



Computer Science Department  
University of Wisconsin-Madison



Computer Science Department  
Technion, Haifa



# Gozal

## Technion's Condor enhancements

High availability, Usability and  
Manageability

[Mark Silberstein](#), Assaf Schuster  
Technion – Israel Institute of Technology

# Introduction

- Gozal (in Hebrew) - nestling
- Started in 2002 as Condor deployment support project in the Distributed Systems Lab , headed by Prof Assaf Schuster
- Today: 15 undergraduates, 4 graduate students, and 2 staff members

# Our Goals

- Adding functionality to Condor
- Using Condor for teaching real distributed systems
- Using Condor for research

# 1. Highly Available Matchmaker

- The problem we want to solve
  - Matchmaker is a **single-point-of-failure**
- Our solution - **Highly Available Matchmaker**
  - **Automatic** failure detection
  - **Transparent** failover (no global configuration change) for the pool entities
  - **Negligible throughput degradation** in case of failure

## 2. Efficient invocation of short data-intensive jobs

- The problem we want to solve
  - **Optimize Condor performance** for large batches of **short** jobs
- Our solution
  - Allows **significant execution time speedup compared to regular Condor**
  - Designed as **add-on module**
    - Condor native executables/configuration file are **unmodified**
  - **Completely transparent**
    - Only change – use `our_condor_submit`

# Performance evaluation

- BLAST invocation with short searches (1-3 sec)
  - Input data=20MB
  - 14 CPUs
  - 100Mb LAN
  - 10000 Jobs
- ~**12 times faster** than regular Condor
  - Even better results expected over WAN

# 3. Resource protection - “sandbox”

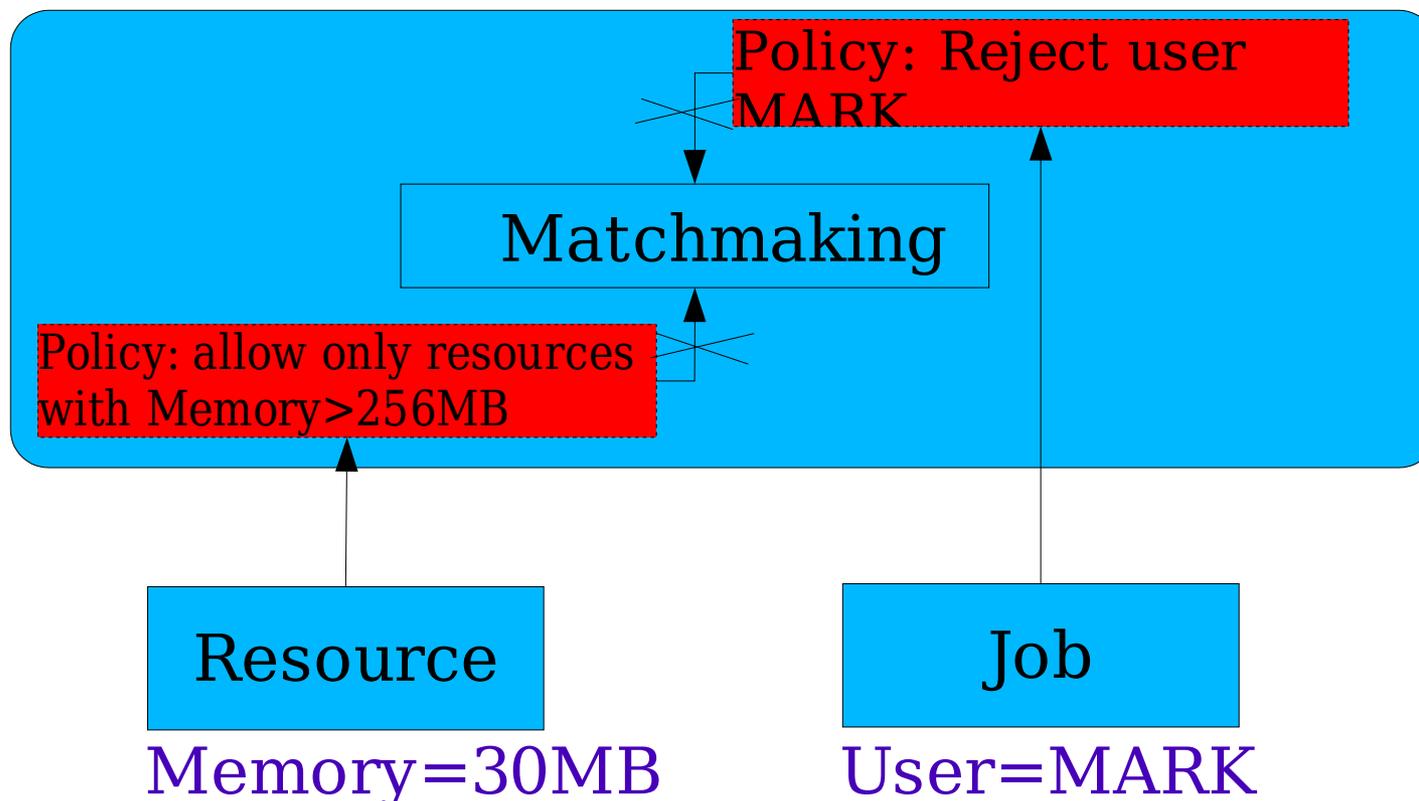
- The problem we want to solve
  - **Malicious job** can render a resource unusable or obtain private information
  - Condor can be exploited for distributed attacks on other systems
- Solution – resource protection sandbox
  - **Auditing of file system** and network access, OS resources
  - Integration with Condor

# 4. Centralized policy enforcement

- Problem we want to solve
  - Changing pool-wide policy requires **global change** of configuration of all resources
  - How to change resource or queue policy when **there is no access** to its configuration file
  - Pool-wide policy is difficult to enforce
    - How to force a user or an owner to specify new requirements or attributes?

# Centralized policy enforcement – our solution

- Negotiator is able to filter “wrong classads”
  - We added “wrong classads” criteria



# Centralized policy enforcement – our solution

- Negotiator can “AND” our expression with the Requirements expressions
  - Example: All computers from my lab must require jobs to **explicitly specify** memory use
- Negotiator can alter specified classads
  - Example: If a (Job Owner is X or Y or Z), add GroupID=1

# 5. Centralized configuration framework

- “The down side of Condor flexibility is its complexity”
- Lack of **centralized** pool configuration
  - Editing multiple copies of configuration files is error-prone
- System administrator **should learn Condor internals** to configure Condor pool

# Centralized configuration framework (cont)

- Centralized DB for pool configurations
  - Resources pull the updates
- Unified GUI for pool management
- Configuration templates abstraction (define once, use many)
  - Functional change to Condor configuration, i.e. `AllowUserOnResource(Username)`,
- Resource configuration groups
  - Resource can be a member of one or more groups
  - Resource inherits configuration from the groups of which it is a member

# Projects

1. Highly available Condor Matchmaker
2. Efficient invocation of short data intensive jobs
3. Resource protection - “sandbox”
4. Centralized policy enforcement
5. Centralized configuration and management framework

# Summary

- We are developing addons for improved usability and manageability of Condor
- We work in tight collaboration with the Condor team
- Some of the projects already have the deliverables available in beta
- The work is in progress by staff members to make releases stable

# Do you want to try it?

- More information at <http://dsl.cs.technion.ac.il/projects/gozal>
- This work would not be possible without
  - Prof Schuster, Eran Issler, Noam Palatin, Gabi Kliot and all the undergraduate students who worked on these projects

and of course **the Condor team and Prof Miron Livni**

- Contact: [marks@tx.technion.ac.il](mailto:marks@tx.technion.ac.il)

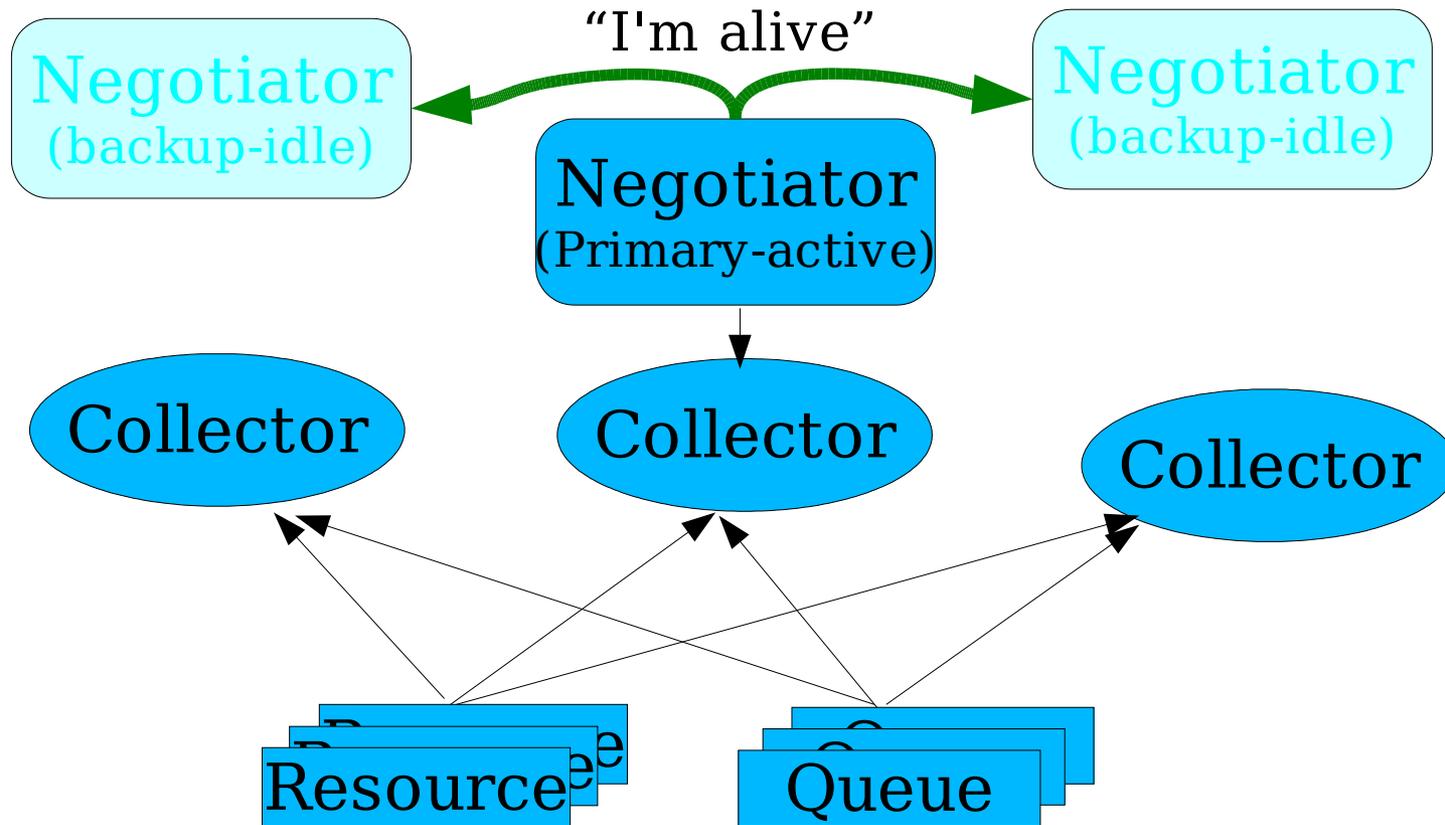
**THANK YOU**

# In case I have time

- Technical zoom on the projects

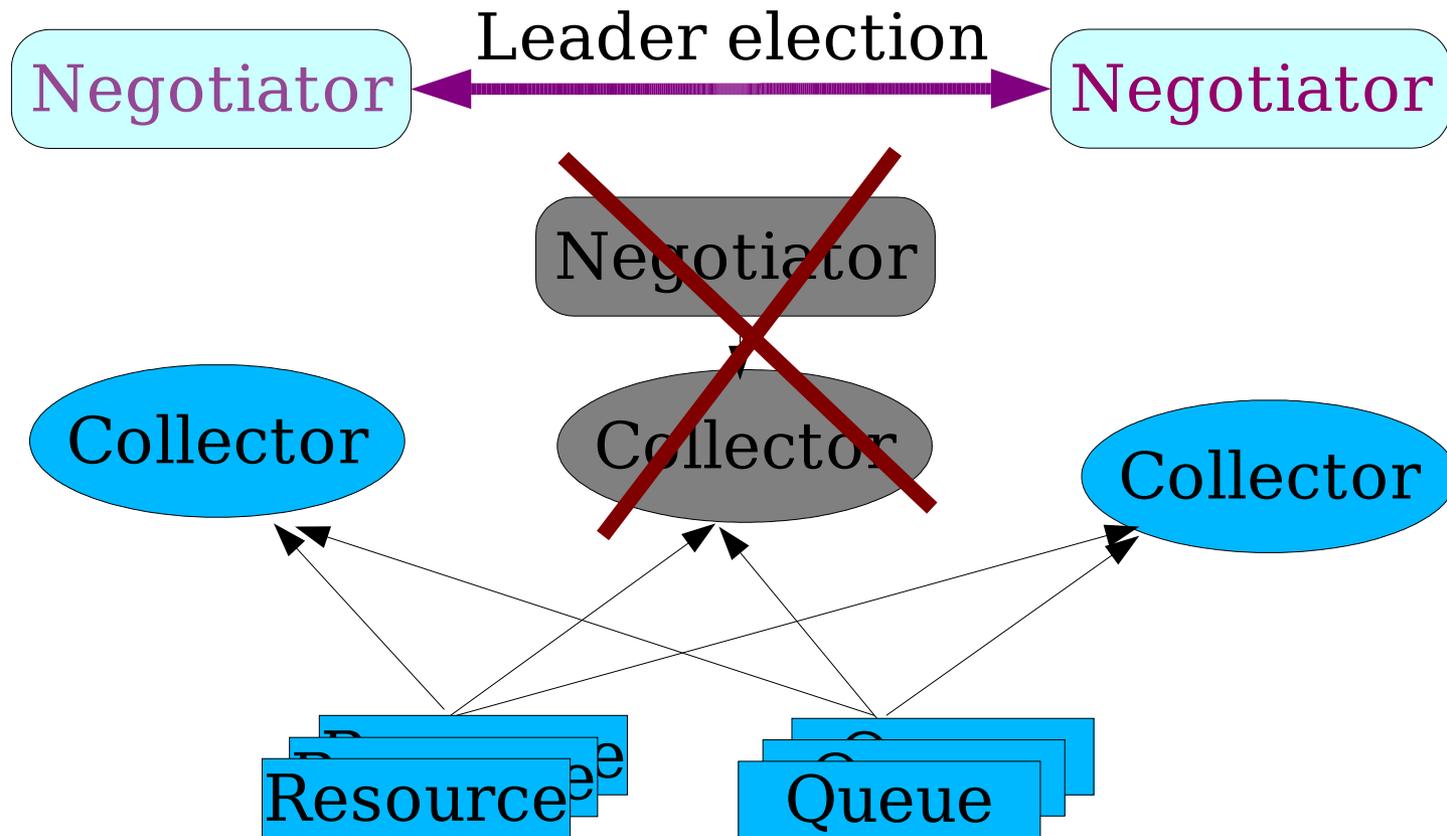
# How it works

## HA matchmaker 1



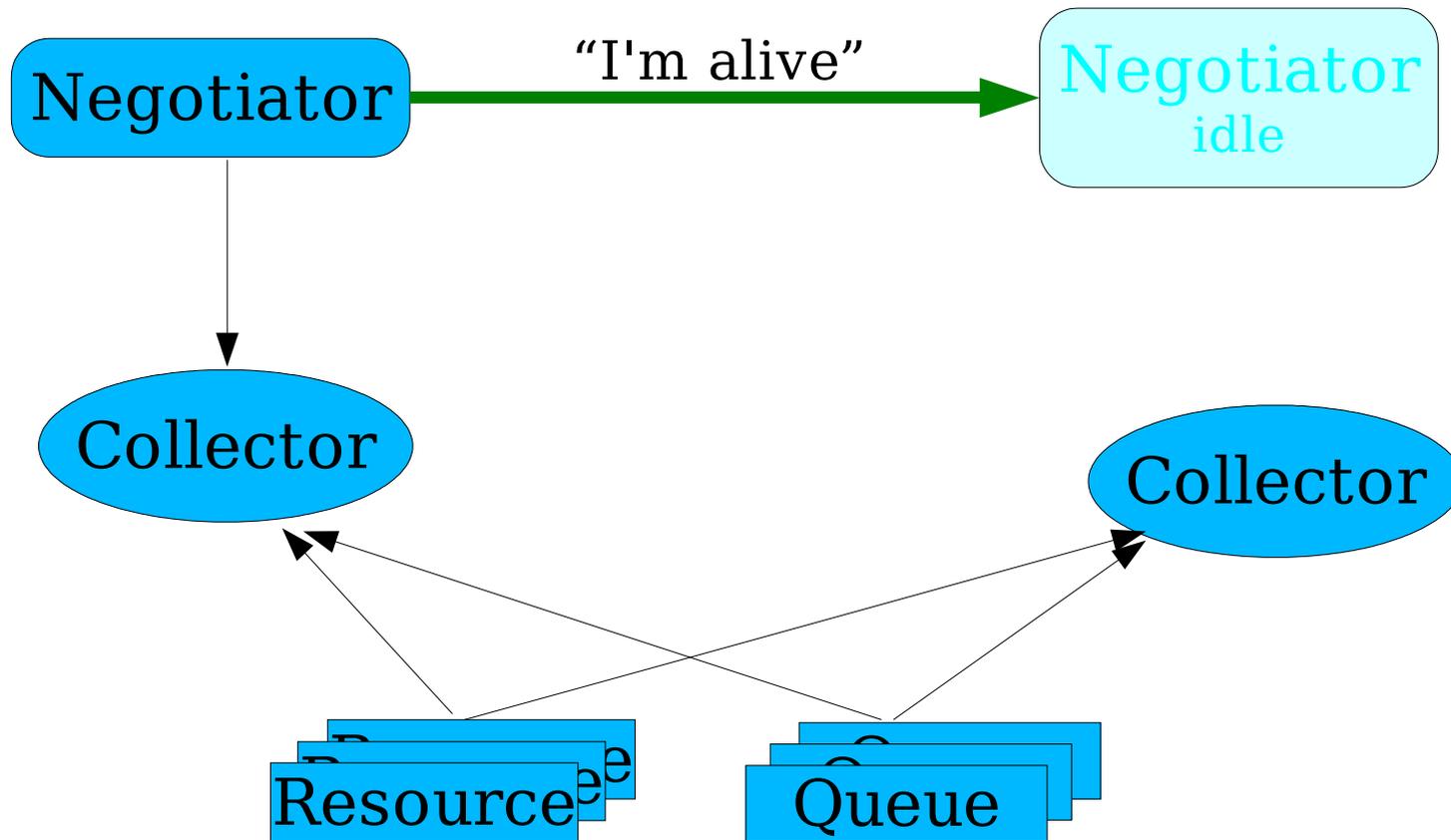
# How it works

## HA matchmaker 2



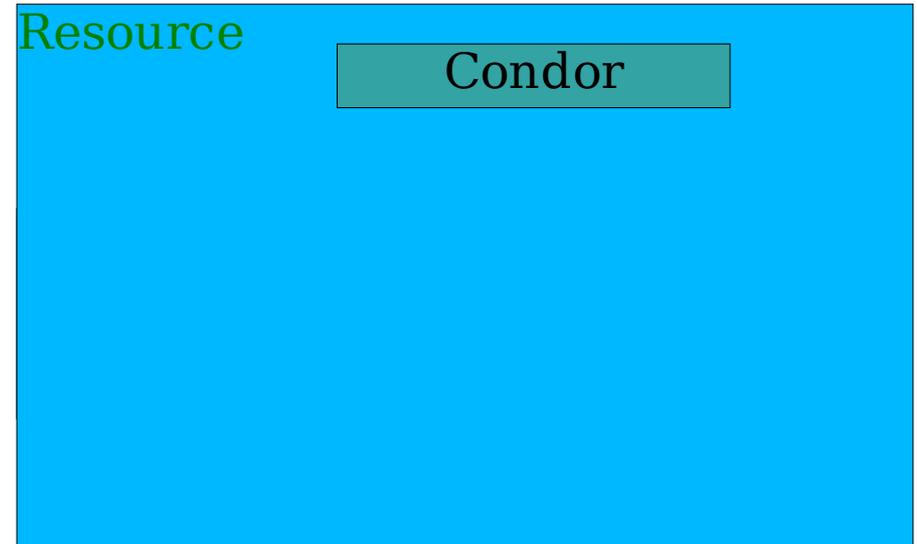
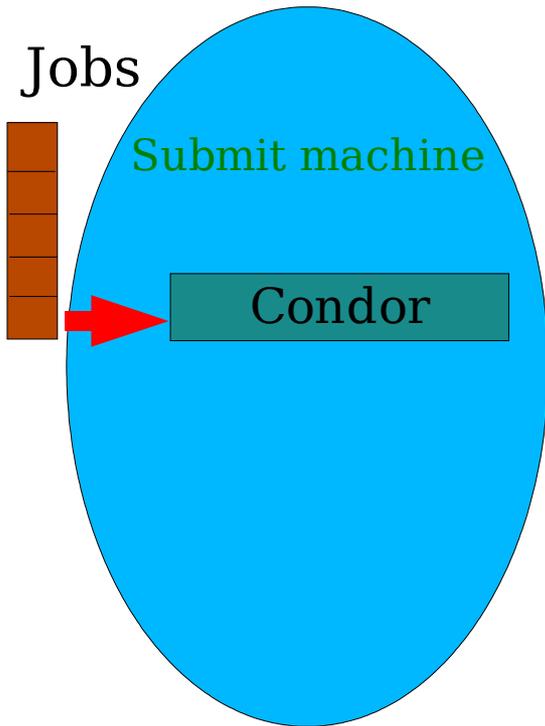
# How it works

## HA matchmaker 3



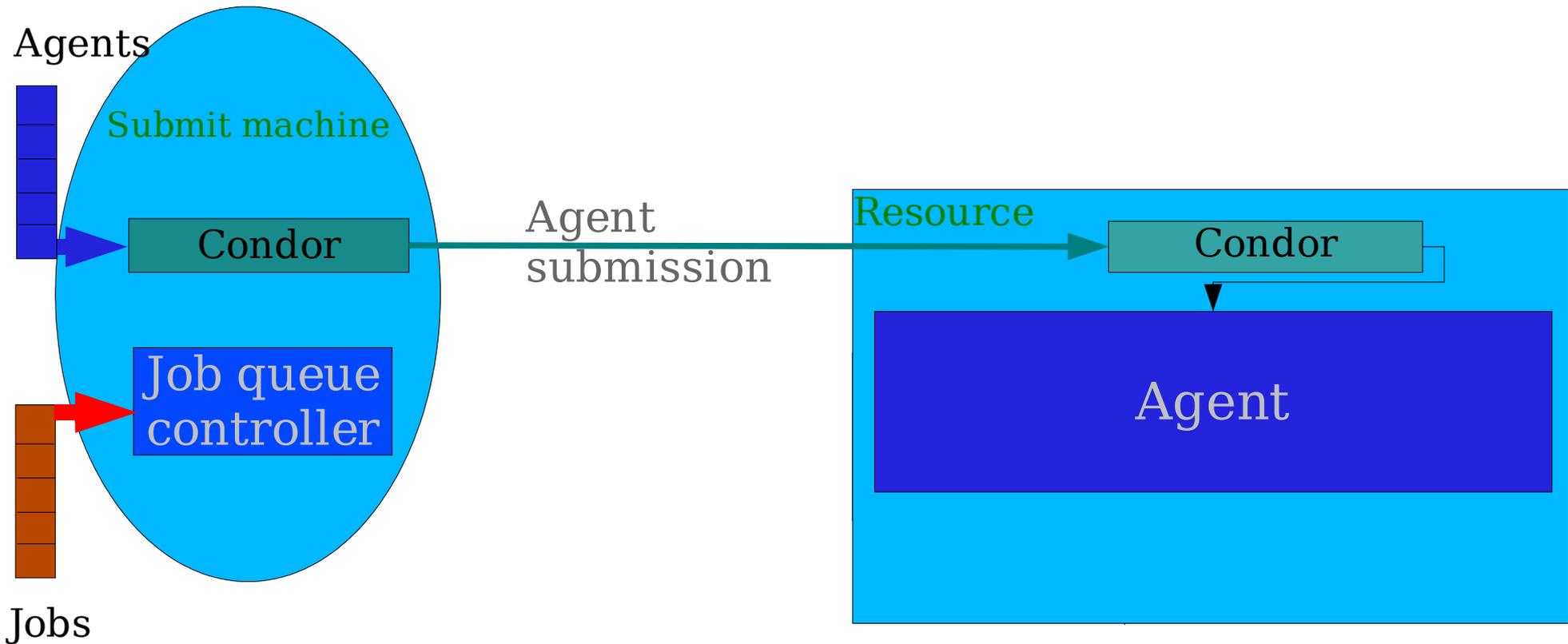
# How it works

## Efficient data-intensive short jobs



# How it works

## Efficient data-intensive short jobs



# How it works

## Efficient data-intensive short jobs

