Condor
High Throughput Computing

Managing Network Resources in Condor

Jim Basney
Computer Sciences Department
University of Wisconsin-Madison
jbasney@cs.wisc.edu

Why is the Network Important?

- **Increase in physical memory per CPU**
  - Larger checkpoints
- **Increase in size of Condor pools**
  - 700 CPUs in our local pool
- **Increase in remote execution across WAN**
  - WAN pools (INFN)
  - Flocking: UW, NCSA, UNM, INFN
  - Remote Submitters: Personal Condor

Types of Network Usage

- Placement
- Periodic Checkpoints
- Preemption
- Remote I/O

Network Management Goals

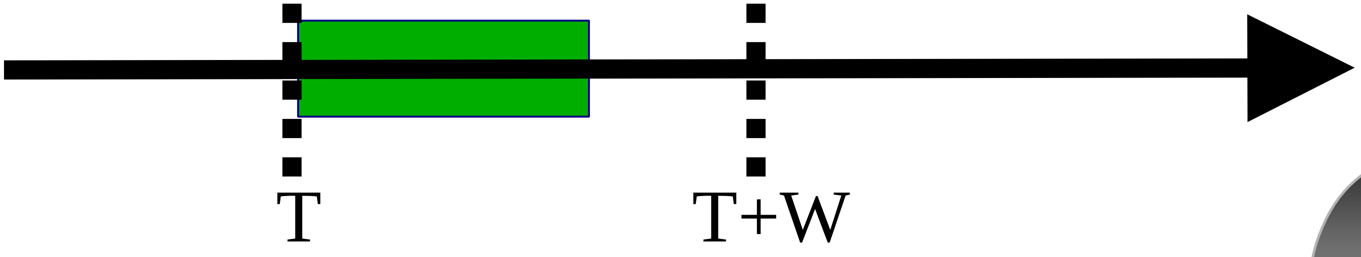
- **Provide Administrative Control**
  - HTC applications must co-exist with other network users
- **Improve Application Efficiency: Goodput**

Monitoring Network Usage

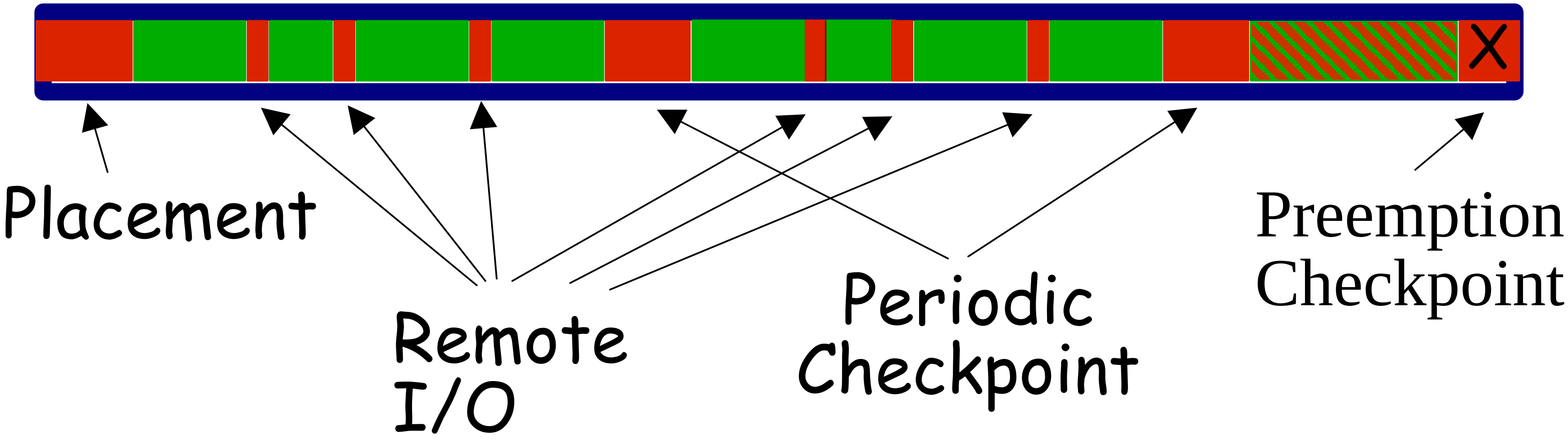
- **Configure Network Routing Info**
- **Monitor Network Usage Per User & Subnet**
  - Checkpoint & Executable Transfers
  - Remote System Calls
- **CondorView Visualization**

Network & CPU Co-Allocation

- **For each Subnet, configure:**
  - Available capacity
  - Allocation window
- **Job Placement requires capacity for**
  - Checkpoint & Executable Transfer
  - Remote I/O (estimated)



- Goodput = Allocation - Network Overhead



Placement

Remote I/O

Periodic Checkpoint

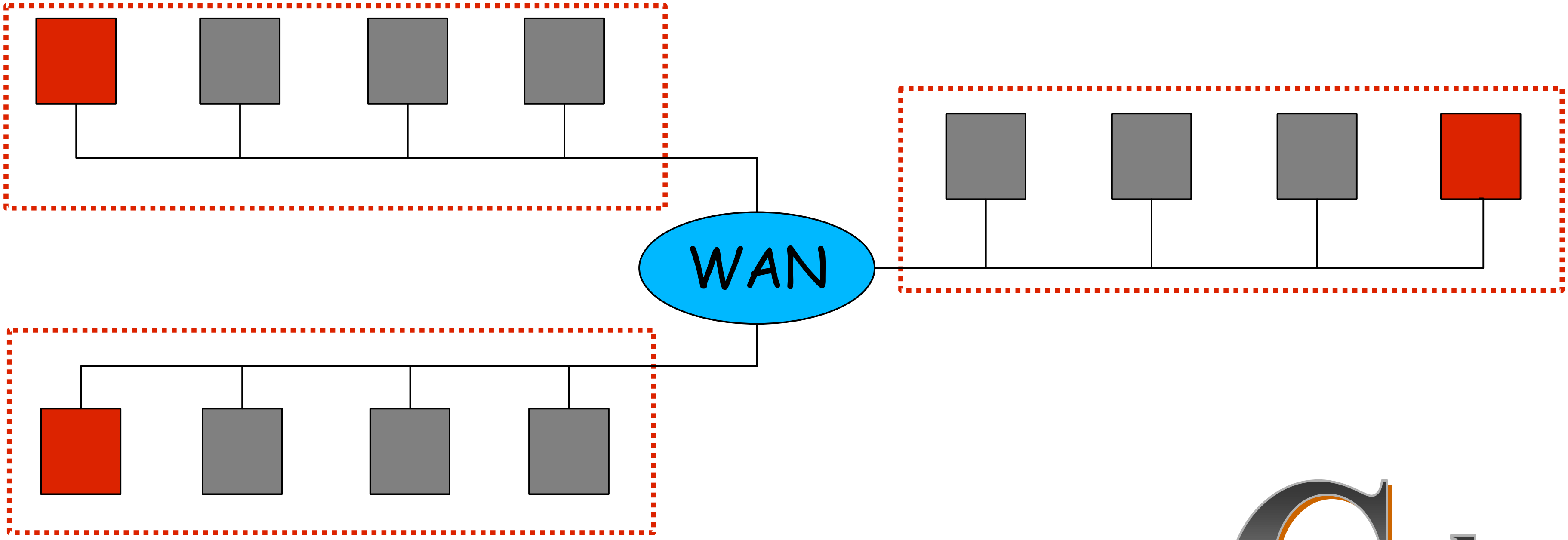
Preemption Checkpoint

Use Network Efficiently

- **Compressed Checkpoints**
  - CPU vs. network resources
- **Incremental Checkpointing**
  - Record changes since last checkpoint
- **Buffered Remote I/O (Doug Thain)**
  - Latency Hiding
  - Avoid multiple reads/writes of same file data

Ckpt and Filesystem Domains

- Provide local access to checkpoint and file storage



Checkpoint Domains

- **Resource offer includes nearest server**
  - CkptServer = “ckpt.cs.wisc.edu”
- **Job must remain in checkpoint domain**
  - LastCkptServer = “ckpt.cs.wisc.edu”
  - Requirements = My.LastCkptServer ==
    Target.CkptServer

Checkpoint Domains (cont.)

- **Job may migrate if no CPUs available in domain**
  - Requirements =
    ```
    (My.LastCkptServer ==
     Target.CkptServer) ||
    (CurrentTime - My.LastPreemptTime >
     86400)
    ```
  - Rank = My.LastCkptServer ==
    Target.CkptServer

Filesystem Domains

- **Resource offer includes filesystem domain**
  - FileSystemDomain = “cs.wisc.edu”
- **Job runs where input data is staged**
  - Requirements =
    Target.FilesystemDomain == “cs.wisc.edu”

Filesystem Domains (cont.)

- **Resource offer may include staged datasets**
  - HasDataSet174 = True
- **Job runs where dataset is staged**
  - Requirements = Target.HasDataSet174;

Co-Allocation Revisited

- **Network-Aware CPU Requests**
  - Requirements =
    CPUBW > 8.0 && RSCBW > 4.0;
  - Rank = RestartBW;
  - Rank = 0 - RSCHops;
- **Time-based capacity specification**
  - Limit WAN bandwidth used during work hours

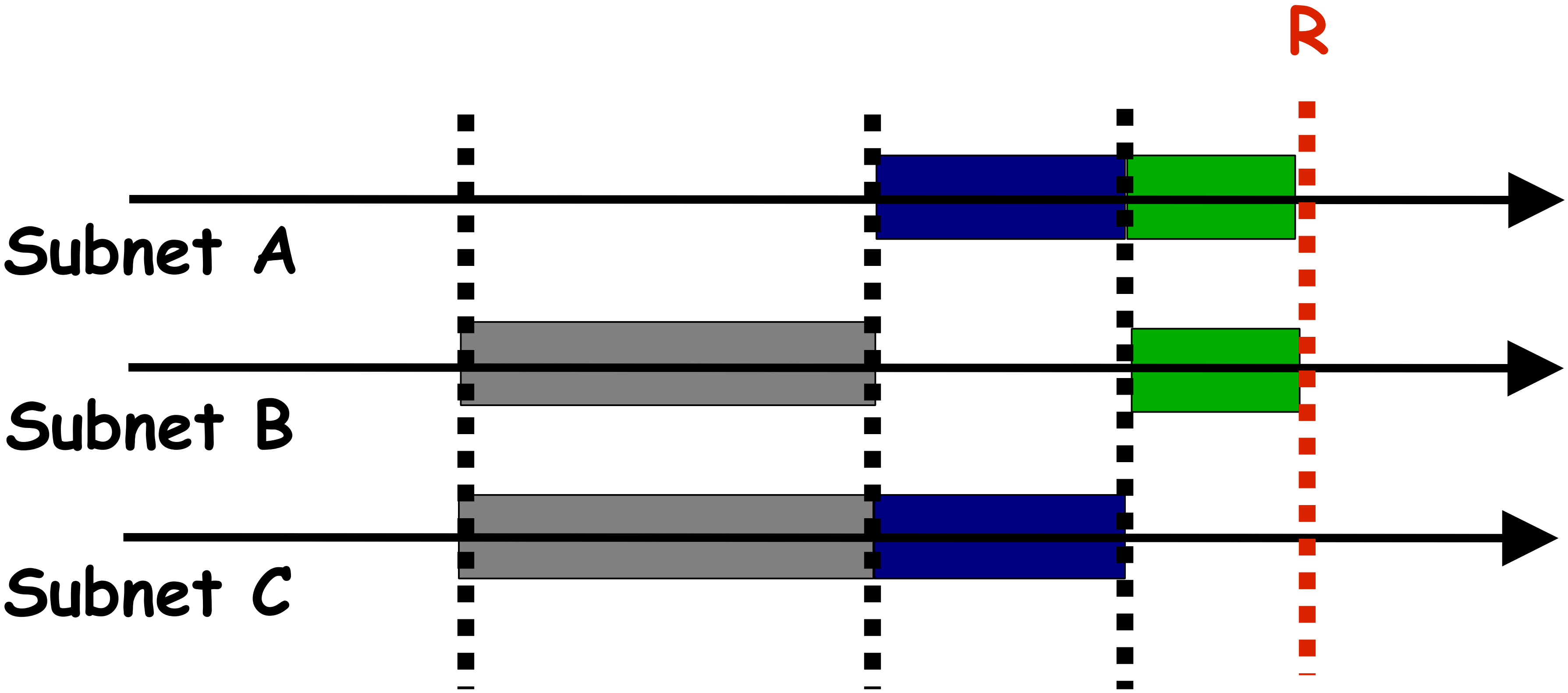
Scheduling Preemption Ckpts

- **Time to checkpoint is limited when preempted**
  - Preempting user doesn't want to wait
- **Simultaneous preemptions**
  - Heavy network demand
  - Slow checkpointing
  - Missed deadlines / Failed checkpoints

Scheduling Preemption Ckpts

- **Many preemption events may be anticipated**
  - Start of class for lab workstation
  - Start of work hours for office workstation
  - System maintenance
- **Schedule preemption checkpoints in advance of reservations**

Scheduling Preemption Ckpts



Scheduling Periodic Ckpts

- **Goals:**
  - Complete checkpoint quickly
  - Don't interfere with more important transfers
- **Perform when network is otherwise idle**
  - Avoid synchronized periodic checkpoints

Network Scheduling

- **Fit jobs to network topology**
  - Place network-intensive jobs on fast networks
  - Place jobs near their data
- **Locate best checkpoint and file servers at run-time**
- **Pre-fetch and store-behind application data when network capacity is available**

Network Scheduling (cont.)

- Balance checkpoint costs with expected allocation time
- Preempt or migrate heavy network users
- Backfill pool with light network users to fully utilize CPUs

Summary

- Making the network a Condor-managed resource
- Provide administrative control over HTC network usage
- Improve execution efficiency by co-scheduling network and CPU resources