

# TAZeR: Accelerating Remote Data for Distributed Scientific Workflows

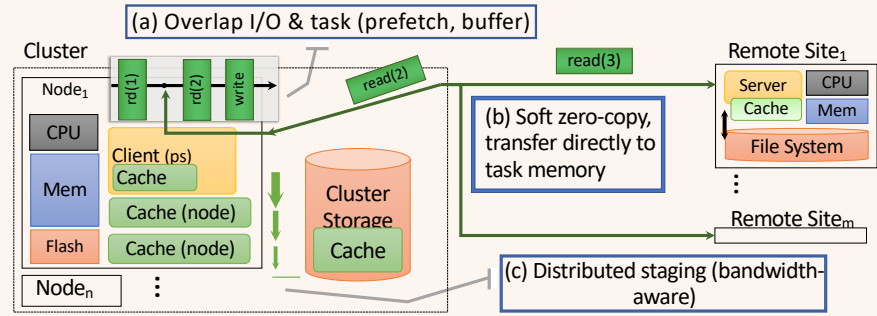
Collaboration Members: US (PNNL) : Ryan Friese(PI), **Burcu O. Mutlu**, Nathan Tallent (*firstname.lastname@pnnl.gov*)  
 Japan (KEK) : Takanori Hara(PI), Ikuo Ueda, Kideki Miyake

Challenge: Remote I/O blocks task, consumes space, bottlenecks data movement

## TAZeR: Transparent Asynchronous Zero-copy Remote I/O

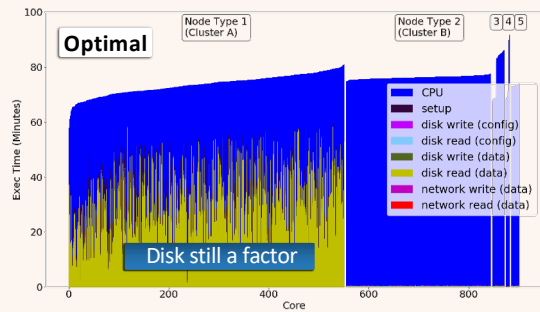
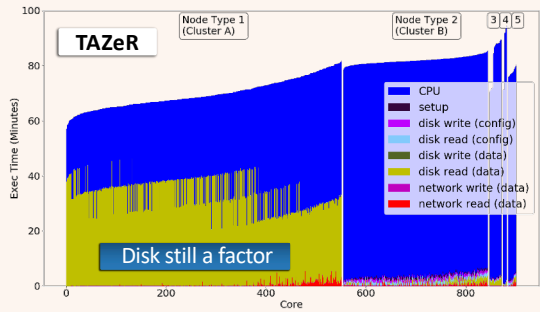
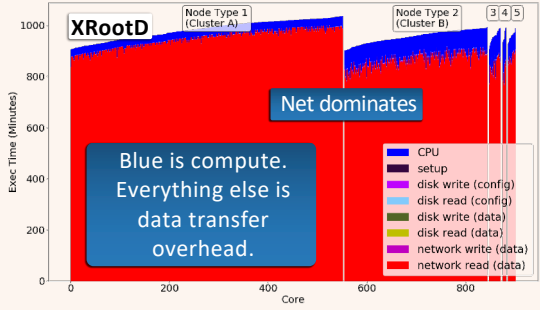
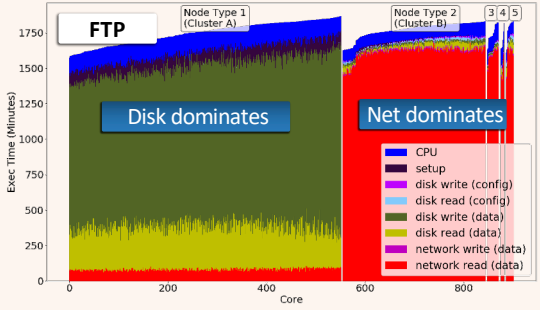
- Reduce data moves
- Reduce blocking
- Increase bandwidth
- Lightweight
- Transparent
- Increase reuse

TAZeR: Hiding the Cost of Remote I/O in Distributed Scientific Workflows  
<http://doi.org/10.1109/BigData47090.2019.9006418>



### Feedback to Belle II/KEK

- With TAZeR:
- ↓ moves
  - ↓ NW
  - ↓ disk
- TAZeR is
- 22x < FTP
  - 12x < XRootD
  - 7% > Optimal



### Effectively Using Remote I/O For Work Composition in Distributed Workflows – BigFlowSim

- *Motivation*: exploration of the workflow parameter space to effectively use remote I/O for composing work in distributed workflows
- Five key remote I/O parameters
  - I/O intensity
  - Access pattern
  - Data (re)use
  - I/O aware locality
  - Workflow I/O Runtime
- BigFlowSim allows developers to quickly model potential workflow I/O decisions and perform reasoned analysis

Effectively Using Remote I/O For Work Composition in Distributed Workflows  
<http://doi.org/10.1109/BigData50022.2020.9378352>

### Data Informed Scaling of I/O tagging and Bandwidth

- *Motivation*: Enabling the execution of data intensive distributed workflows in resource constrained environments
- Assigning staging resources to objects to have a dynamic staging policy
- The way we stage objects changes with access patterns
- Dynamically adjusting staging policies to improve staging performance