Status Report on MANA-2.0: A Future-Proof Design for Long-Running MPI-based Simulations for HPC

Gene Cooperman*
gene@ccs.neu.edu

Khoury College of Computer Sciences
Northeastern University, Boston, USA

HPC User Forum Spring 2022

*Partially supported by NSF Grant OAC-1740218, and by a grant from Intel Corporation.

MANA: Past, Present and Future

MANA: *MPI-Agnostic Network-Agnostic* checkpointing.

Past: Showed feasibility in academic prototype (Garg et al., HPDC'19)

Present: Three-way collaboration: NERSC, MemVerge, and DMTCP
(fully functional free and open source version,
while **MemVerge, Inc.** provides paid commercial support)

Near Future (medium-scale checkpointing):

1. **Robust, efficient transparent checkpointing at medium scale HPC**
2. **Killer App: Chaining of allocations for long-running computations (similar to spot instances in the Cloud)**
3. **Validate a subset of applications for MANA-compatible *system-level* checkpointing — support on-demand, real-time computing**

Extreme scale computing:

- **MANA-style split processes as one component in a larger ecosphere**

MANA and the HPC Ecosphere

We are not arguing for transparent checkpointing (MANA or otherwise) as the ultimate solution to *extreme scale* checkpointing. Rather, it is one component in a larger ecosphere.

In many extreme scale proposals for fault tolerance, if you look closely at the "fine print", you will often find:

> *If a component fails, then we resort to checkpointing each sub-computation, and restoring from a checkpoint.*

Checkpointing of that sub-computation may be done either:

- by transparent checkpointing; or
- by application-specific checkpointing with the help of a checkpointing library (for example, VeloC); *(But beware of closed-source vendor code.)*

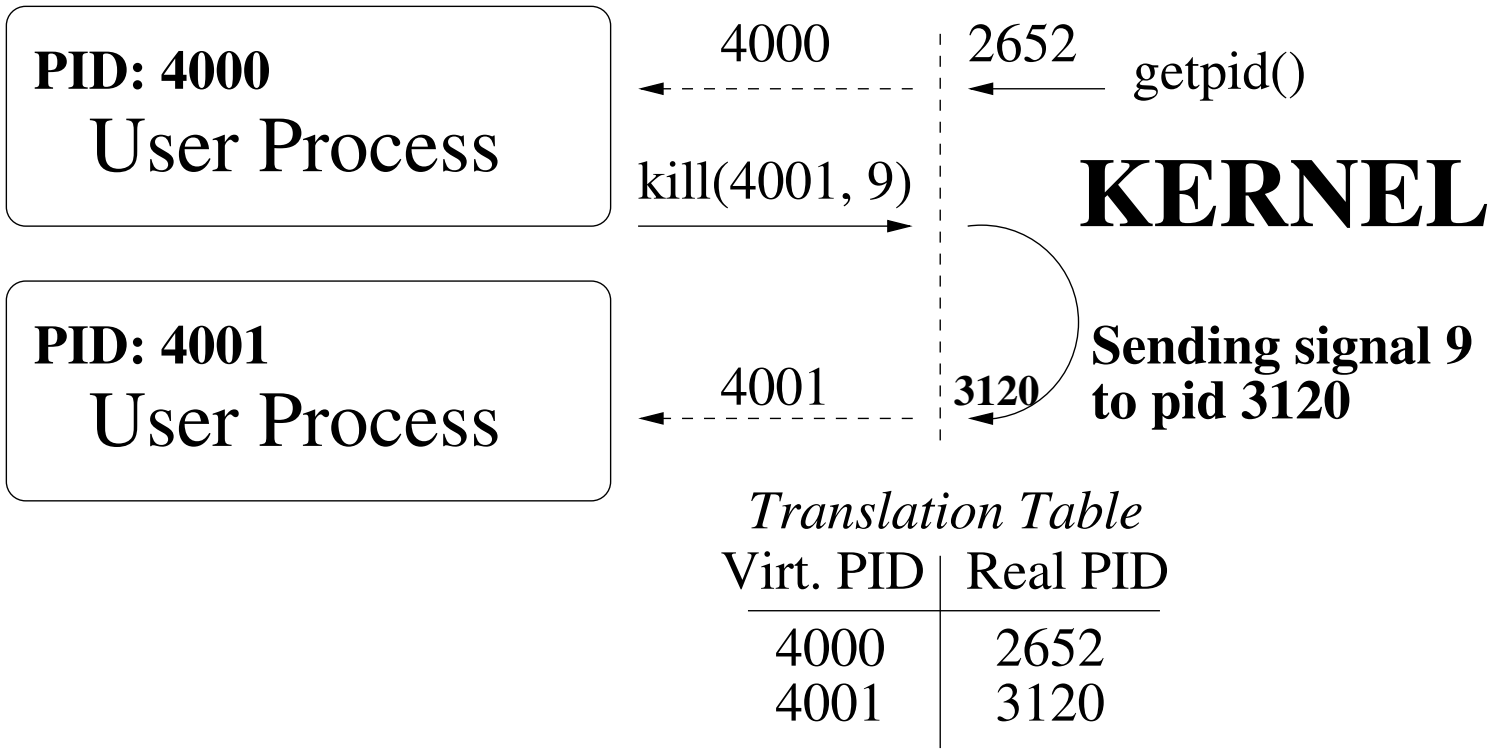
GOAL of Checkpointing: system-aware; compatible with other subsystems.

Solution: Customize library; or create custom plugin for transparent ckpt.

> *Example: A small DMTCP plugin was built five years ago in collaboration with Franck Cappello, demonstrating support for the API of the VeloC library (including incremental checkpointing).*

A Simple DMTCP Plugin: Virtualizing the Process Id

- **PRINCIPLE:**
  The user sees only virtual pids; The kernel sees only real pids



Translation Table

Virt. PID	Real PID
4000	2652
4001	3120

Other possible plugins: key-value database; selective saving of memory; for HPC-aware Containers (e.g., Apptainer/Singularity)

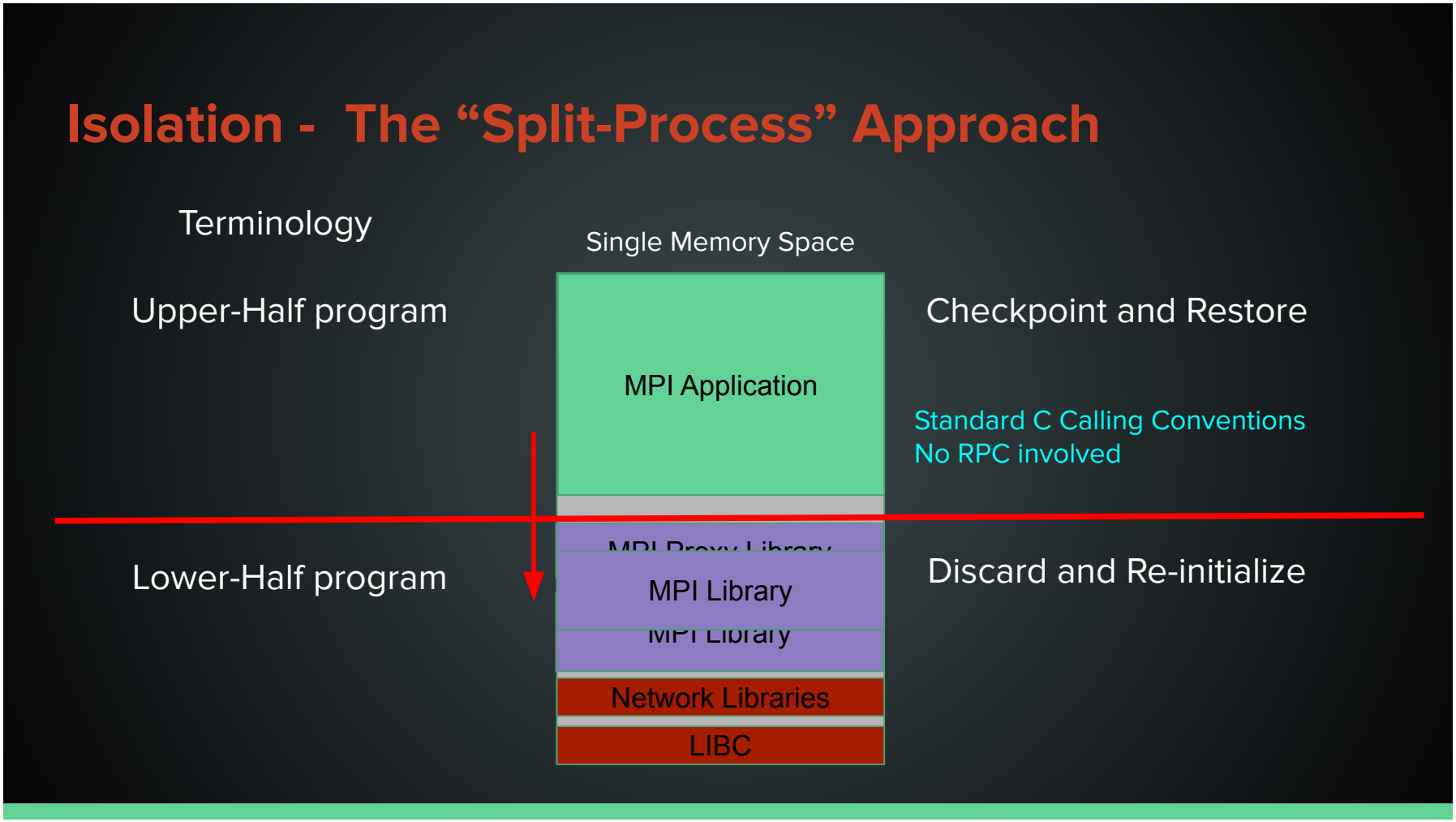
MANA (Past): History and Origins in Thesis of Rohan Garg

MANA: *MPI-Agnostic Network-Agnostic* checkpointing.

---

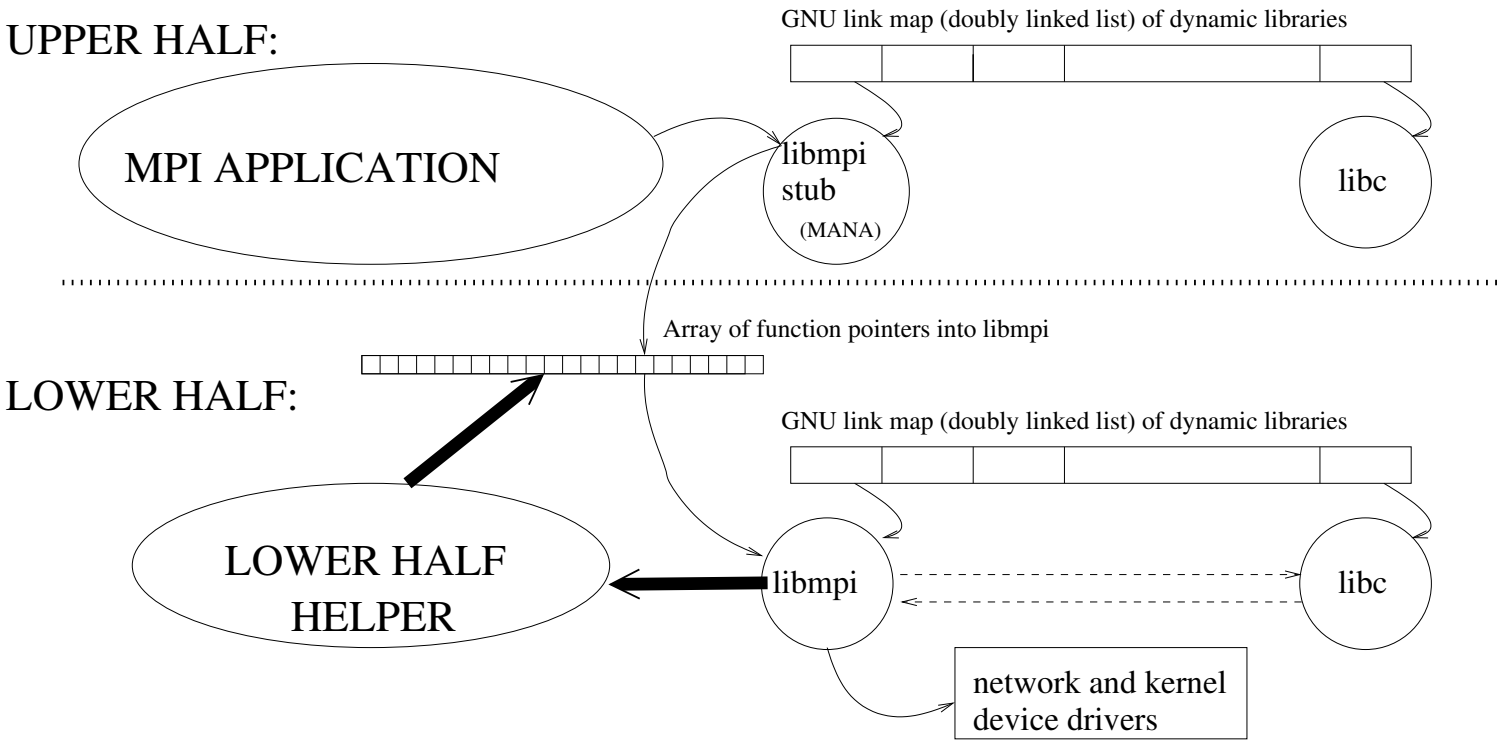
1. DMTCP (Distributed MultiThreaded Transparent CheckPointing) developed 15 years ago.
2. Transparent checkpointing of MPI over TCP/IP demonstrated (Jason Ansel et al., IPDPS'09).
3. Transparent checkpointing using an InfiniBand plugin (Jiajun Cao et al., HPDC'14)
4. Newer networks: InfiniBand extensions, Intel Omni-Path, Cray GNI/Aries, HPE Cray Slingshot, ...
5. MANA: DMTCP + plugin for split processes ("good enough" for Gromacs at small scale) (Rohan Garg et al., HPDC'19)
6. CRAC: DMTCP + plugin for CUDA with split processes. (Twinkle Jain et al., SC'20)

MANA (Past): Thesis of Rohan Garg: Split Process

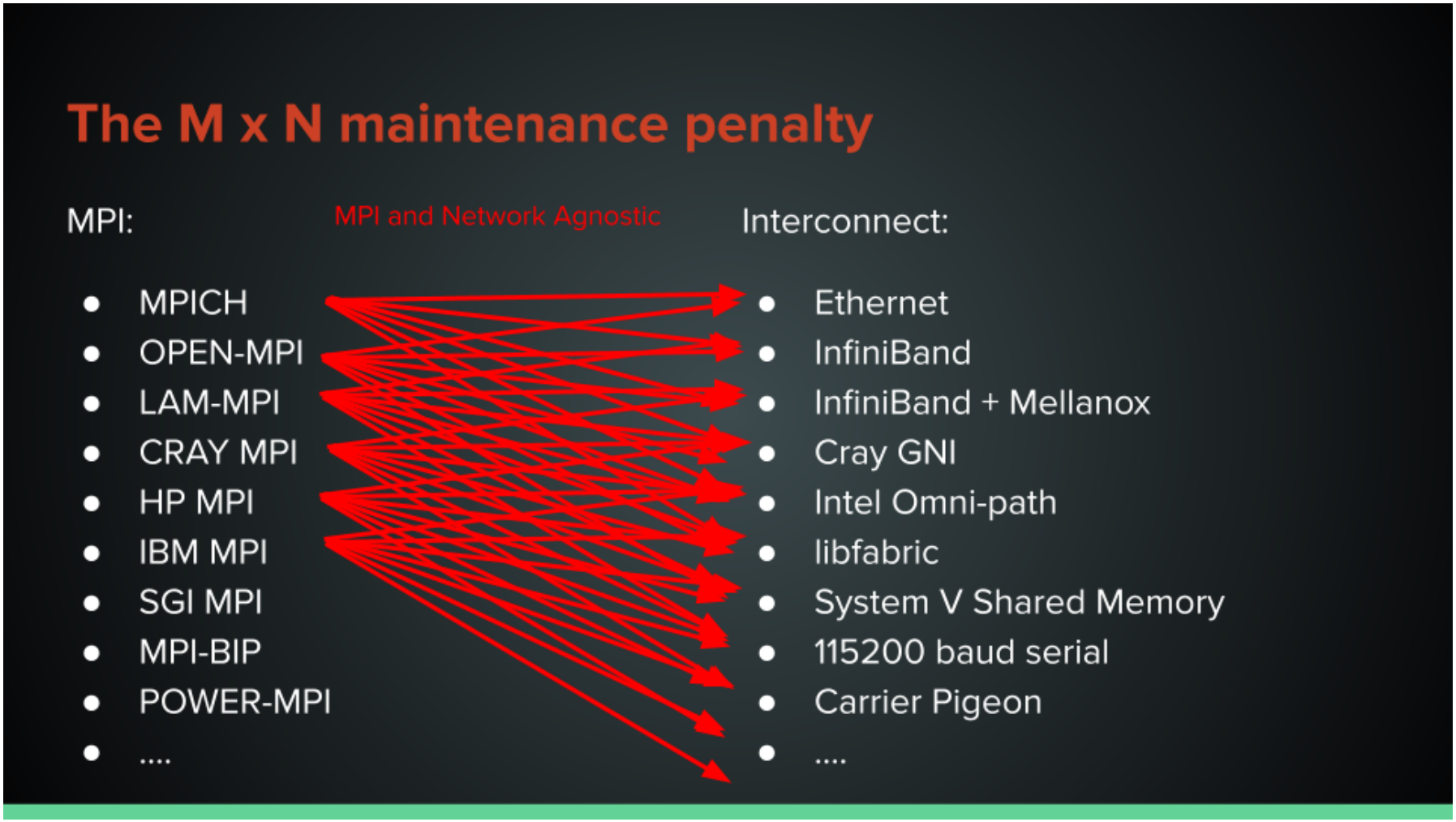


"MANA for MPI: MPI-Agnostic Network-Agnostic Transparent Checkpointing", Rohan Garg et al. (HPDC'19)

MANA (Past): Split Process: details



MANA (Past): Split Process: Solves the $m \times n$ problem



MANA/CRAC (Past): CUDA support by Twinkle Jain

As before, the key is Split processes

Integration of MANA with CRAC is planned for the near future.

Additional difficulties to overcome: MPI mostly assumes that the caller allocates new memory.

But in CUDA, the callee can allocate new memory from the "lower half", and then it's the responsibility of the "upper half" to save and restore that memory.

(See "CRAC: Checkpoint-Restart Architecture for CUDA with Streams and UVM", Twinkle Jain et al. (SC'20))

MANA (Present): Three-way collaboration

Present: Three-way collaboration: MemVerge, NERSC, & DMTCP

(MANA is fully functional free and open source on top of DMTCP, while **MemVerge, Inc.** can provide paid commercial support)

In progress: Validate commonly used applications at NERSC at scale: Currently validating VASP, CP2K and Nimrod

Many collaborators in current MANA development effort:

- Zhengji Zhao and Rebecca Hartman-Baker (NERSC/LBNL)
- Jun Gan, Illio Suardi, Dahong Li, Tom Wong, Yue Li (MemVerge)
- Kapil Arya (Microsoft Res. Lab.), Rohan Garg (Nutanix); former stud.
- Yao Xu and Twinkle Jain (current PhD students at Northeastern U.)

MANA (Future)

GOALS of MANA-2.0: to work well in production today; to be "future-proof" and ready for future architectures
(e.g., VeloC, MPI-4 Sessions, Kokkos, ULFM, Reinit)

Tools available with MANA/DMTCP:

split process: Add "glue" library to lower half that interacts with an external system (Recall: Lower half is not checkpointed.)

map topology to hardware on restart: topology saved prior to checkpoint; start lower half (map to hardware); then restore upper-half application: (Example: MANA handling of MPI_Cart_create())

process virtualization: virtualize any ids that persist between ckpt and restart

interposition: "spy" on application: detect and save dynamic information; use later at ckpt time (track application state; anticipate needs)

Moral: *MANA doesn't care what's in the lower half!* The lower half library might be code to talk to a different MPI Sessions. Or it might also include special subsystems (CUDA, OpenACC, FPGA, Apptainer/Singularity).

MANA: Past, Present and Future

MANA: *MPI-Agnostic Network-Agnostic* checkpointing.

Past: Showed feasibility in academic prototype (Garg et al., HPDC'19)

Present: Three-way collaboration: NERSC, MemVerge, and DMTCP
(fully functional free and open source version,
while **MemVerge, Inc.** provides paid commercial support)

Near Future (medium-scale checkpointing):

1. **Robust, efficient transparent checkpointing at medium scale HPC**
2. **Killer App: Chaining of allocations for long-running computations (similar to spot instances in the Cloud)**
3. **Validate a subset of applications for MANA-compatible *system-level* checkpointing — support on-demand, real-time computing**

Extreme scale computing:

- **MANA-style split processes as one component in a larger ecosphere**

QUESTIONS?

The following organizations are acknowledged for past and present support: