A Personal Cloud Controller

Yuan Luo

School of Informatics and Computing, Indiana University Bloomington, USA

PRAGMA 26 Workshop

DATA TO INSIGHT CENTER
INDIANA UNIVERSITY
Pervasive Technology Institute

SDSC
SAN DIEGO SUPERCOMPUTER CENTER

Problems

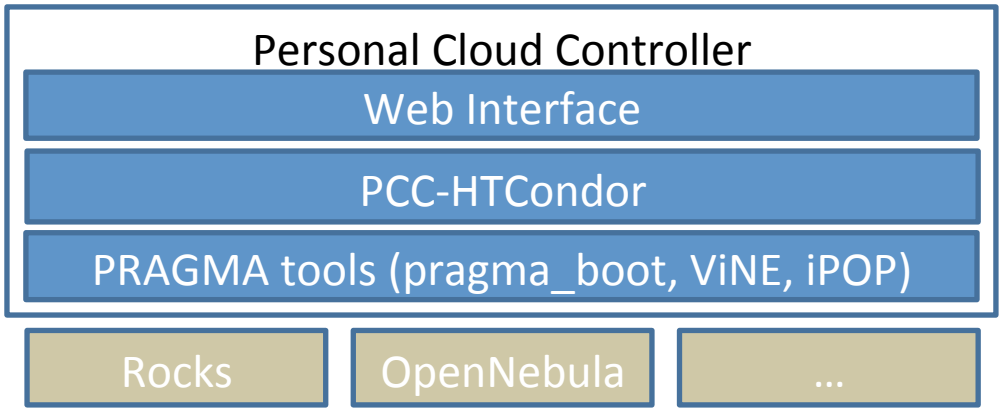
- To fully leverage existing resources for their applications, users often need to manually configure available resources to a custom configuration, a time consuming task.
- Labs may have a suite of applications with different requirements and runtime profiles (e.g., compute intensive, data intensive, data sensitive) necessitating different scheduling approaches.
- Collaborative e-Science projects typically require data processing to be performed on distributed data sets. Some data are so sensitive that they cannot be processed out of certain administrative boundaries.
- Checkpointing is a direction to achieve fault tolerance. But in some cases, checkpointing is expensive as it consumes large amount of time and resources.

Solution Space

- User Controllability
- Resource Allocation
- Sensitive Data Processing
- Fault Tolerance for Long-Run Applications

Personal Cloud Controller
(Solution to user controllability)

- Personal Cloud Controller (PCC) integrates and leverages PRAGMA tools to easily deploy and manage virtual clusters.
- PCC is lightweight and extends HTCondor for scheduling and fault tolerance capabilities. PCC
  - Provides flexible interface to enable a high level of user controllability,
  - Automates configuration, deployment, and fault recovery on behalf of user.



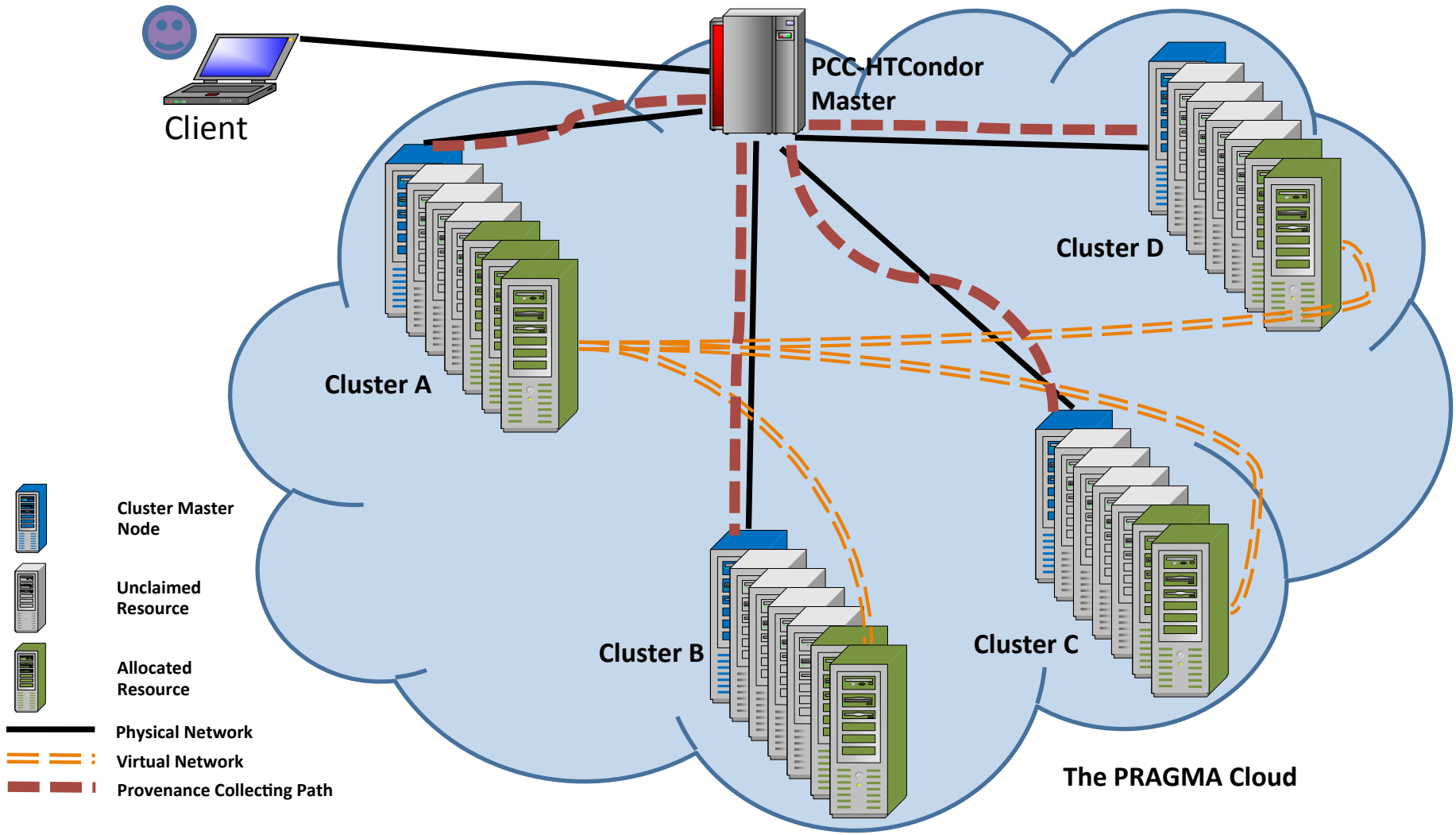
Personal Cloud Controller – *cont'd*

Fundamentals of the following…

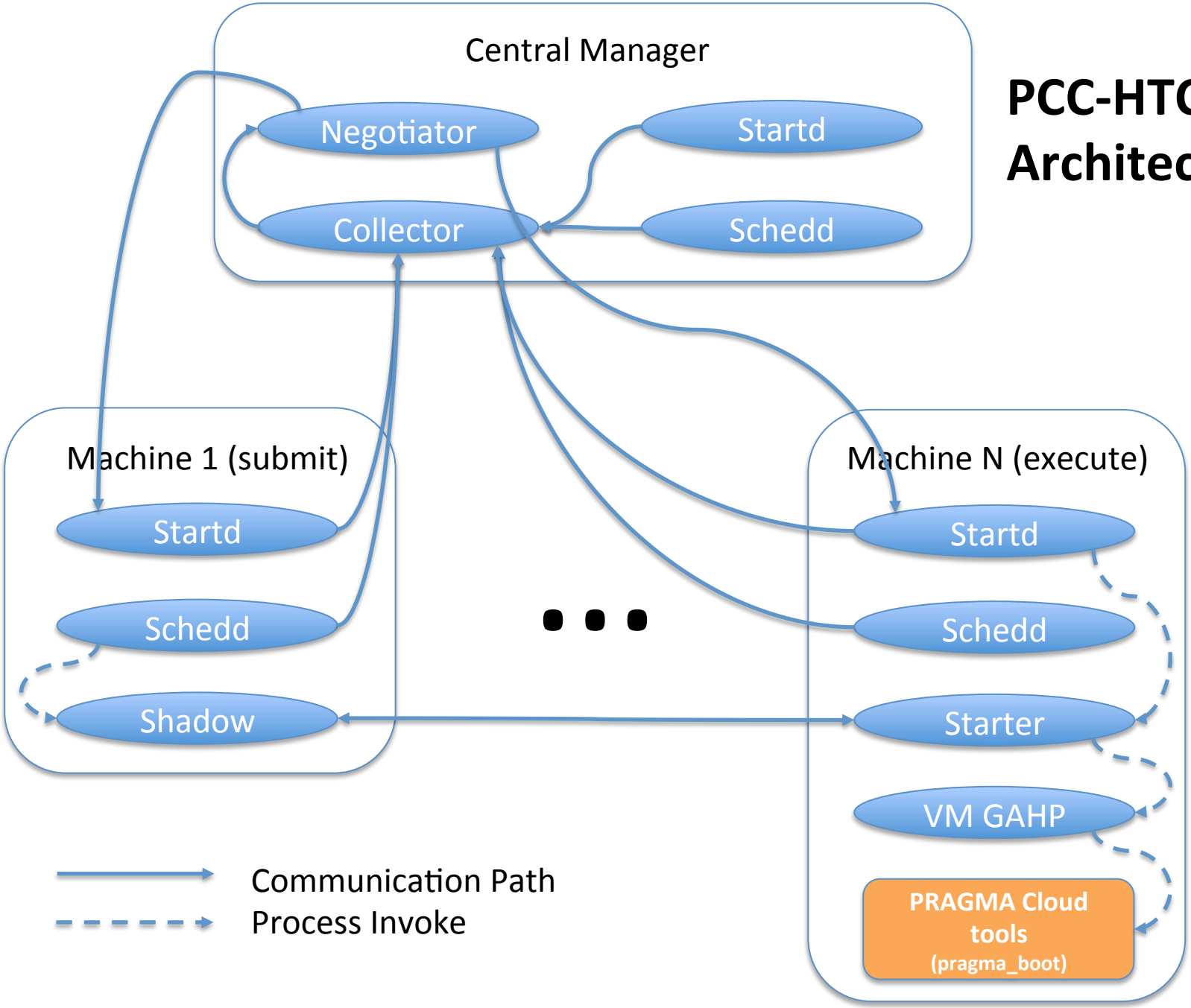
- Resource Allocation
  - Resources and applications advertising
  - Match resources and applications
- Sensitive Data Processing
  - PCC enabled PRAGMA Cloud
- Long-Run Applications Fault Tolerance
  - HTCondor's checkpointing mechanism

    (complementary to provenance-based fault tolerance)

PCC Enabled PRAGMA Cloud



PCC-HTCondor Architecture



Ongoing and Future Work

- Develop a set of application-driven resource allocation strategies and scheduling algorithm based on data locality, sensitivity, etc.
- Extend the Hierarchical MapReduce model to support sensitive data processing.
- Develop a provenance-based fault-tolerance model to handle long-run applications.

Acknowledgement

- The author would like to thank
  - Dr. Beth Plale (Indiana University)
  - Dr. Philip Papadopoulos, and Shava Smallen (UCSD)
- This work funded in part by NSF Award OCI 1234983

Thank you!