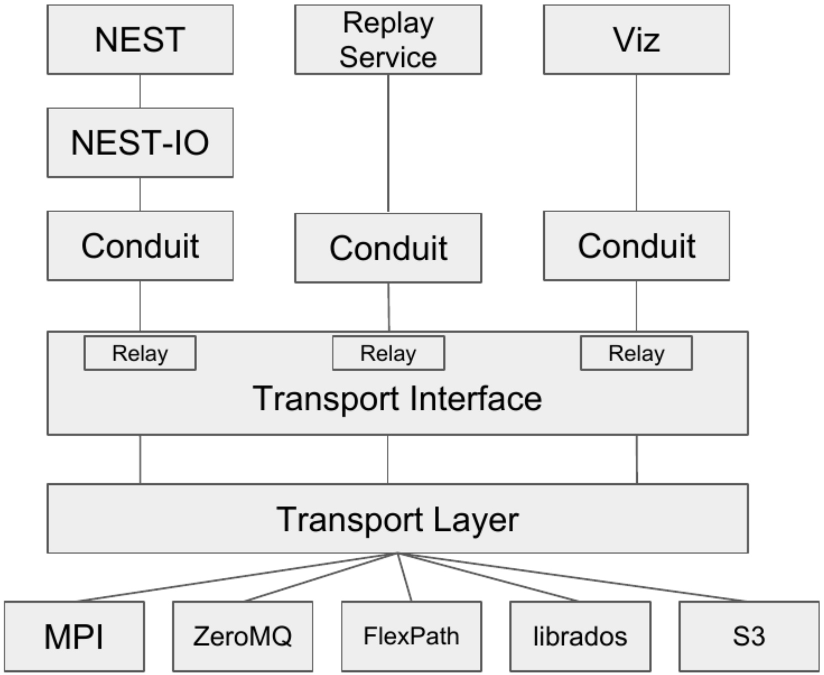
- TerrSysMP coupled workflow
  - Different workflow components coupled through OASIS3
  - Exchange of 2-/3-dimensional data structures
- Limitations of coupler
  - MPI based
  - Restrictive data semantics
- Similar problem class: Coupling to viz



Target Solutions

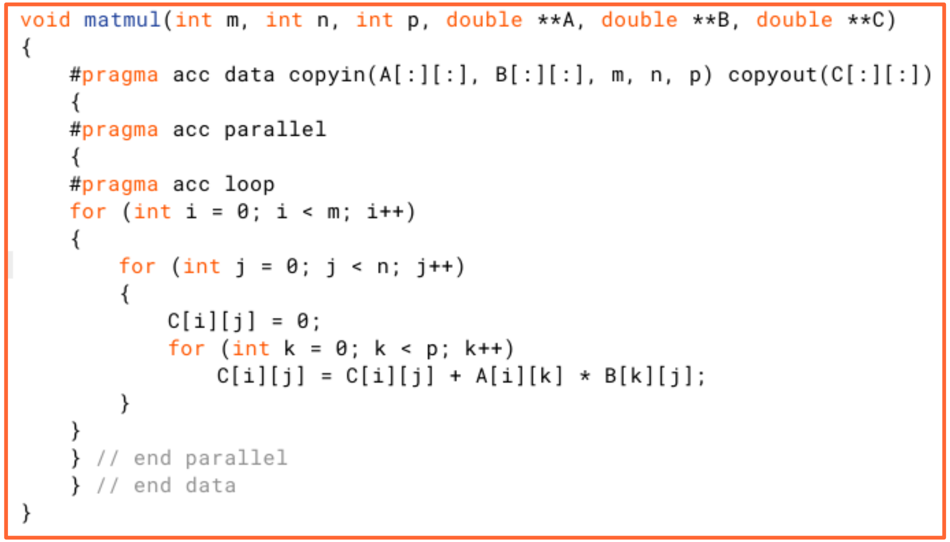
Coupling Simulation and Viz

- Brain modelling needs
  - Monitor simulation
  - Steer simulation
- Challenges
  - Proprietary data models/ coupling APIs
  - Restrictive options for data transport



Data-Aware Code Analyser

- Existing parallelware analyser
  - Identification of parallel patterns
  - Can, e.g., be used for proposing parallelisation/acceleration directives
- Maestro aims for (semi)automatic identification of
  - In-memory data layout
  - Data access patterns



```
void matmul(int m, int n, int p, double **A, double **B, double **C)
{
    #pragma acc data copyin(A[:][:], B[:][:], m, n, p) copyout(C[:][:])
    {
    #pragma acc parallel
    {
    #pragma acc loop
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            C[i][j] = 0;
            for (int k = 0; k < p; k++)
                C[i][j] = C[i][j] + A[i][k] * B[k][j];
        }
    }
    } // end parallel
    } // end data
}
```

Summary

Summary

- Lacking data- and memory-awareness
- Maestro wants to overcome this limitation by
  - Defining data object models with a rich semantics
  - Provide an object-like interface to existing data
  - Allow applications to give (or take back) control of data handling to Maestro
- Strongly application driven approach
  - Crucial for getting design and scope right
  - Needed for being able to demonstrate benefits
  - Ambition to provide benefits to key research fields