Branch-and-Bound with Peer-to-Peer for Large-Scale Grids

Alexandre di Costanzo

INRIA - CNRS - I3S - Université de Sophia Antipolis

Ph.D. defense
Friday 12th October 2007

Advisor: Prof. Denis Caromel

The Big Picture

The Big Picture

- Objective

The Big Picture

- Objective

Solving combinatorial optimization problems with Grids

The Big Picture

- Objective

Solving combinatorial optimization problems with Grids

- Approach

The Big Picture

- Objective

Solving combinatorial optimization problems with Grids

- Approach

Parallel Branch-and-Bound and Peer-to-Peer

The Big Picture

- Objective

Solving combinatorial optimization problems with Grids

- Approach

Parallel Branch-and-Bound and Peer-to-Peer

- Contributions

The Big Picture

- Objective

Solving combinatorial optimization problems with Grids

- Approach

Parallel Branch-and-Bound and Peer-to-Peer

- Contributions

1. Branch-and-Bound framework for Grids

The Big Picture

- Objective

  Solving combinatorial optimization problems with Grids

- Approach

  Parallel Branch-and-Bound and Peer-to-Peer

- Contributions

  1. Branch-and-Bound framework for Grids

  2. Peer-to-Peer Infrastructure for Grids

The Big Picture

- Objective

Solving combinatorial optimization problems with Grids

- Approach

Parallel Branch-and-Bound and Peer-to-Peer

- Contributions

1. Branch-and-Bound framework for Grids

2. Peer-to-Peer Infrastructure for Grids

3. Large-scale experiments

Agenda

- Context, Problem, and Related Work
- Contributions
  - Branch-and-Bound Framework for Grids
  - Desktop Grid with Peer-to-Peer
  - Mixing Desktops & Clusters
- Perspectives & Conclusion

Context

Context

- Combinatorial Optimization Problems (COPs)
  - costly to solve (finding the best solution)

Context

- Combinatorial Optimization Problems (COPs)
  - costly to solve (finding the best solution)
- Branch-and-Bound (BnB)
  - well adapted for solving COPs
  - relatively easy to provide parallel version

Context

- Combinatorial Optimization Problems (COPs)
  - costly to solve (finding the best solution)
- Branch-and-Bound (BnB)
  - well adapted for solving COPs
  - relatively easy to provide parallel version
- Grid Computing
  - large pool of resources
  - large-scale environment

Branch-and-Bound

Branch-and-Bound

Consists of a partial enumeration of all feasible solutions and returns the guaranteed optimal solution

Branch-and-Bound

Consists of a partial enumeration of all feasible solutions and returns the guaranteed optimal solution

- Feasible solutions are organized as a tree: **search-tree**

Branch-and-Bound

Consists of a partial enumeration of all feasible solutions and returns the guaranteed optimal solution

- Feasible solutions are organized as a tree: **search-tree**
- 3 operations:

Branch-and-Bound

Consists of a partial enumeration of all feasible solutions and returns the guaranteed optimal solution

- Feasible solutions are organized as a tree: **search-tree**
- 3 operations:
  - **Branching**: split in sub-problems

Branch-and-Bound

Consists of a partial enumeration of all feasible solutions and returns the guaranteed optimal solution

- Feasible solutions are organized as a tree: **search-tree**
- 3 operations:
  - **Branching**: split in sub-problems
  - **Bounding**: compute lower/upper bounds (*objective function*)

Branch-and-Bound

Consists of a partial enumeration of all feasible solutions and returns the guaranteed optimal solution

- Feasible solutions are organized as a tree: **search-tree**
- 3 operations:
  - **Branching**: split in sub-problems
  - **Bounding**: compute lower/upper bounds (*objective function*)
  - **Pruning**: eliminate bad branches

Branch-and-Bound

Consists of a partial enumeration of all feasible solutions and returns the guaranteed optimal solution

- Feasible solutions are organized as a tree: **search-tree**
- 3 operations:
  - **Branching**: split in sub-problems
  - **Bounding**: compute lower/upper bounds (*objective function*)
  - **Pruning**: eliminate bad branches
- Well adapted for solving COPs [Papadimitriou 98]

Branch-and-Bound

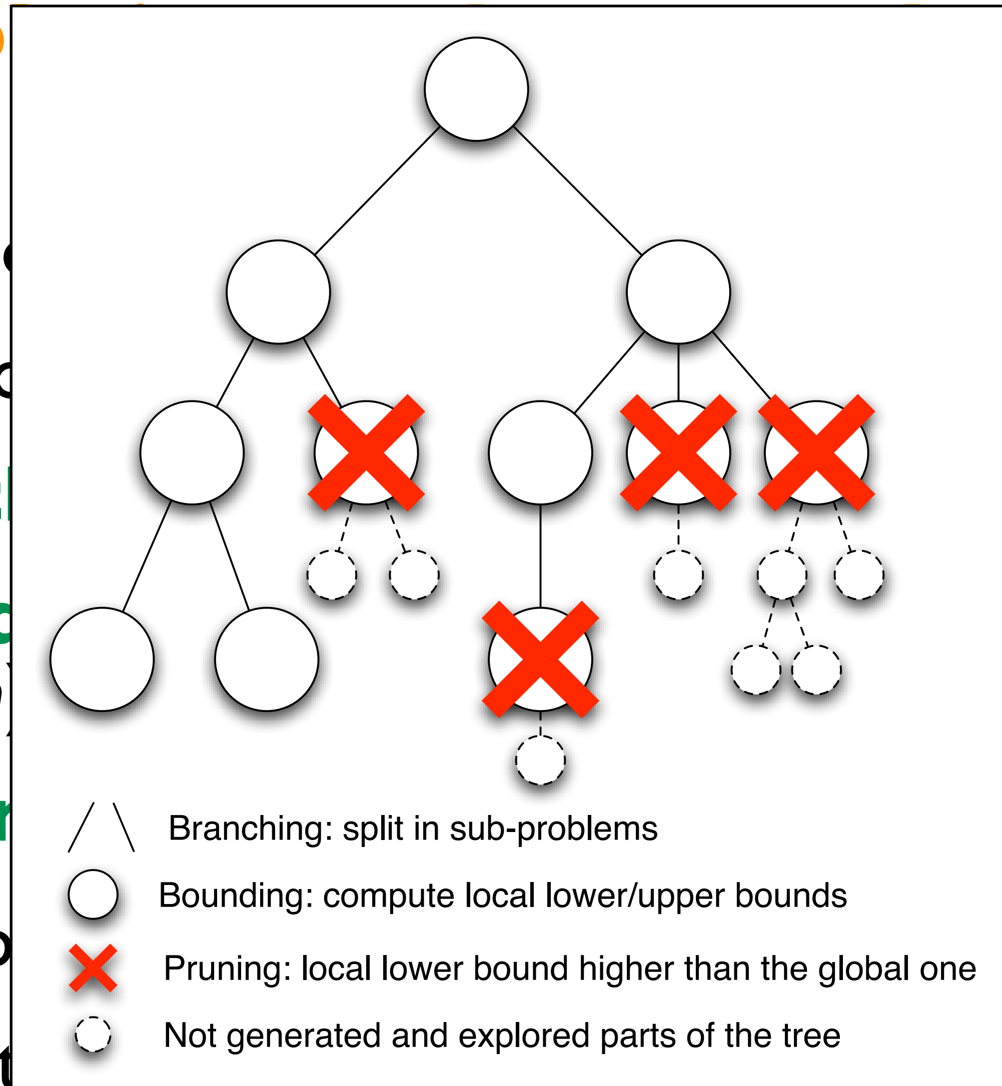
Consists of a partial enumeration of all feasible solutions and returns the guaranteed optimal solution

- Feasible solutions are organized as a tree: **search-tree**
- 3 operations:
  - **Branching**: split in sub-problems
  - **Bounding**: compute lower/upper bounds (*objective function*)
  - **Pruning**: eliminate bad branches
- Well adapted for solving COPs [Papadimitriou 98]
  - return the best combinaison out of all

Branch-and-Bound

Consists of a partial enumeration of all feasible so… guaranteed

- Feasible s… : **search-tree**
- 3 operati…
  - **Branc…**
  - **Bound…** …unds (*objective function*)…
  - **Pruni…**
- Well adap…
  - return t…



Parallel Branch-and-Bound

Parallel Branch-and-Bound

- COPs are difficult to solve:
  - enumeration size & NP-hard class

Parallel Branch-and-Bound

- COPs are difficult to solve:
  - enumeration size & NP-hard class
- Many studies on parallel approach [Gendron 94, Crainic 06]
  - *node-based*: parallel bounding on sub-problems
  - *tree-based*: building the tree in parallel
  - *multi-search*: several trees are generated in parallel

Parallel Branch-and-Bound

- COPs are difficult to solve:
  - enumeration size & NP-hard class
- Many studies on parallel approach [Gendron 94, Crainic 06]
  - *node-based*: parallel bounding on sub-problems
  - *tree-based*: building the tree in parallel
  - *multi-search*: several trees are generated in parallel
- *Tree-based* is the most studied [Authié 95 , Crainic 06]
  - the solution tree is not known beforehand
  - no part of the tree may be estimated at compilation
  - tasks are dynamically generated
  - task allocations to processors must be done dynamically
  - distributing issues, such as load-balancing & information sharing

Parallel Branch-and-Bound

- COPs are difficult to solve:
  - enumeration size & NP-hard class
- Many studies on parallel approach [Gendron 94, Crainic 06]
  - *node-based*: parallel bounding on sub-problems
  - *tree-based*: building the tree in parallel
  - *multi-search*: several trees are generated in parallel
- *Tree-based* is the most studied [Authié 95 , Crainic 06]
  - the solution tree is not known beforehand
  - no part of the tree may be estimated at compilation
  - tasks are dynamically generated
  - task allocations to processors must be done dynamically
  - distributing issues, such as load-balancing & information sharing
- Sharing a global bound for optimizing the prune operation

Parallel Branch-and-Bound

- COPs are difficult to solve:
  - enumeration size & NP-hard class
- Many studies on parallel approach [Gendron 94, Crainic 06]
  - *node-based*: parallel bounding on sub-problems
  - *tree-based*: building the tree in parallel
  - *multi-search*: several trees are generated in parallel
- *Tree-based* is the most studied [Authié 95 , Crainic 06]
  - the solution tree is not known beforehand
  - no part of the tree may be estimated at compilation
  - tasks are dynamically generated
  - task allocations to processors must be done dynamically
  - distributing issues, such as load-balancing & information sharing
- Sharing a global bound for optimizing the prune operation

Parallel Branch-and-Bound

- COPs are difficult to solve:
  - enumeration size & NP-hard class
- Many studies on parallel approach [Gendron 94, Crainic 06]

Proposition: Parallel BnB + Grid

Tree-based is the most known by users &
Related difficulties are known

  - no part of the tree may be estimated at compilation
  - tasks are dynamically generated
  - task allocations to processors must be done dynamically
  - distributing issues, such as load-balancing & information sharing
- Sharing a global bound for optimizing the prune operation

BnB Related Work

Frameworks	**Algorithms**	**Parallelization**	**Machines**
PUBB	Low-level, Basic B&B	SPMD	Cluster/PVM
BOB++	Low-level, Basic B&B	SPMD	Cluster/MPI
PPBB-Lib	Basic B&B	SPMD	Cluster/PVM
PICO	Basic B&B, Mixed-integer LP	hier. master-worker	Cluster/MPI
MallBa	Low-level, Basic B&B	SPMD	Cluster/MPI
ZRAM	Low-level, Basic B&B	SPMD	Cluster/PVM
ALPS/BiCePS	• Low-level • Basic B&B • Mixed-integer LP • Branch&Price&Cut	hier. master-worker	Cluster/MPI
Metacomputing MW	Basic B&B	master-worker	Grids/Condor
Symphony	Mixed-integer LP, Branch&Price&Cut	master-worker	Cluster/PVM

BnB Related Work

Frameworks	Algorithms	Parallelization	Machines
PUBB	Low-level, Basic B&B	SPMD	Cluster/PVM
BOB++	Low-level, Basic B&B	SPMD	Cluster/MPI
PPBB-Lib	Basic B&B	SPMD	Cluster/PVM
PICO	Basic B&B, Mixed-integer LP	hier. master-worker	Cluster/MPI
MallBa	Low-level, Basic B&B	SPMD	Cluster/MPI
ZRAM	Low-level, Basic B&B	SPMD	Cluster/PVM
ALPS/BiCePS	•Low-level •Basic B&B •Mixed-integer LP •Branch&Price&Cut	hier. master-worker	Cluster/MPI
Metacomputing MW	Basic B&B	master-worker	Grids/Condor
Symphony	Mixed-integer LP, Branch&Price&Cut	master-worker	Cluster/PVM

Most Previous work are based on SPMD and target clusters

better for sharing bounds

Grid Computing

Grid Computing

- Distributed shared computing infrastructure
  - multi-institutional virtual organization

Grid Computing

- Distributed shared computing infrastructure
    - multi-institutional virtual organization
- Provide large pool of resources

Grid Computing

- Distributed shared computing infrastructure
  - multi-institutional virtual organization
- Provide large pool of resources
- New challenges
  - geographically distributed
  - deployment
  - scalability
  - communication
  - fault-tolerance
  - multiple administrative domains, heterogeneity, high performance, programming model, *etc.*

Grid Computing

- Distributed shared computing infrastructure
  - multi-institutional virtual organization
- Provide large pool of resources
- New challenges
  - geographically distributed
  - deployment
  - scalability
  - communication
  - fault-tolerance
  - multiple administrative domains, heterogeneity, high performance, programming model, *etc.*

Parallel BnB related work: SPMD

Grids are not adapted for SPMD (heterogeneity, latency, *etc.*)

Grid Computing

Grids involve new concepts for development & execution of applications

Grid Computing

Grids involve new concepts for development & execution of applications

Grid Fabric

Schedulers, Networking, OS
Federated Hardware Resources

Grid Computing

Grids involve new concepts for development & execution of applications

Grid Middleware Infrastructure

Super-Schedulers, Resource Trading, Information, Security, *etc.*

Grid Fabric

Schedulers, Networking, OS
Federated Hardware Resources

Grid Computing

Grids involve new concepts for development & execution of applications

Grid Programming	Models, Tools, High-Level Access to Middleware
Grid Middleware Infrastructure	Super-Schedulers, Resource Trading, Information, Security, *etc.*
Grid Fabric	Schedulers, Networking, OS Federated Hardware Resources

Grid Computing

Grids involve new concepts for development & execution of applications

Grid Applications & Portals	
Grid Programming	Models, Tools, High-Level Access to Middleware
Grid Middleware Infrastructure	Super-Schedulers, Resource Trading, Information, Security, *etc.*
Grid Fabric	Schedulers, Networking, OS Federated Hardware Resources

Grid Computing

Grids involve new concepts for development & execution of applications

Grid Applications & Portals

Grid Programming ← Branch-and-Bound API

Grid Middleware Infrastructure — Super-Schedulers, Resource Trading, Information, Security, *etc.*

Grid Fabric — Schedulers, Networking, OS
Federated Hardware Resources

Challenges

Combinatorial Optimization Problems

costly to solve

Challenges

Combinatorial Optimization Problems
costly to solve

Branch-and-Bound
adapted for solving COPs
easy to parallelize

Challenges

Combinatorial Optimization Problems
costly to solve

Branch-and-Bound
adapted for solving COPs
easy to parallelize

Grid Computing
huge number of resources

Challenges

Combinatorial Optimization Problems
costly to solve

Branch-and-Bound
adapted for solving COPs
easy to parallelize

+

Grid Computing
huge number of resources

Use Branch-and-Bound on Grids for solving COPs

Challenges

Combinatorial Optimization Problems
costly to solve

Branch-and-Bound
adapted for solving COPs
easy to parallelize

+

Grid Computing
huge number of resources

Use Branch-and-Bound on Grids for solving COPs

Efficient communications with Grids is difficult
Problem with sharing bounds

Agenda

- Context, Problem, and Related Work
- Contributions
  - Branch-and-Bound Framework for Grids
  - Desktop Grid with Peer-to-Peer
  - Mixing Desktops & Clusters
- Perspectives & Conclusion

BnB for Grids Related Work

BnB for Grids Related Work

- Aida *et al.* focus on the design:
  - hierarchical master-worker scales on Grids

BnB for Grids Related Work

- Aida *et al.* focus on the design:
  - hierarchical master-worker scales on Grids
- Foster *et al.* fully decentralized approach:
  - communication overhead

BnB for Grids Related Work

- Aida *et al.* focus on the design:
  - hierarchical master-worker scales on Grids
- Foster *et al.* fully decentralized approach:
  - communication overhead
- ParadisEO focus on meta-heuristics (not exact solution):
  - master-worker

BnB for Grids Related Work

- Aida *et al.* focus on the design:
  - hierarchical master-worker scales on Grids
- Foster *et al.* fully decentralized approach:
  - communication overhead
- ParadisEO focus on meta-heuristics (not exact solution):
  - master-worker

- Skeletons with farm or divide-and-conquer
- Satin for divide-and-conquer

BnB on Grids: Problems and Solutions

BnB on Grids: Problems and Solutions

BnB on Grids: Problems and Solutions

Latency	

BnB on Grids: Problems and Solutions

Latency	Asynchronous communications

BnB on Grids: Problems and Solutions

Latency	Asynchronous communications
Scalability	

BnB on Grids: Problems and Solutions

Latency	Asynchronous communications
Scalability	Hierarchical master-worker

BnB on Grids: Problems and Solutions

Latency	Asynchronous communications
Scalability	Hierarchical master-worker
Solution tree size	

BnB on Grids: Problems and Solutions

Latency	Asynchronous communications
Scalability	Hierarchical master-worker
Solution tree size	Dynamically generated by splitting tasks

BnB on Grids: Problems and Solutions

Latency	Asynchronous communications
Scalability	Hierarchical master-worker
Solution tree size	Dynamically generated by splitting tasks
Share the best bounds	

BnB on Grids: Problems and Solutions

Latency	Asynchronous communications
Scalability	Hierarchical master-worker
Solution tree size	Dynamically generated by splitting tasks
Share the best bounds	Efficient parallelism & communication

BnB on Grids: Problems and Solutions

Latency	Asynchronous communications
Scalability	Hierarchical master-worker
Solution tree size	Dynamically generated by splitting tasks
Share the best bounds	Efficient parallelism & communication
Faults	

BnB on Grids: Problems and Solutions

Latency	Asynchronous communications
Scalability	Hierarchical master-worker
Solution tree size	Dynamically generated by splitting tasks
Share the best bounds	Efficient parallelism & communication
Faults	Fault-tolerance

BnB on Grids: Problems and Solutions

Latency	Asynchronous communications
Scalability	Hierarchical master-
	ed by splitting tasks
Share the best bounds	Efficient parallelism & communication
Faults	Fault-tolerance

Objective: hide Grid difficulties to users
Especially communication problems

BnB Framework - Entities and Roles

BnB Framework - Entities and Roles

- Root Task:
  - implemented by users
  - *objective-function* and splitting/branching operation

BnB Framework - Entities and Roles

- Root Task:
  - implemented by users
  - *objective-function* and splitting/branching operation
- Master: Entry Point
  - splits the problem in tasks
  - collects partial-results ➟ the best solution

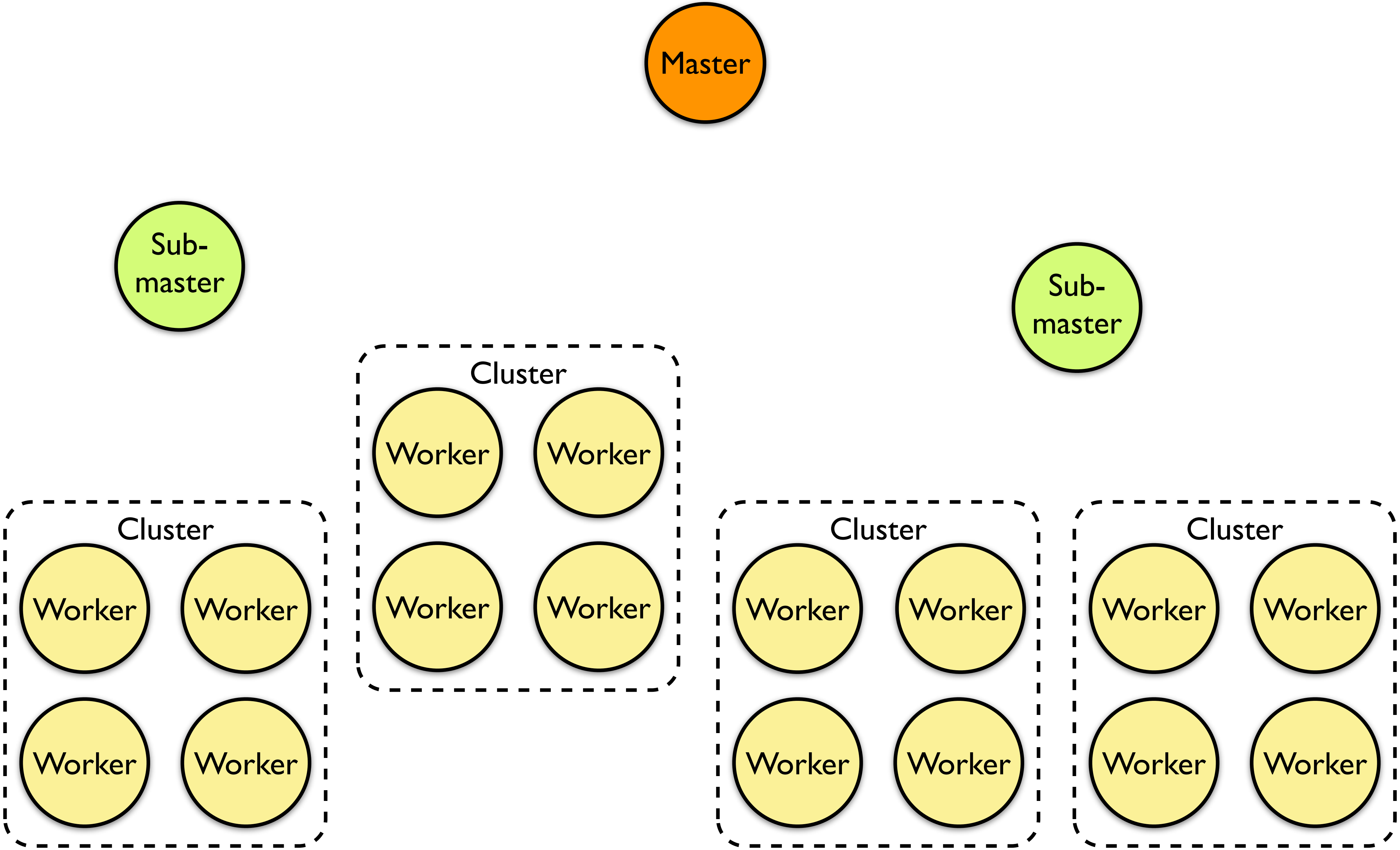
BnB Framework - Entities and Roles

- Root Task:
  - implemented by users
  - *objective-function* and splitting/branching operation
- Master: Entry Point
  - splits the problem in tasks
  - collects partial-results ➟ the best solution
- Sub-Master:
  - intermediary between master and workers

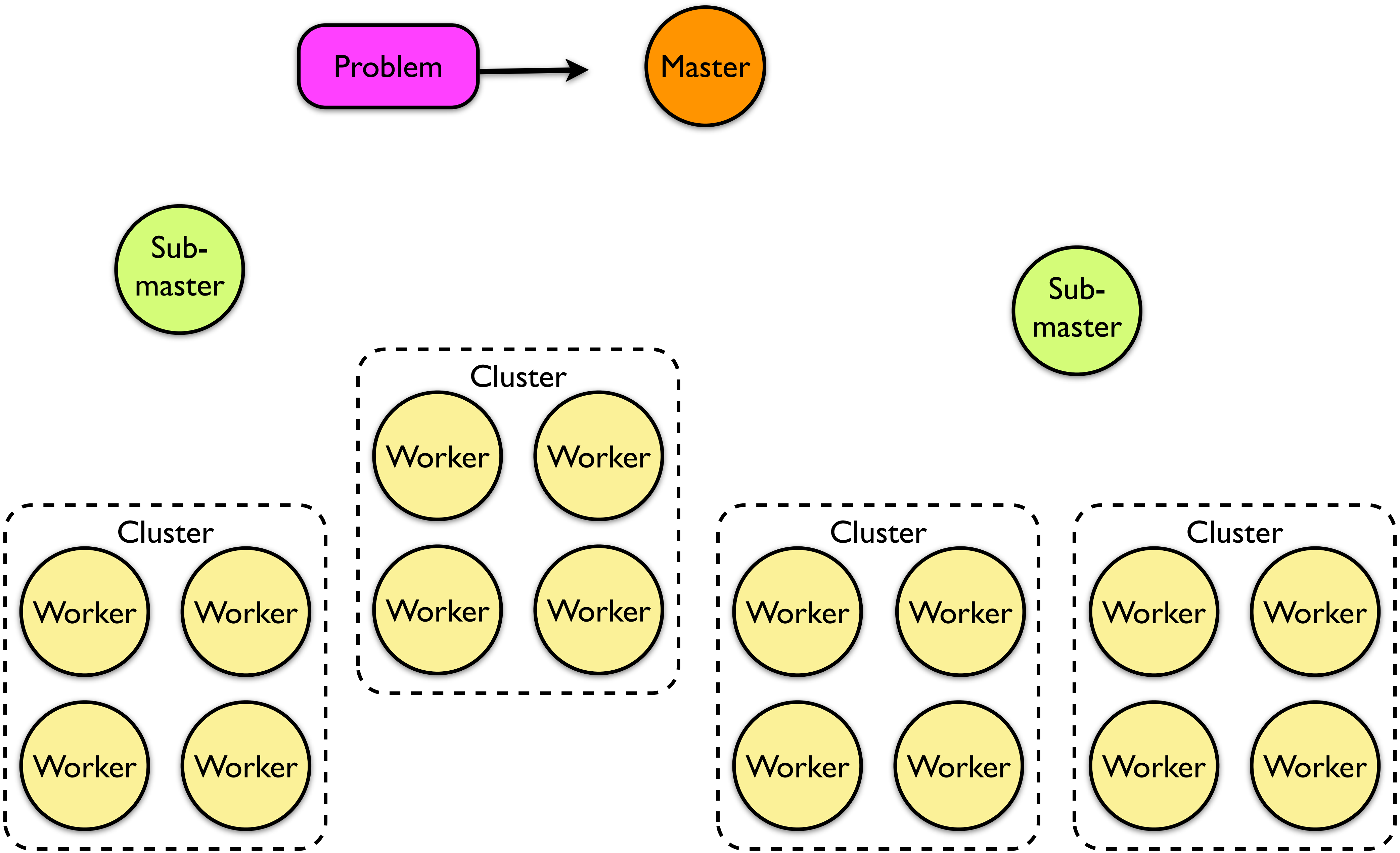
BnB Framework - Entities and Roles

- Root Task:
  - implemented by users
  - *objective-function* and splitting/branching operation
- Master: Entry Point
  - splits the problem in tasks
  - collects partial-results ➟ the best solution
- Sub-Master:
  - intermediary between master and workers
- Worker:
  - computes tasks

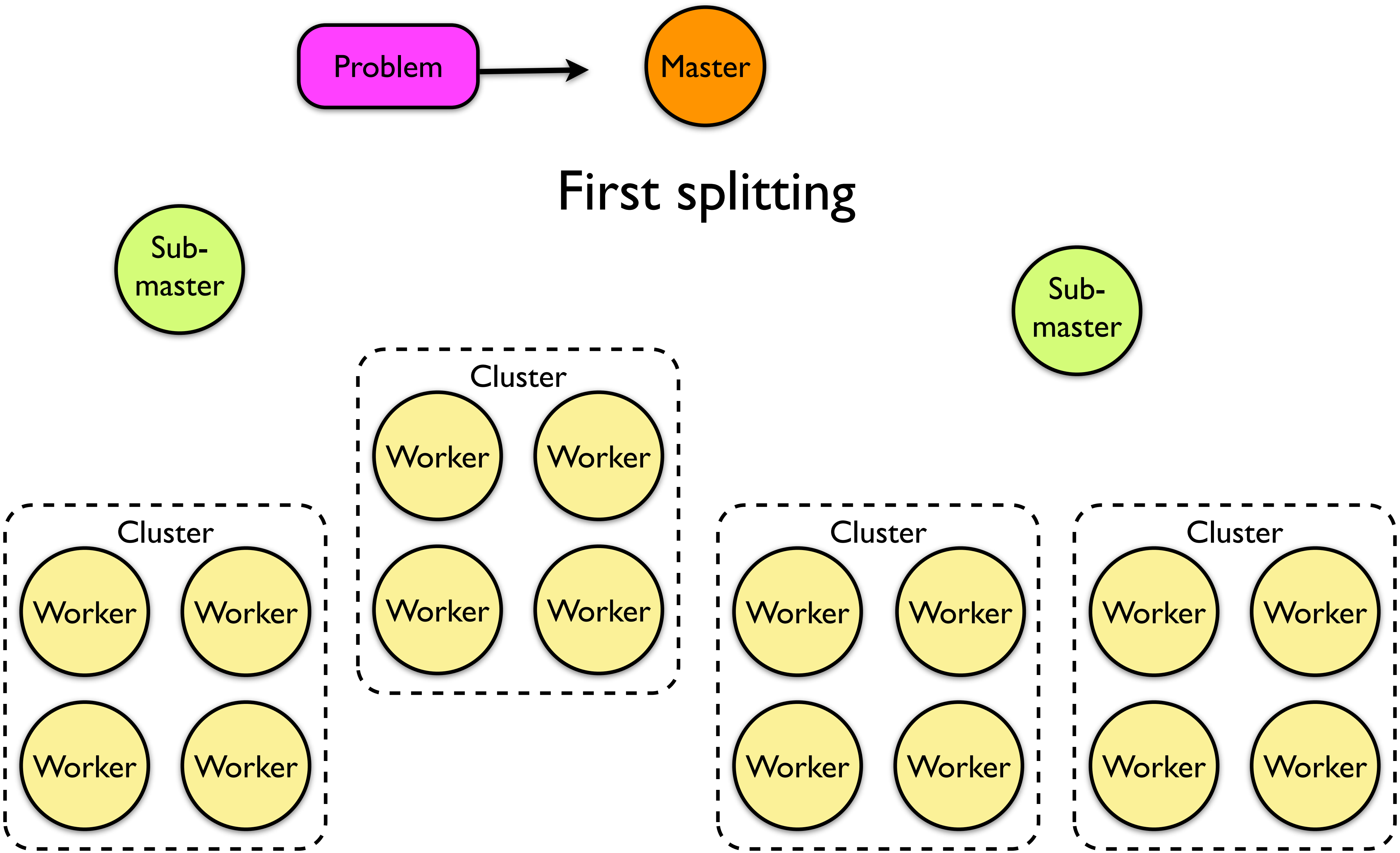
BnB Framework - Architecture



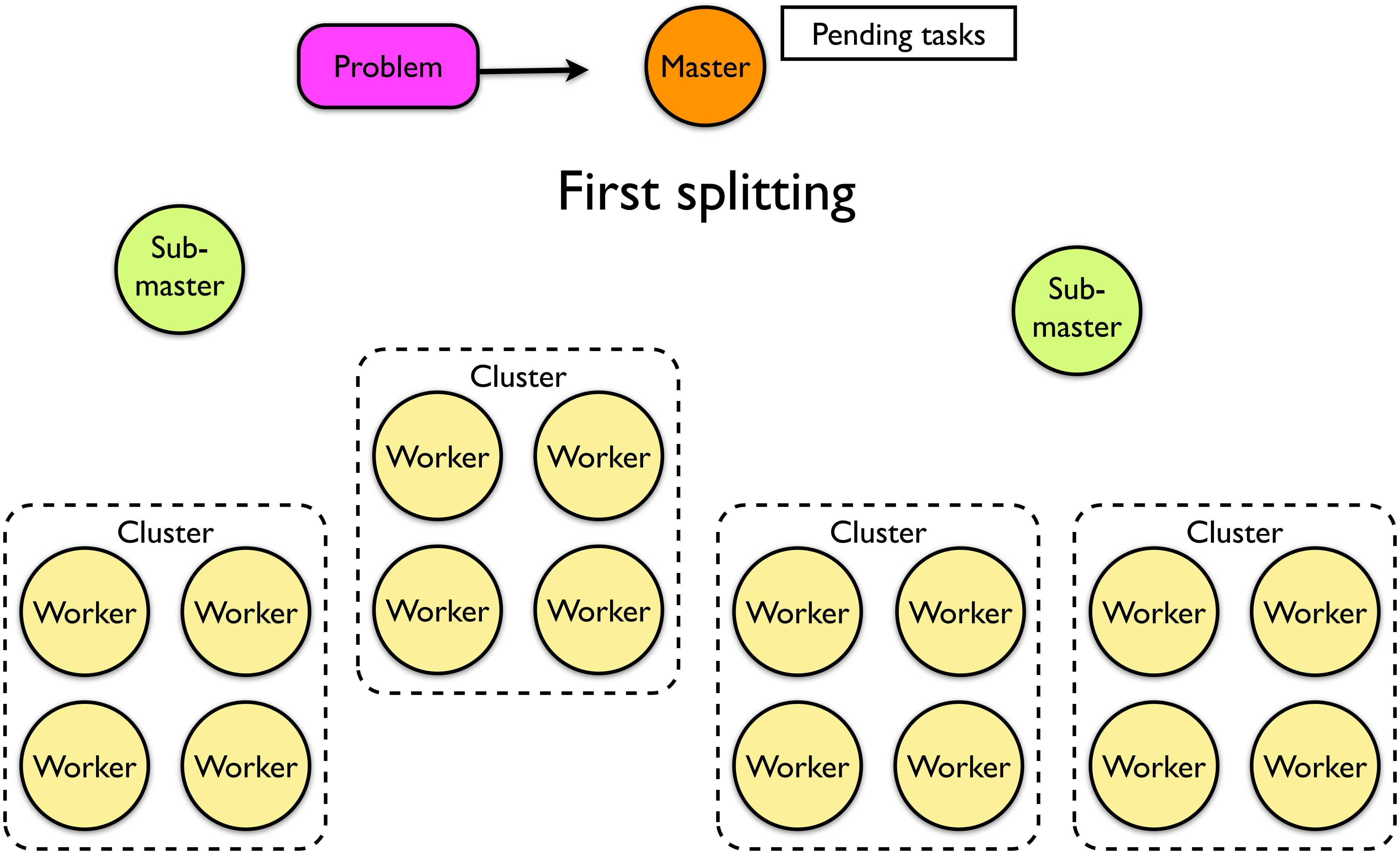
BnB Framework - Architecture



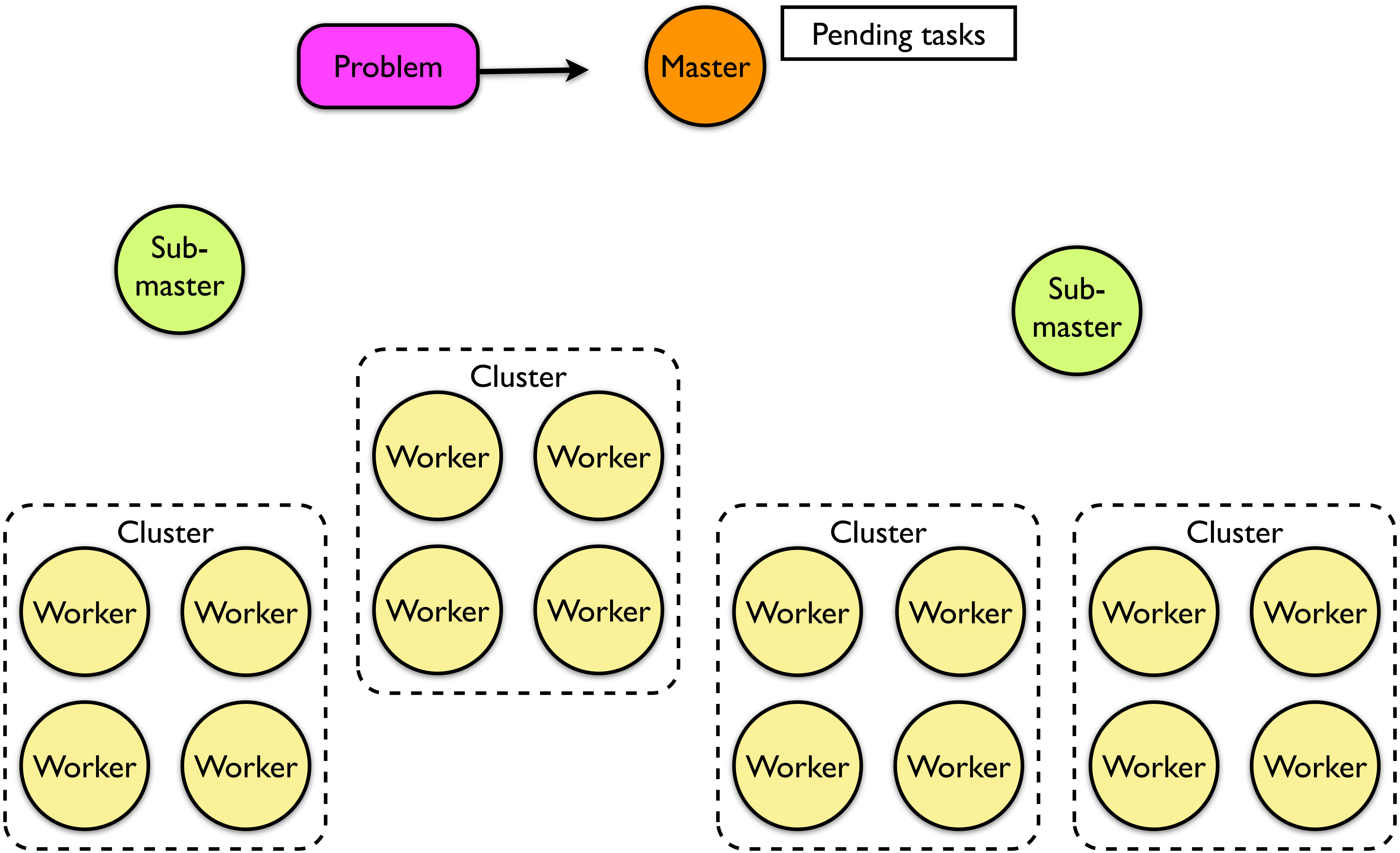
BnB Framework - Architecture



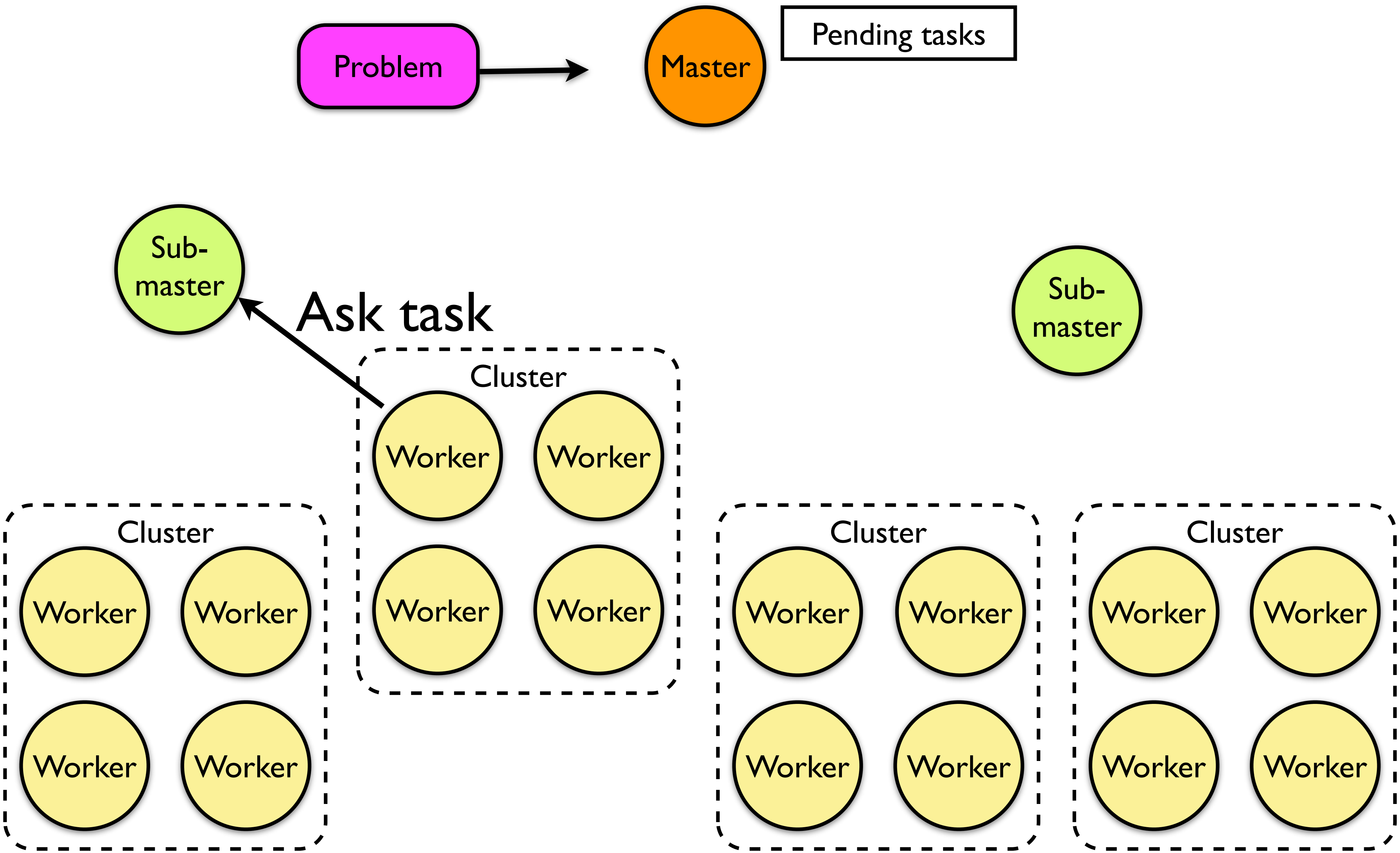
BnB Framework - Architecture



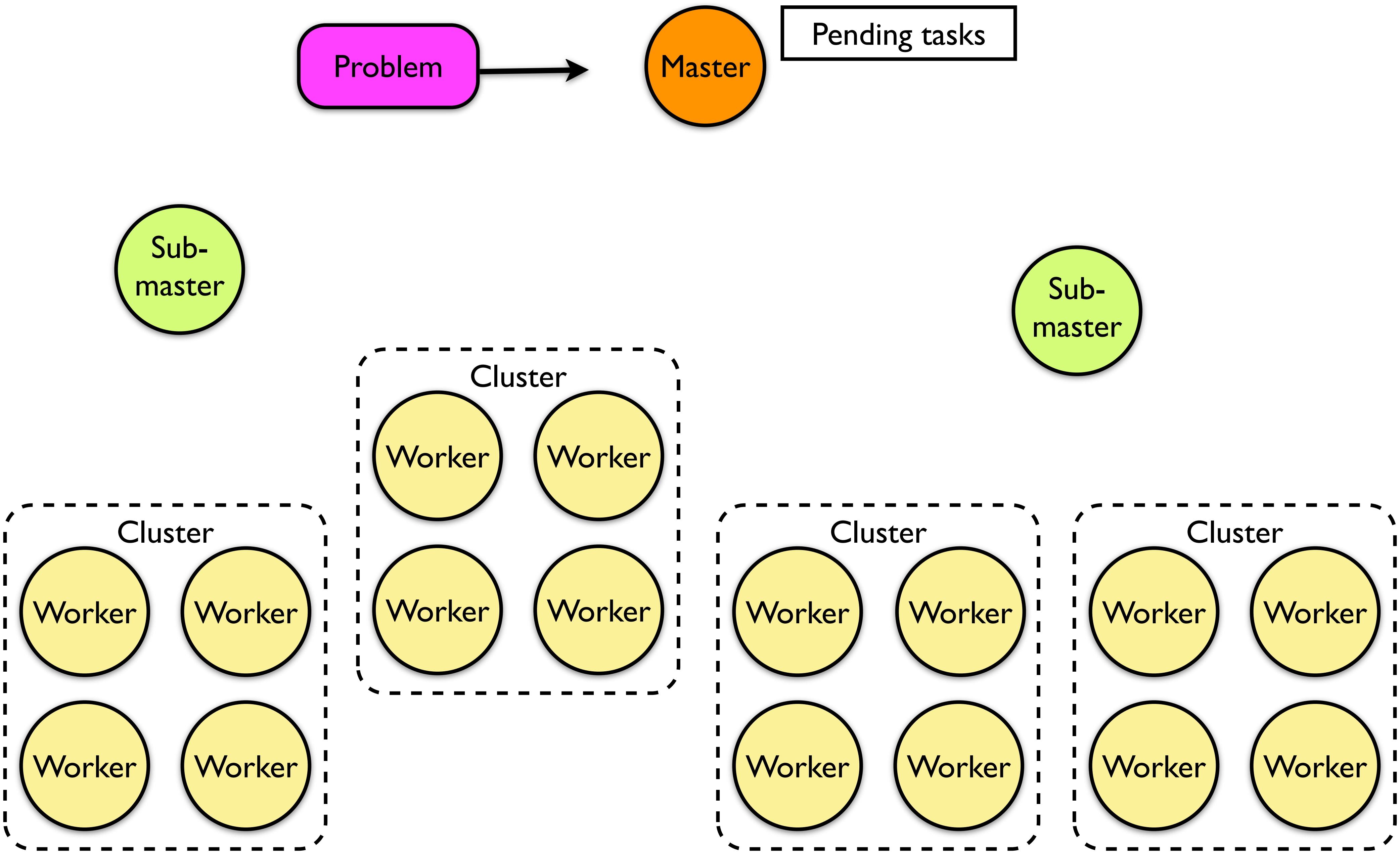
BnB Framework - Architecture



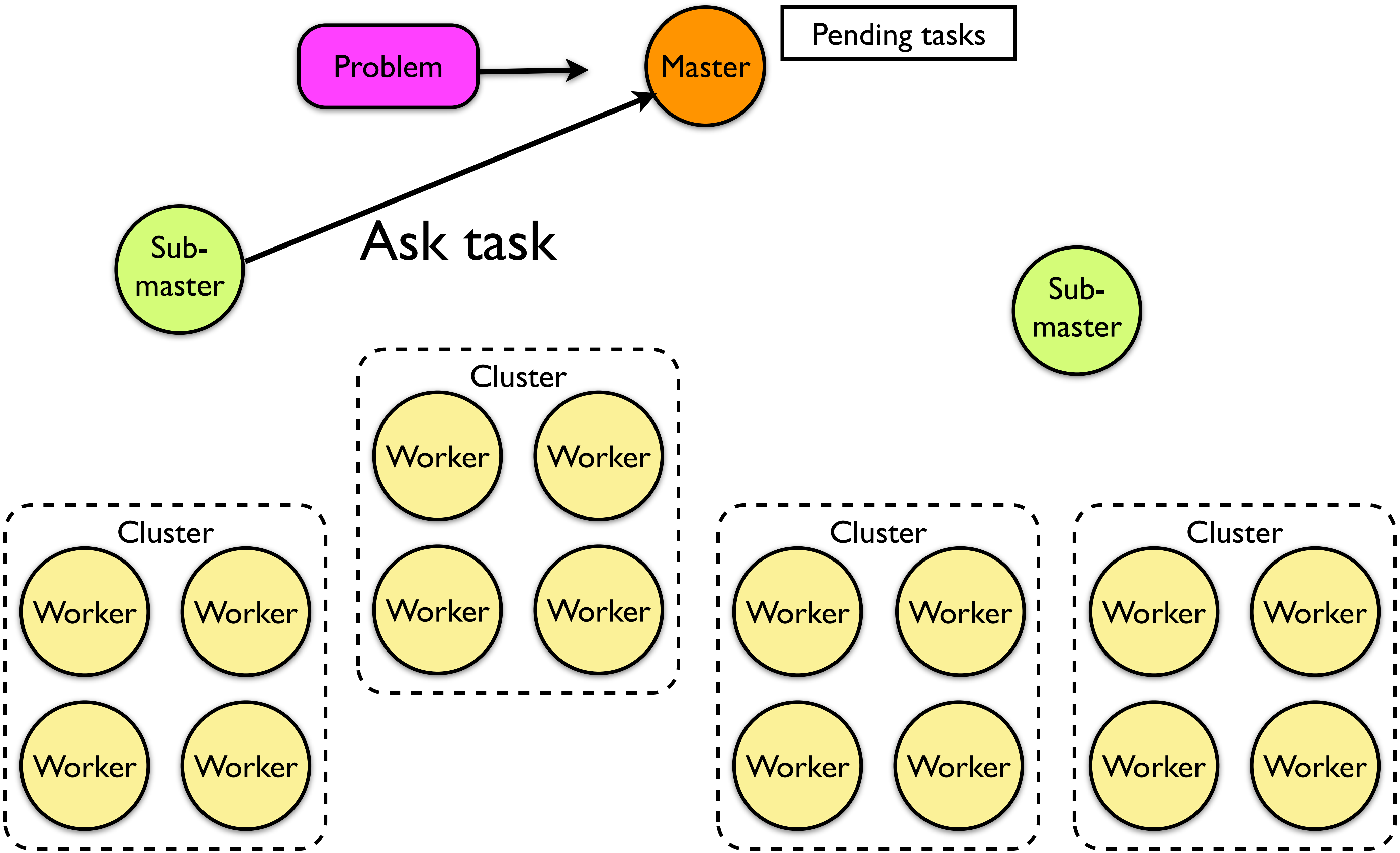
BnB Framework - Architecture



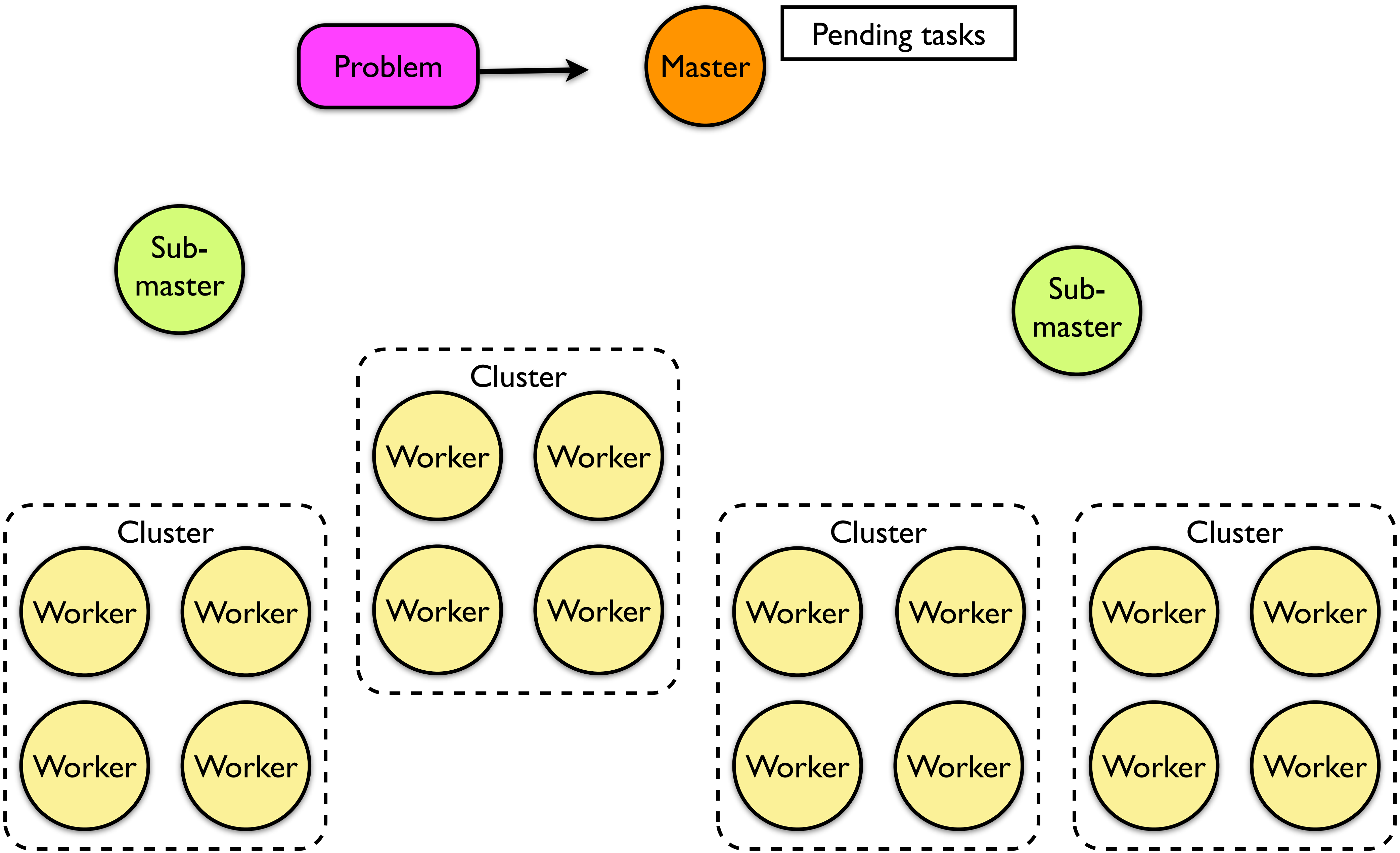
BnB Framework - Architecture



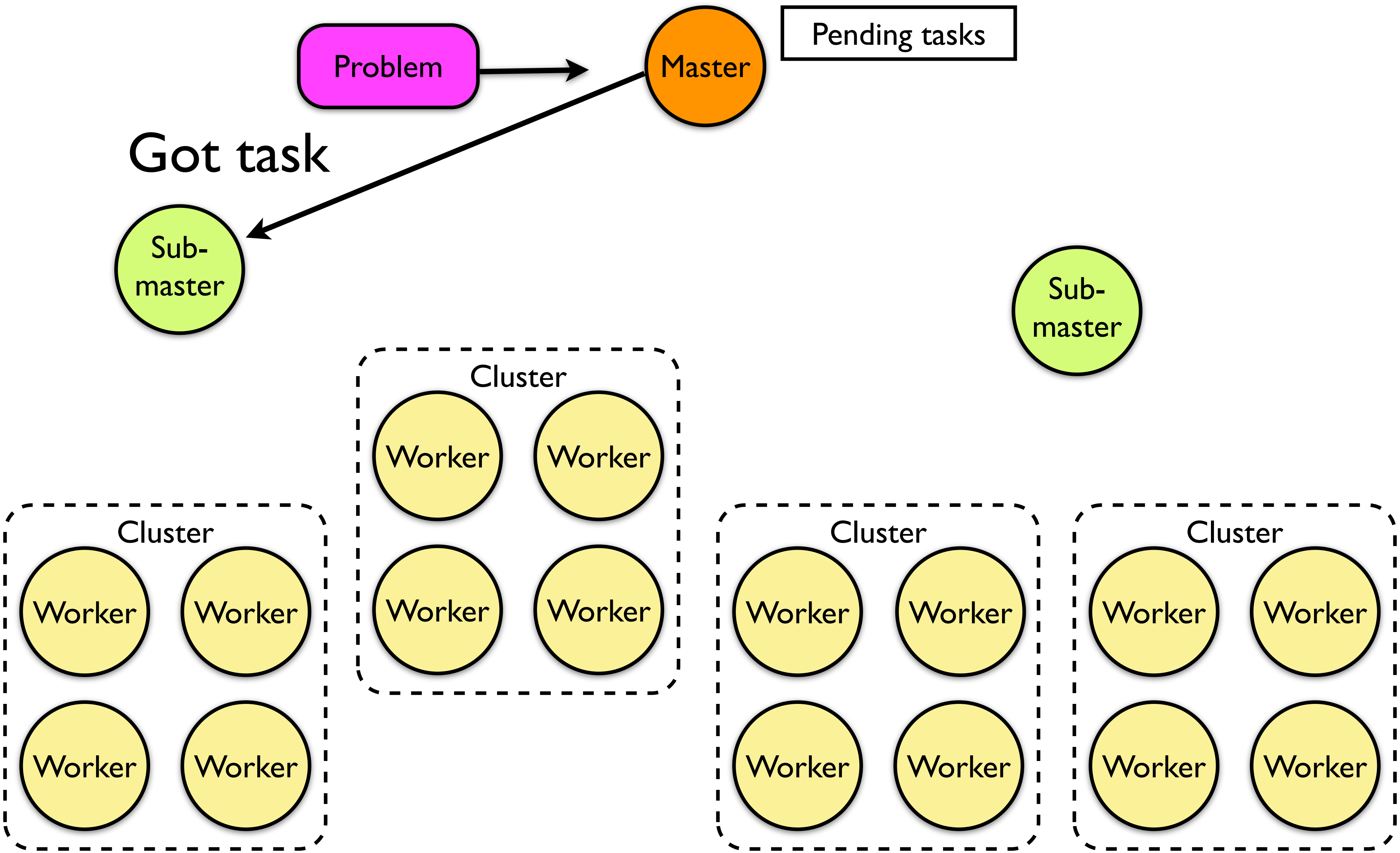
BnB Framework - Architecture



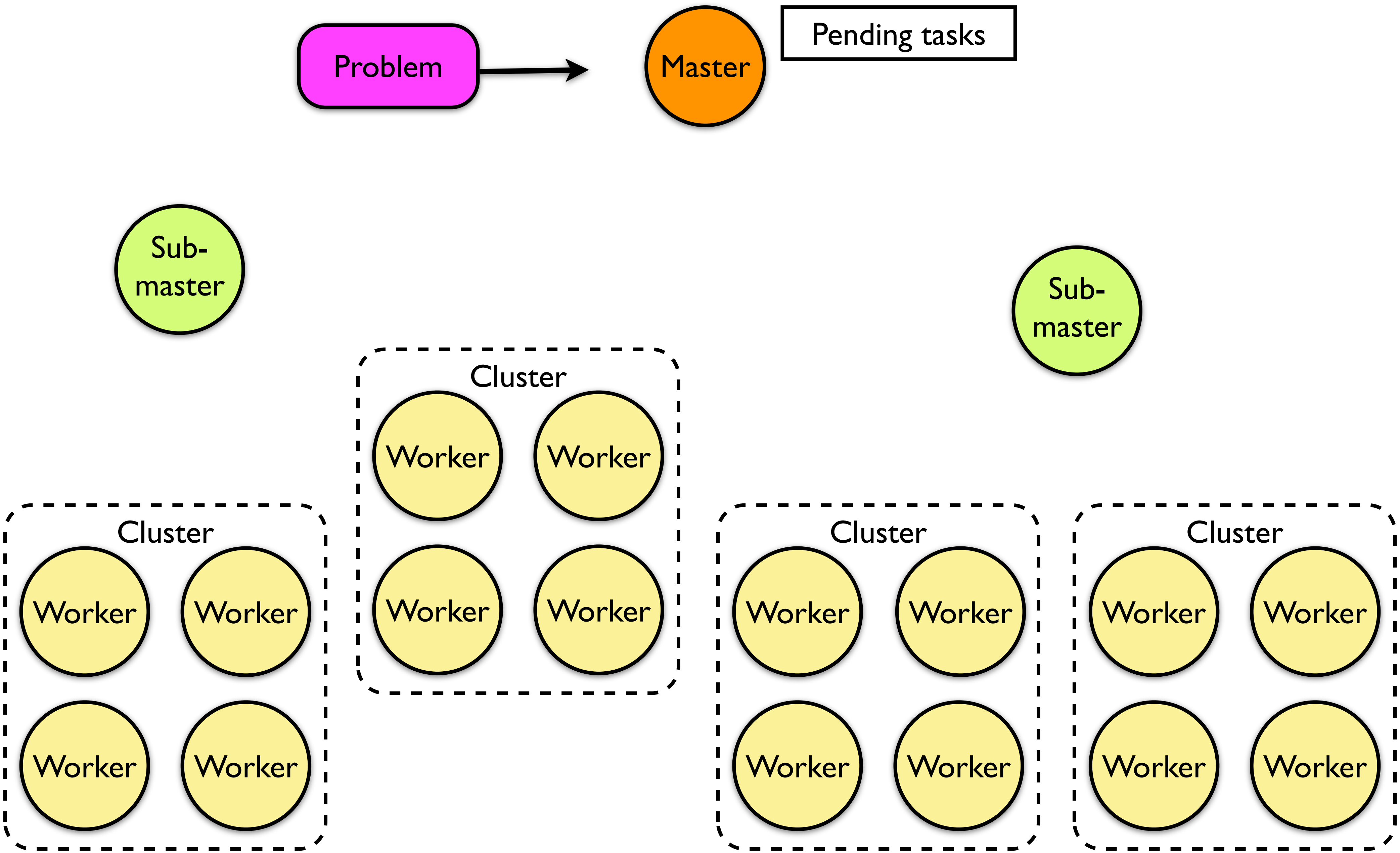
BnB Framework - Architecture



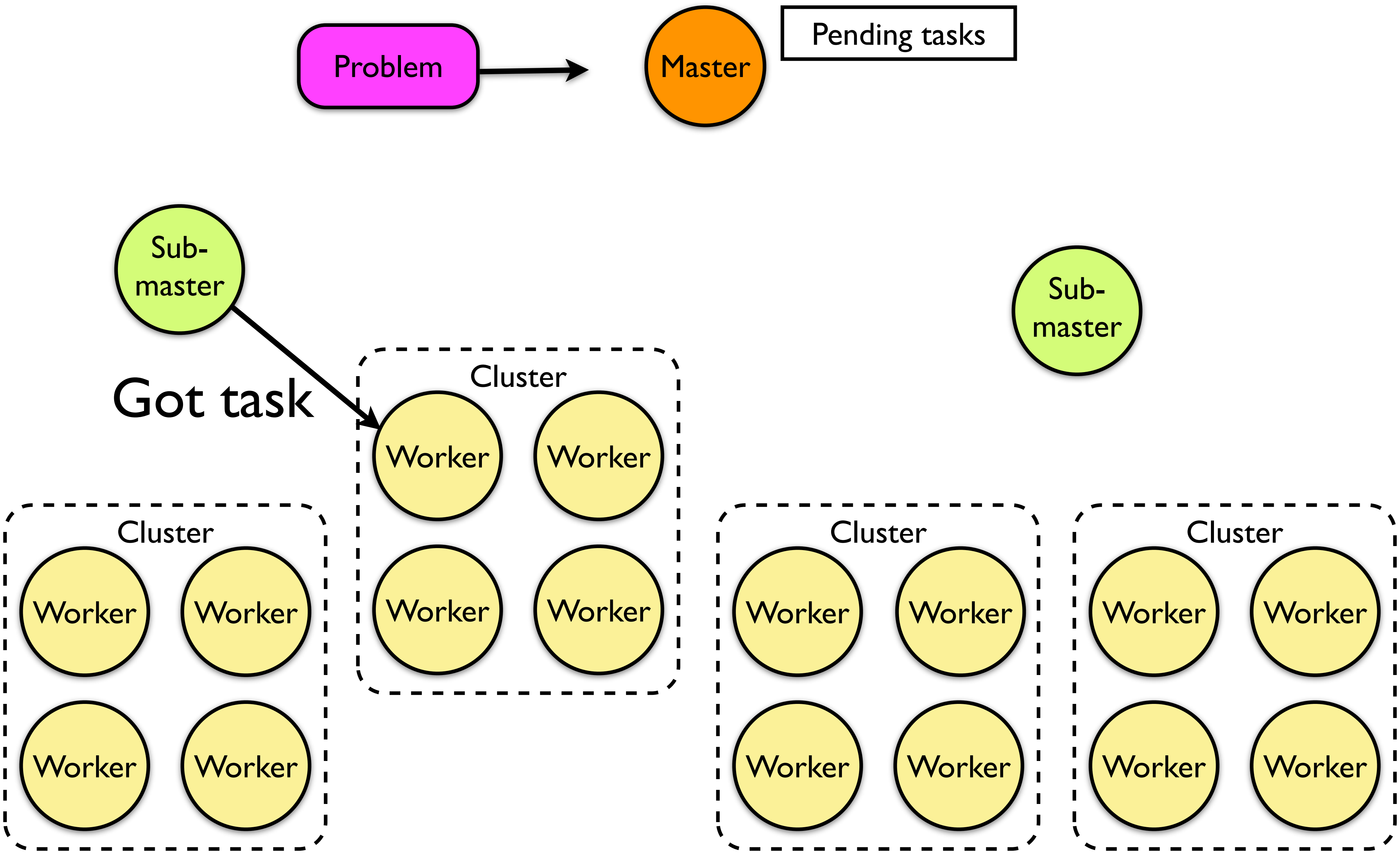
BnB Framework - Architecture



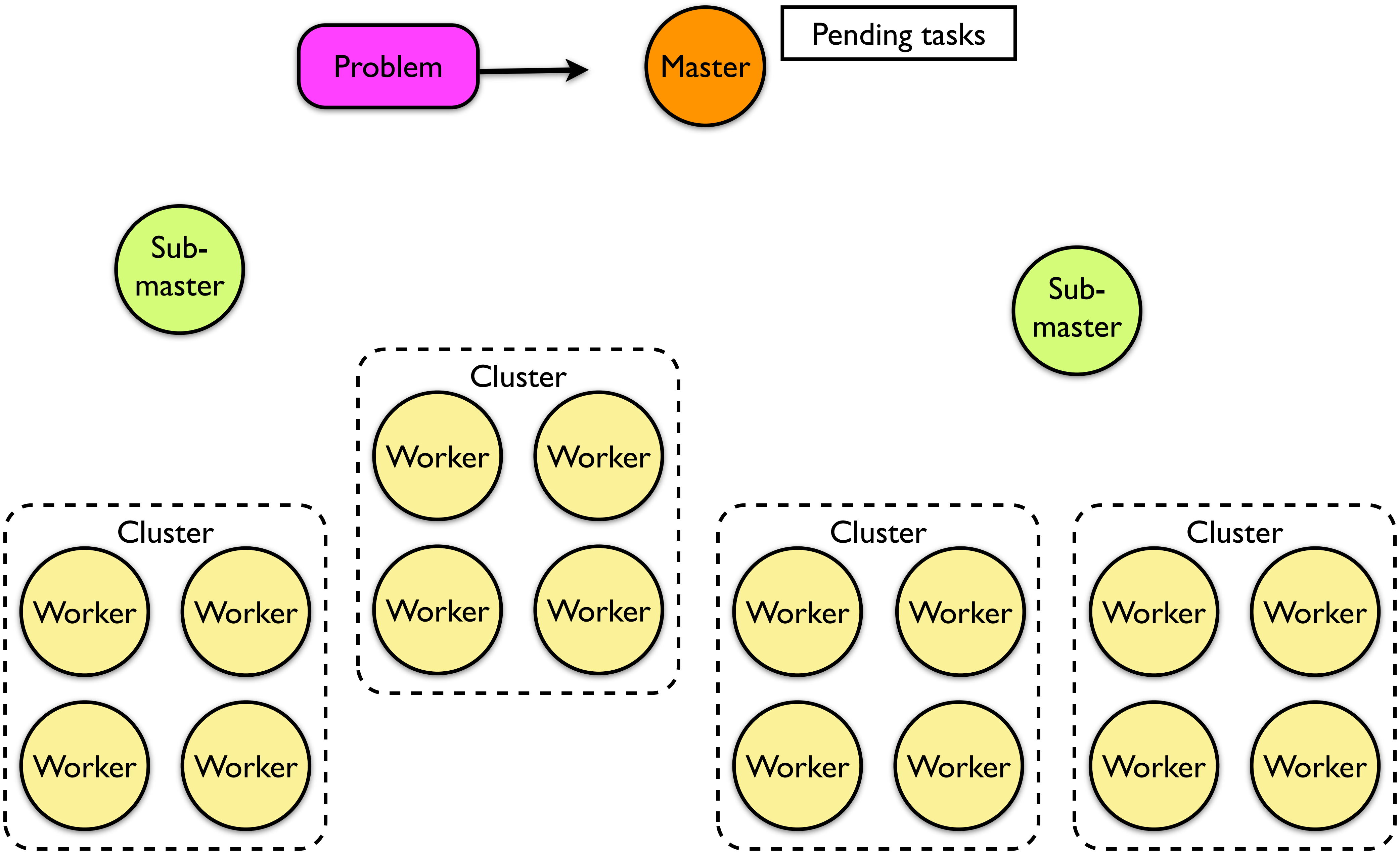
BnB Framework - Architecture



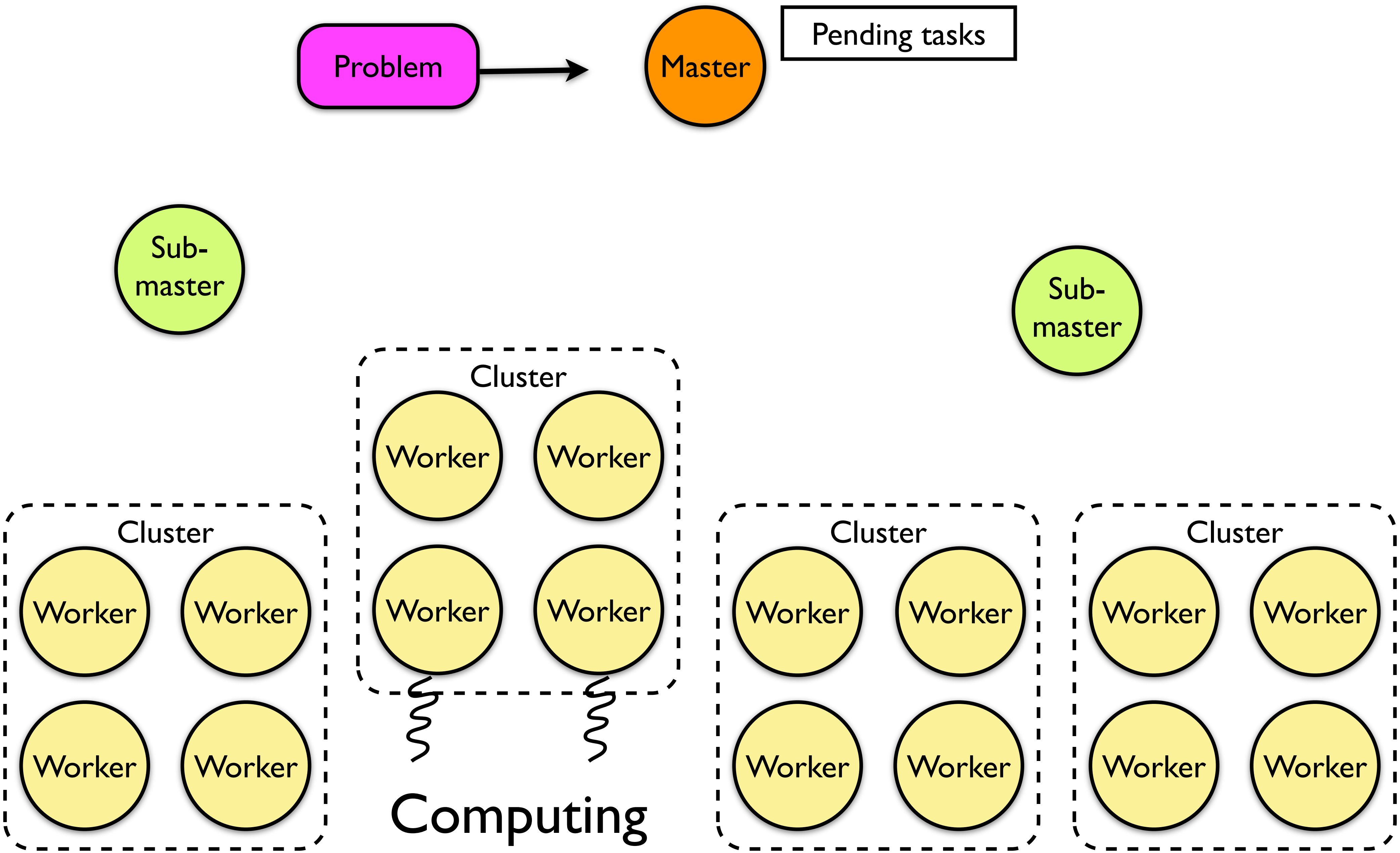
BnB Framework - Architecture



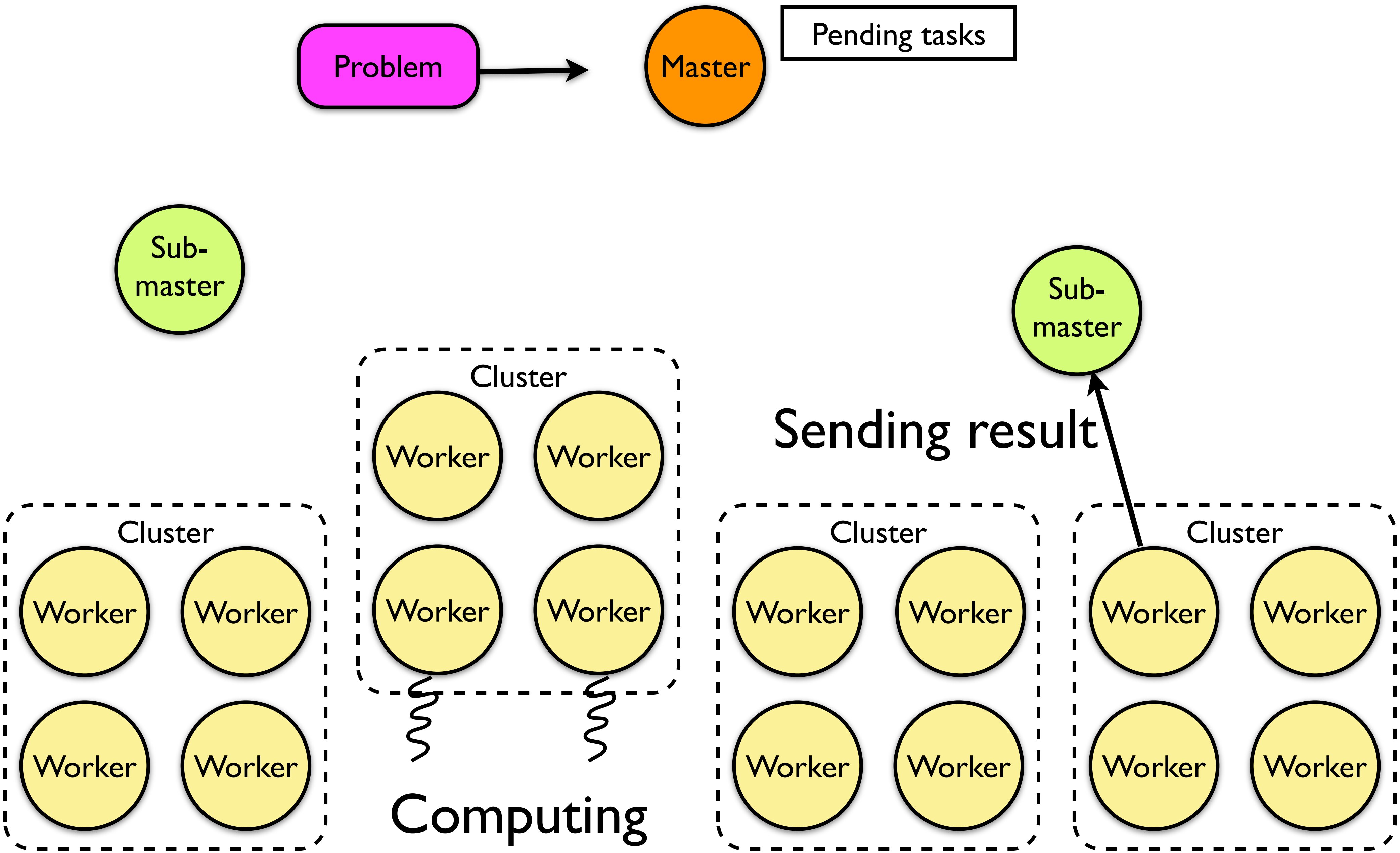
BnB Framework - Architecture



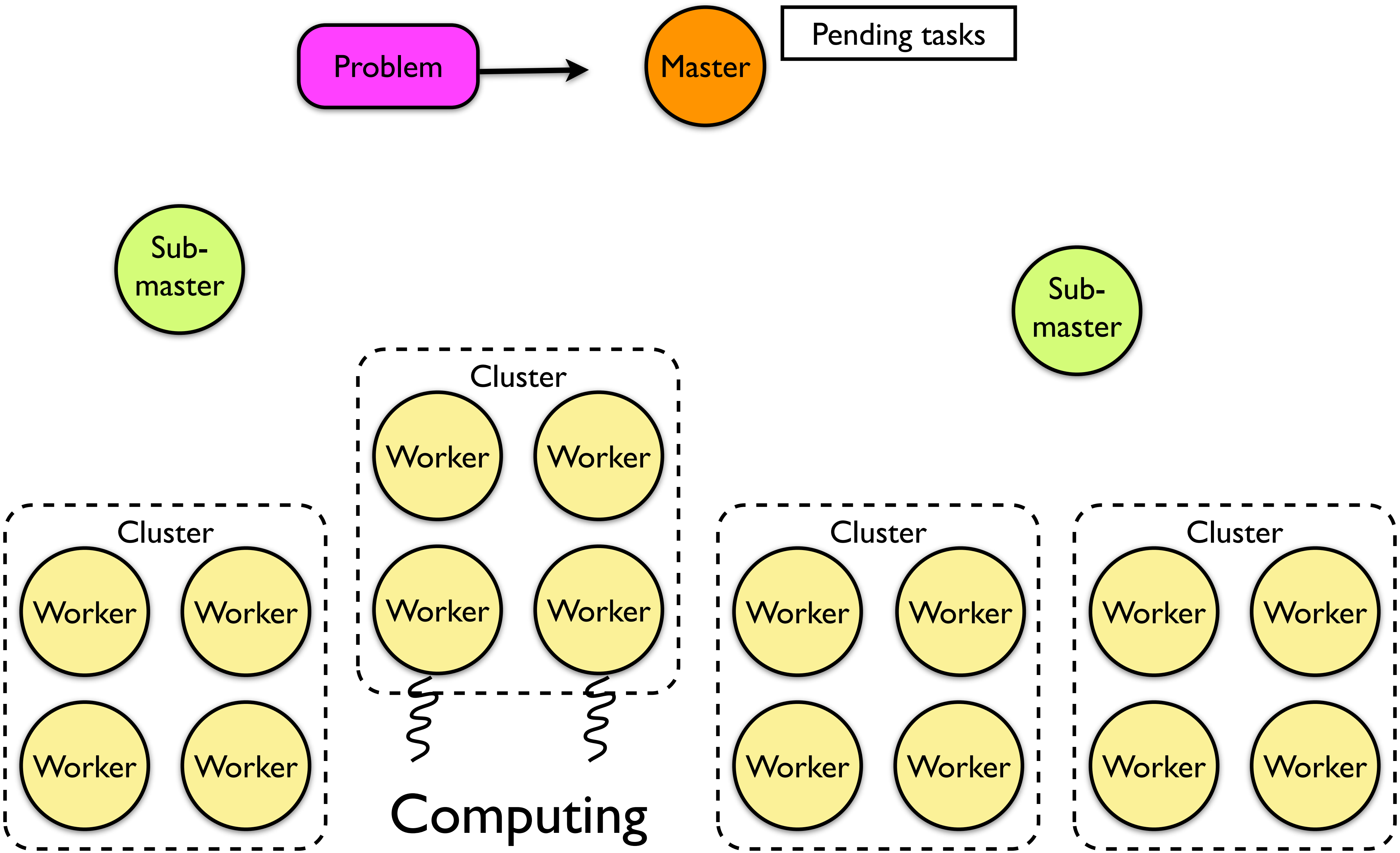
BnB Framework - Architecture



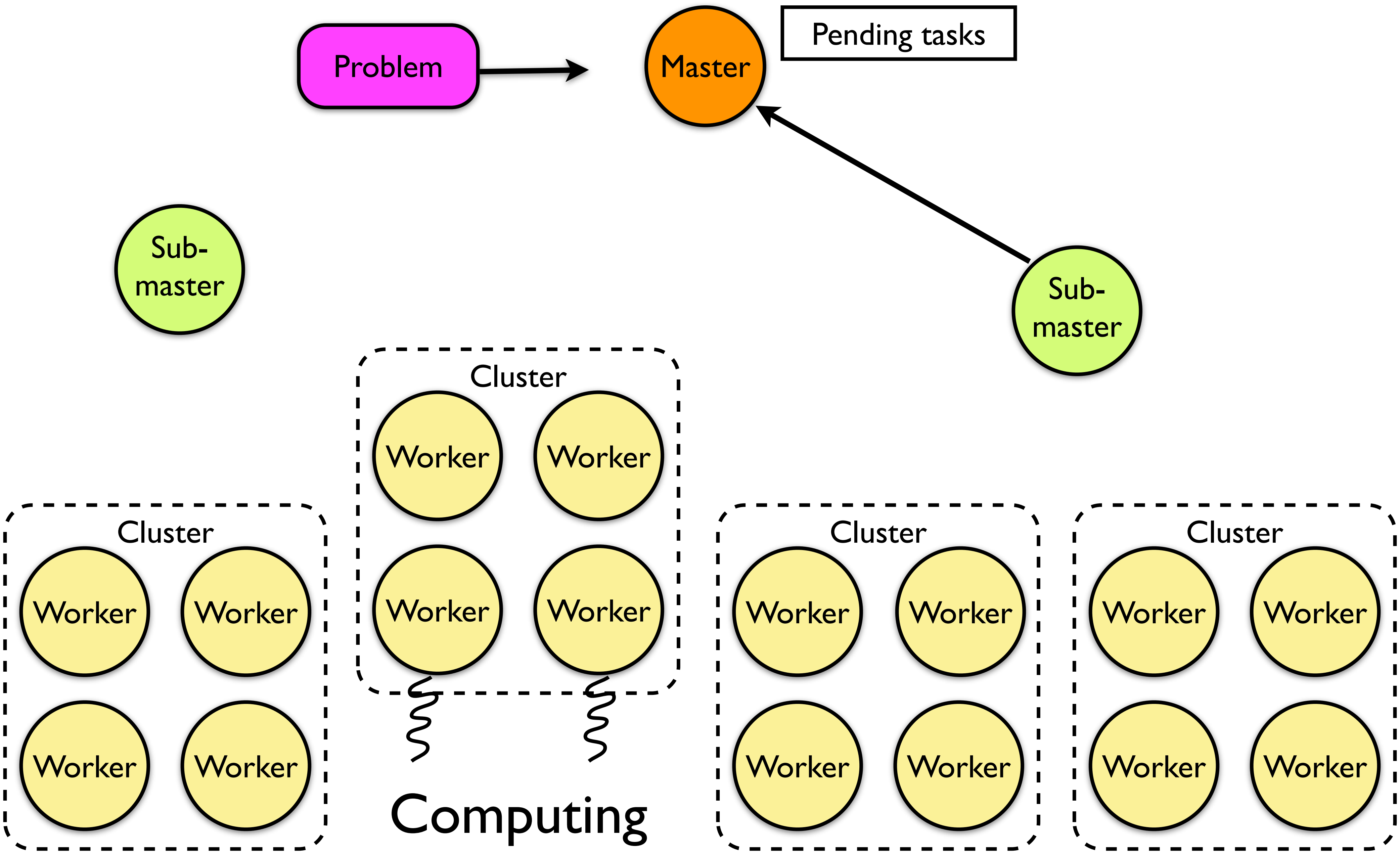
BnB Framework - Architecture



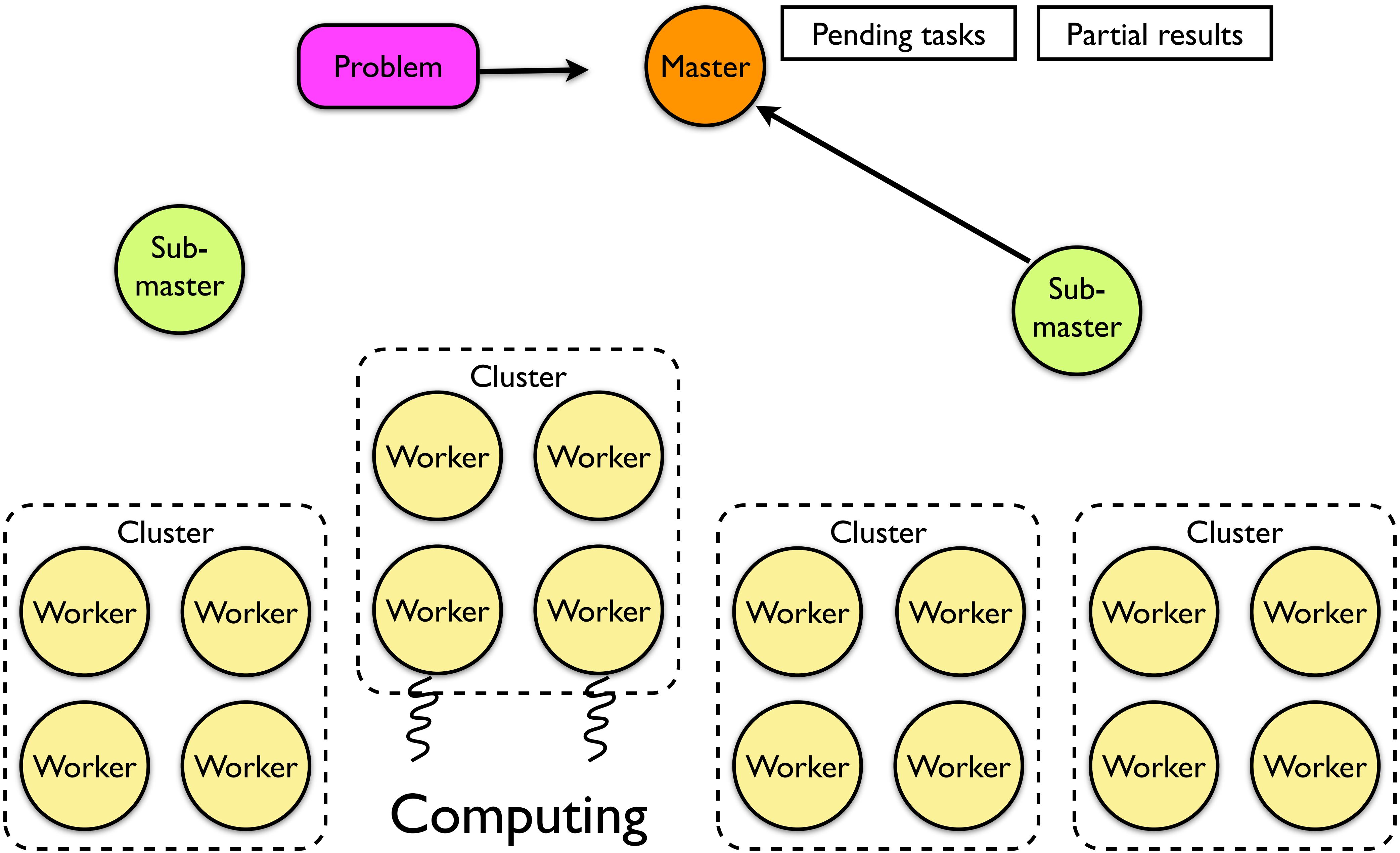
BnB Framework - Architecture



BnB Framework - Architecture



BnB Framework - Architecture



BnB Framework - Solutions

BnB Framework - Solutions

- Context ProActive Java Grid middleware:
  - latency ➟ asynchronous communication
  - underlaying Grid infrastructure ➟ deployment framework (abstraction)

BnB Framework - Solutions

- Context ProActive Java Grid middleware:
  - latency ➟ asynchronous communication
  - underlaying Grid infrastructure ➟ deployment framework (abstraction)
- Implement the tree-based parallel
- Master-worker architecture
- Problem: workers need to share bounds
  - difficult to adapt SPMD for Grids (heterogeneity, distribution, *etc.*)

BnB Framework - Solutions

- Context ProActive Java Grid middleware:
  - latency ➟ asynchronous communication
  - underlaying Grid infrastructure ➟ deployment framework (abstraction)
- Implement the tree-based parallel
- Master-worker architecture
- Problem: workers need to share bounds
  - difficult to adapt SPMD for Grids (heterogeneity, distribution, *etc.*)
  - Solution 1: Master keeps the bound

BnB Framework - Solutions

- Context ProActive Java Grid middleware:
  - latency ➟ asynchronous communication
  - underlaying Grid infrastructure ➟ deployment framework (abstraction)
- Implement the tree-based parallel
- Master-worker architecture
- Problem: workers need to share bounds
  - difficult to adapt SPMD for Grids (heterogeneity, distribution, *etc.*)
  - Solution 1: Master keeps the bound
    - previous work shows that not scale [Aida 2003]

BnB Framework - Solutions

- Context ProActive Java Grid middleware:
  - latency ➟ asynchronous communication
  - underlaying Grid infrastructure ➟ deployment framework (abstraction)
- Implement the tree-based parallel
- Master-worker architecture
- Problem: workers need to share bounds
  - difficult to adapt SPMD for Grids (heterogeneity, distribution, *etc.*)
  - Solution 1: Master keeps the bound
    - previous work shows that not scale [Aida 2003]
  - Solution 2: Message framework (Enterprise Service Bus)

BnB Framework - Solutions

- Context ProActive Java Grid middleware:
  - latency ➟ asynchronous communication
  - underlaying Grid infrastructure ➟ deployment framework (abstraction)
- Implement the tree-based parallel
- Master-worker architecture
- Problem: workers need to share bounds
  - difficult to adapt SPMD for Grids (heterogeneity, distribution, *etc.*)
  - Solution 1: Master keeps the bound
    - previous work shows that not scale [Aida 2003]
  - Solution 2: Message framework (Enterprise Service Bus)
    - Grid middleware dependent / Good for SOA

BnB Framework - Solutions

- Context ProActive Java Grid middleware:
  - latency ➟ asynchronous communication
  - underlaying Grid infrastructure ➟ deployment framework (abstraction)
- Implement the tree-based parallel
- Master-worker architecture
- Problem: workers need to share bounds
  - difficult to adapt SPMD for Grids (heterogeneity, distribution, *etc.*)
  - Solution 1: Master keeps the bound
    - previous work shows that not scale [Aida 2003]
  - Solution 2: Message framework (Enterprise Service Bus)
    - Grid middleware dependent / Good for SOA
  - Solution 3: Broadcasting

BnB Framework - Solutions

- Context ProActive Java Grid middleware:
  - latency ➟ asynchronous communication
  - underlaying Grid infrastructure ➟ deployment framework (abstraction)
- Implement the tree-based parallel
- Master-worker architecture
- Problem: workers need to share bounds
  - difficult to adapt SPMD for Grids (heterogeneity, distribution, *etc.*)
  - Solution 1: Master keeps the bound
    - previous work shows that not scale [Aida 2003]
  - Solution 2: Message framework (Enterprise Service Bus)
    - Grid middleware dependent / Good for SOA
  - Solution 3: Broadcasting
    - 1 to n communication cannot scale

BnB Framework - Solutions

- Context ProActive Java Grid middleware:
  - latency ➟ asynchronous communication
  - underlaying Grid infrastructure ➟ deployment framework (abstraction)
- Implement the tree-based parallel
- Master-worker architecture
- Problem: workers need to share bounds
  - difficult to adapt SPMD for Grids (heterogeneity, distribution, *etc.*)
  - Solution 1: Master keeps the bound
    - previous work shows that not scale [Aida 2003]
  - Solution 2: Message framework (Enterprise Service Bus)
    - Grid middleware dependent / Good for SOA
  - Solution 3: Broadcasting
    - 1 to n communication cannot scale
    - ✓ hierarchical broadcasting scale [Baduel 05]

Organizing Communications for Broadcasting

Organizing Communications for Broadcasting

Idea: Grids are composed of clusters ➟ organizing Workers in groups

Organizing Communications for Broadcasting

Idea: Grids are composed of clusters ➟ organizing Workers in groups

- clusters are high-performance communication environments

Organizing Communications for Broadcasting

Idea: Grids are composed of clusters ➟ organizing Workers in groups

- clusters are high-performance communication environments

Organizing Communications for Broadcasting

Idea: Grids are composed of clusters ➟ organizing Workers in groups

- clusters are high-performance communication environments

- Solution:
  - add a new entity for organizing communications: Leader
  - Leader is a Worker chose by the Master for each group

Organizing Communications for Broadcasting

Idea: Grids are composed of clusters ➟ organizing Workers in groups

- clusters are high-performance communication environments
- Solution:
    - add a new entity for organizing communications: Leader
    - Leader is a Worker chose by the Master for each group

Organizing Communications for Broadcasting

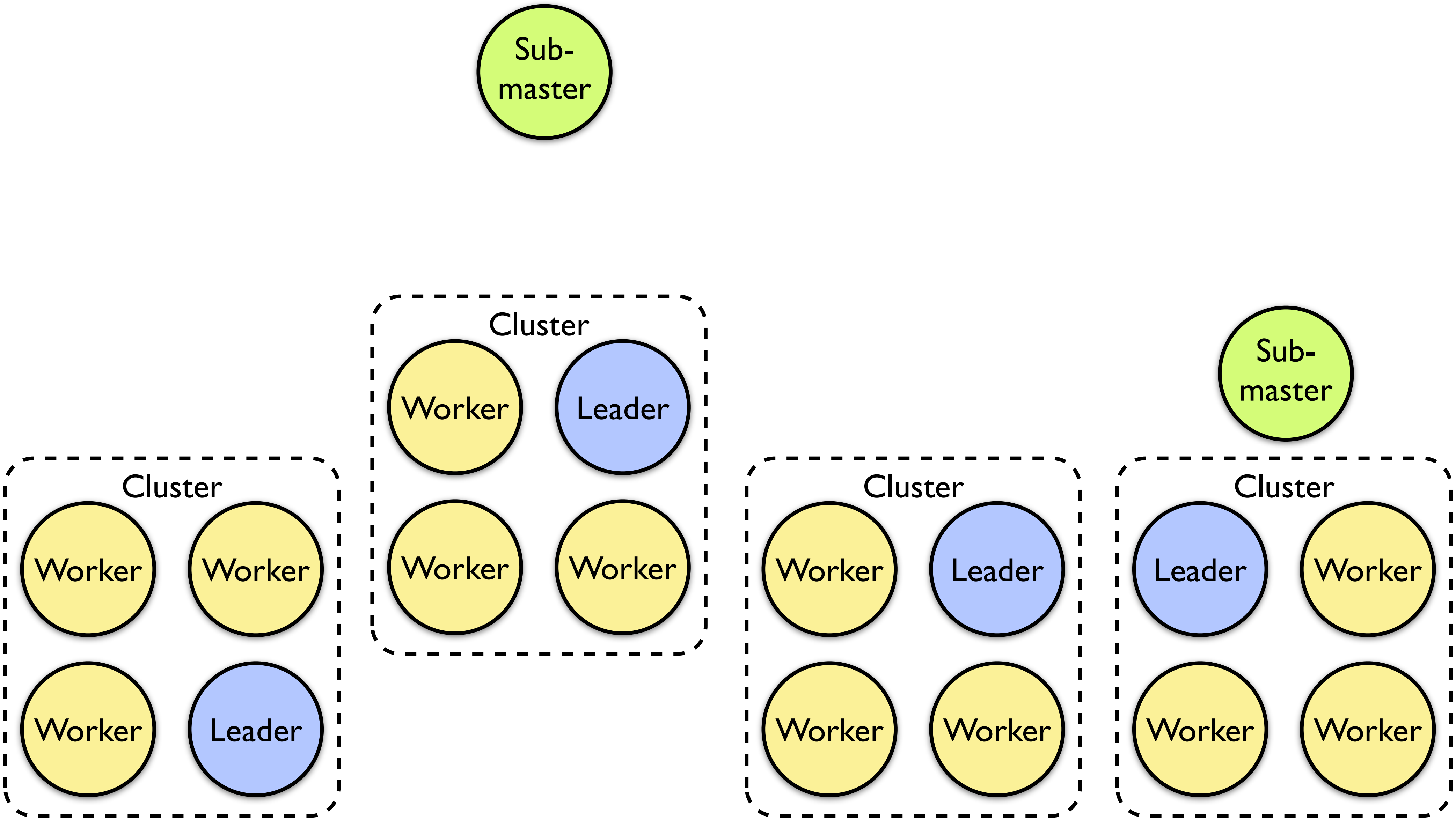
Idea: Grids are composed of clusters ➟ organizing Workers in groups

- clusters are high-performance communication environments

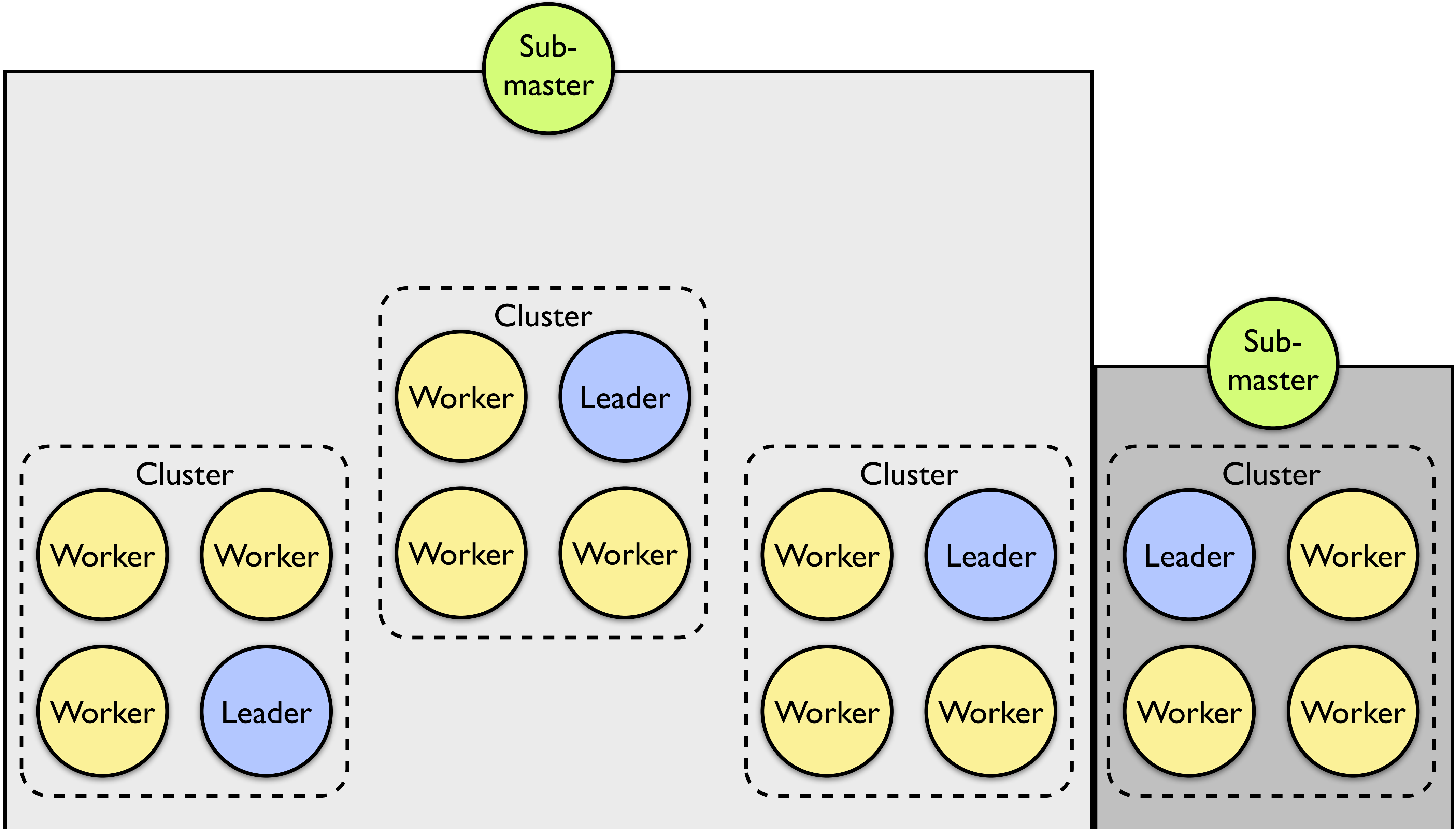
- Solution:
  - add a new entity for organizing communications: Leader
  - Leader is a Worker chose by the Master for each group

- Process to update Bounds:
  1. the Worker broadcasts the new Bound inside its group
  2. the group Leader broadcasts the new Bound to all Leaders
  3. each Leader broadcasts the new value inside their groups

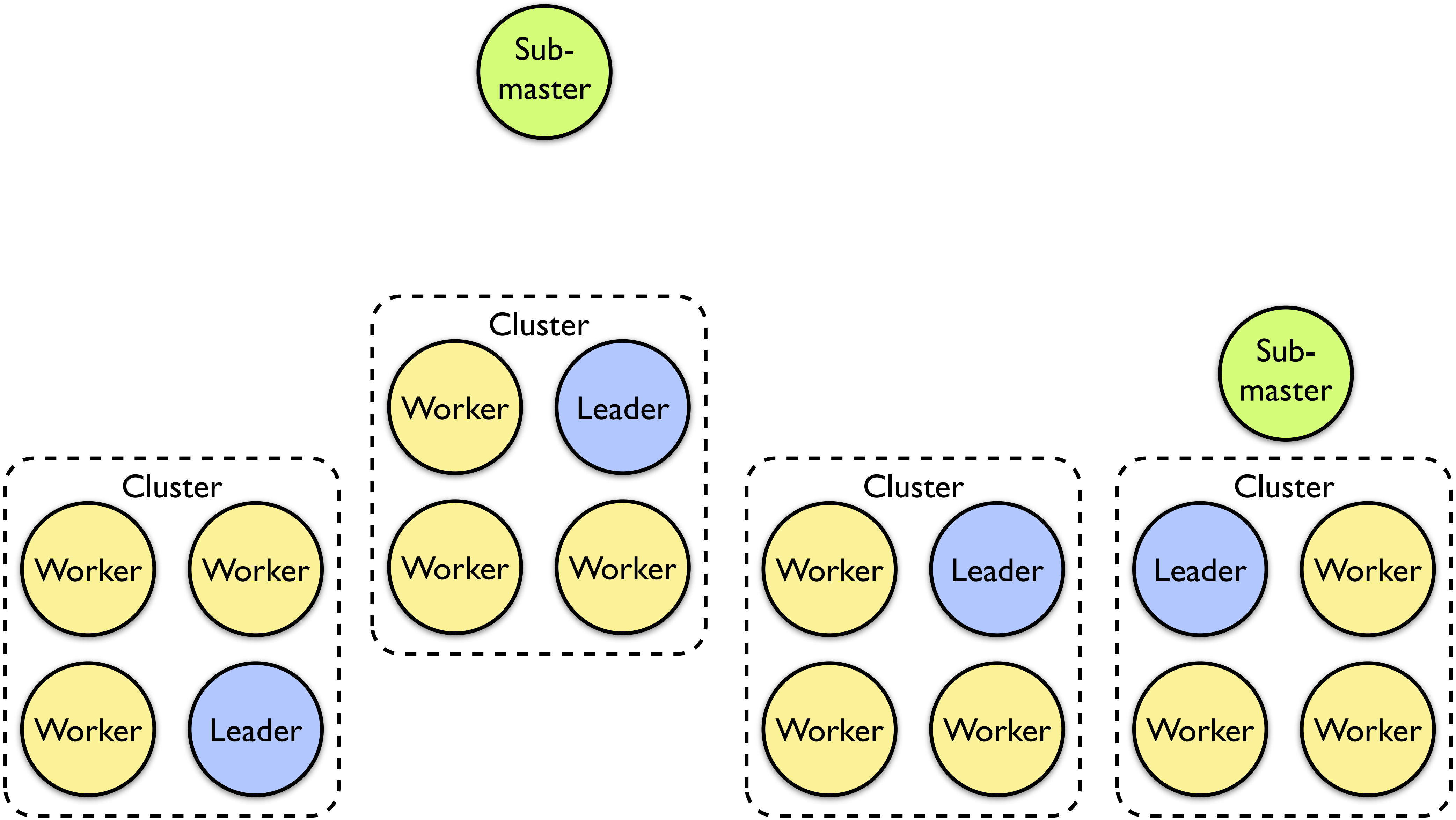
BnB framework - Communications for Sharing Bound



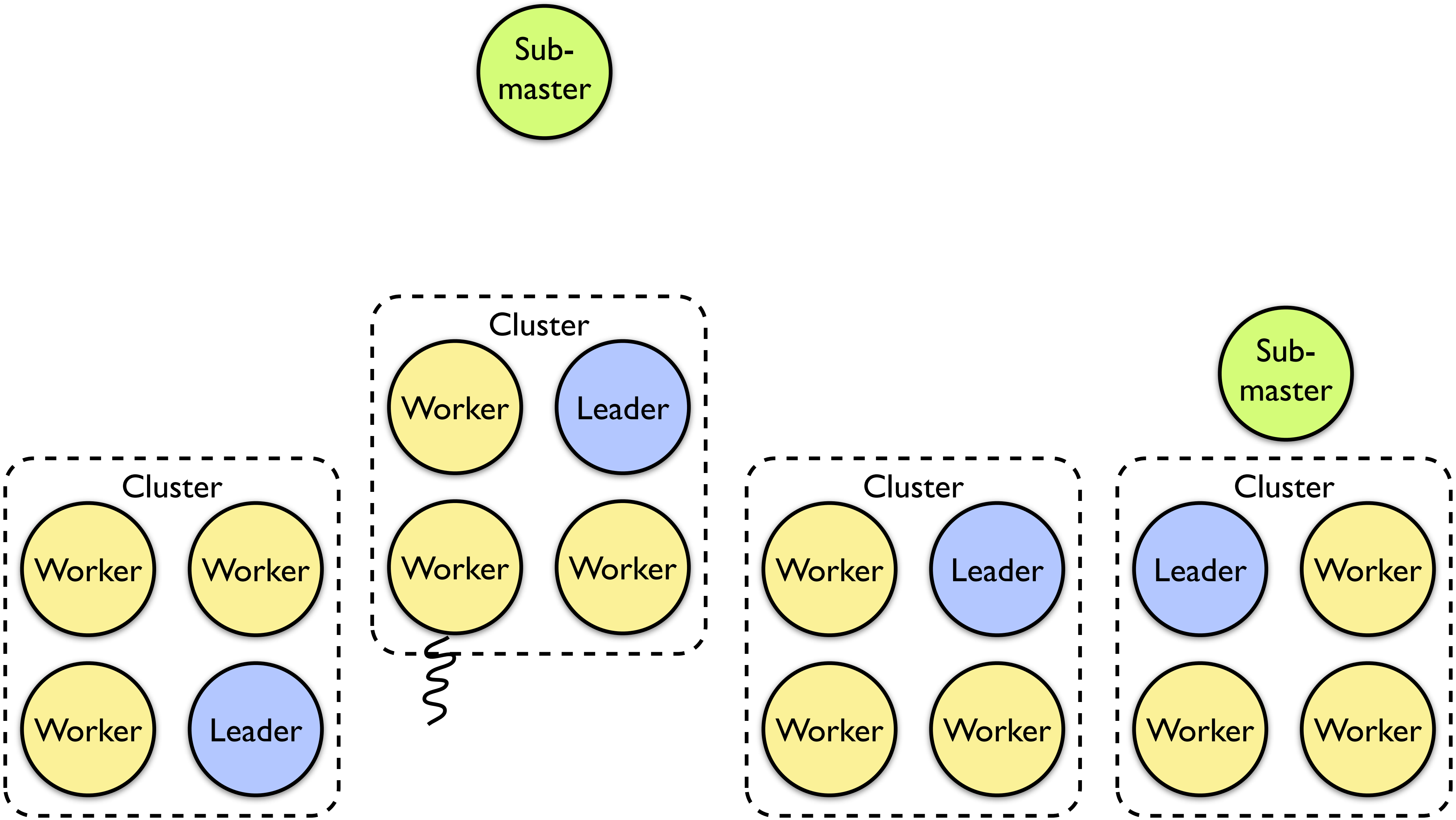
BnB framework - Communications for Sharing Bound



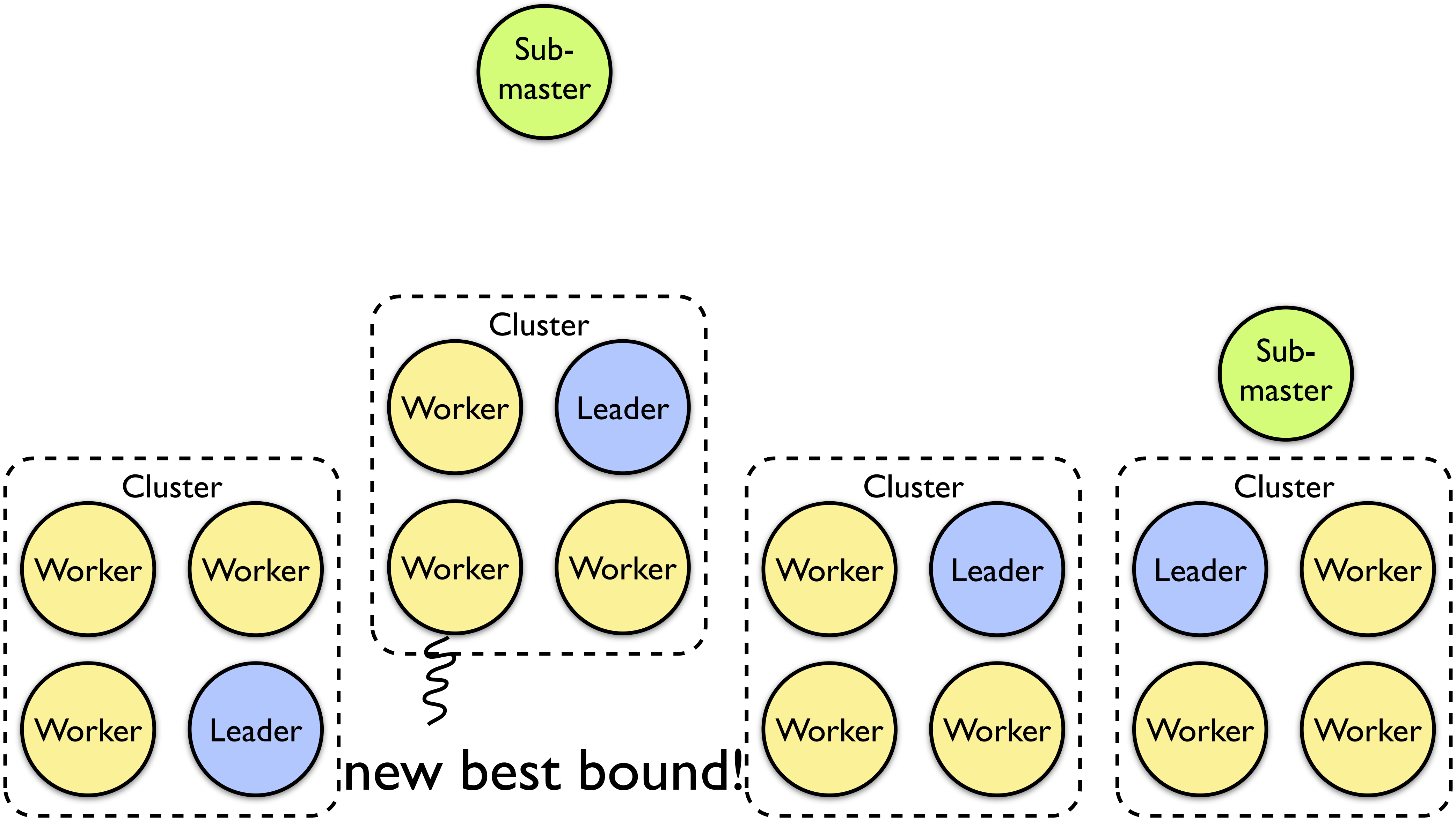
BnB framework - Communications for Sharing Bound



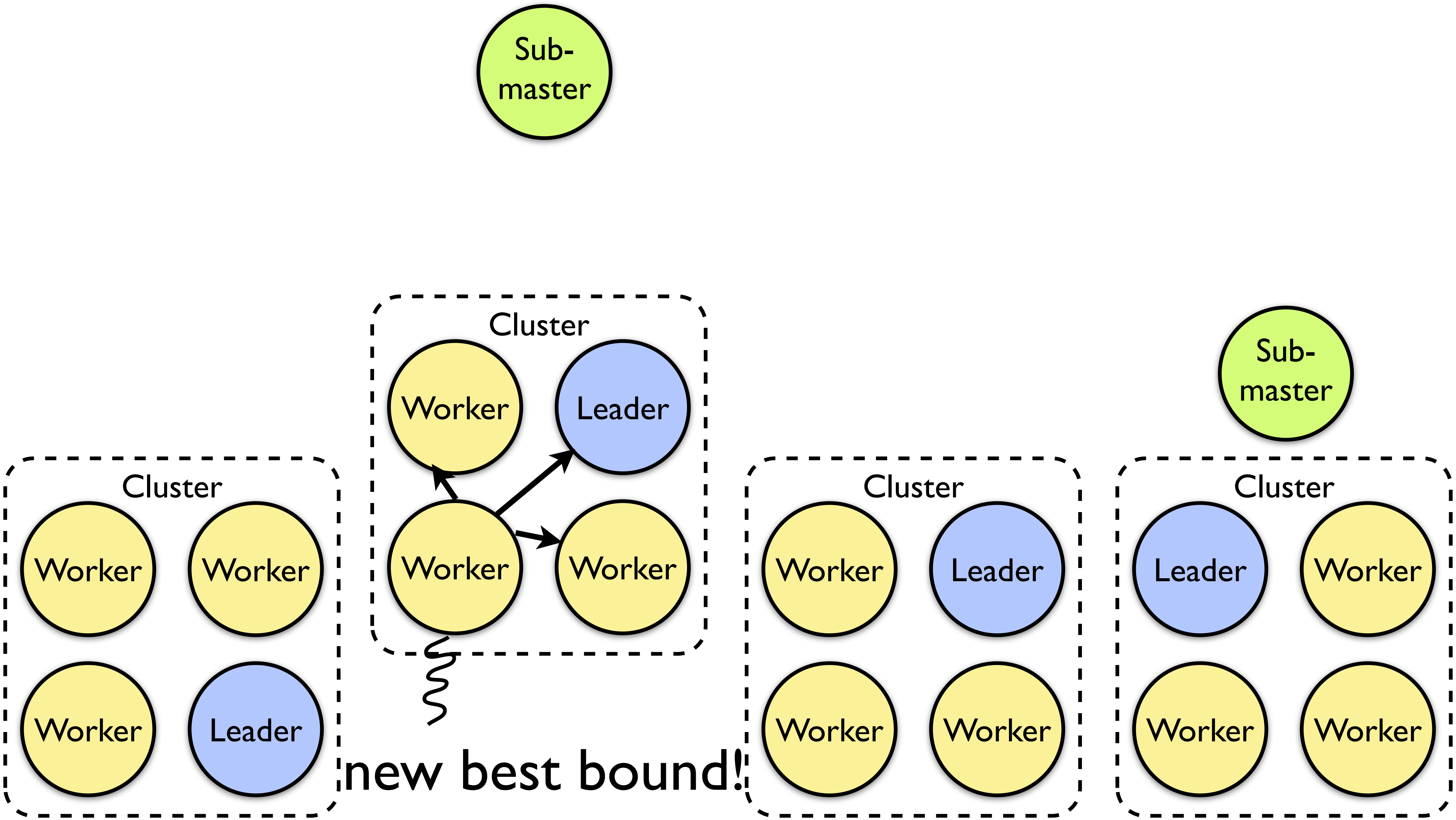
BnB framework - Communications for Sharing Bound



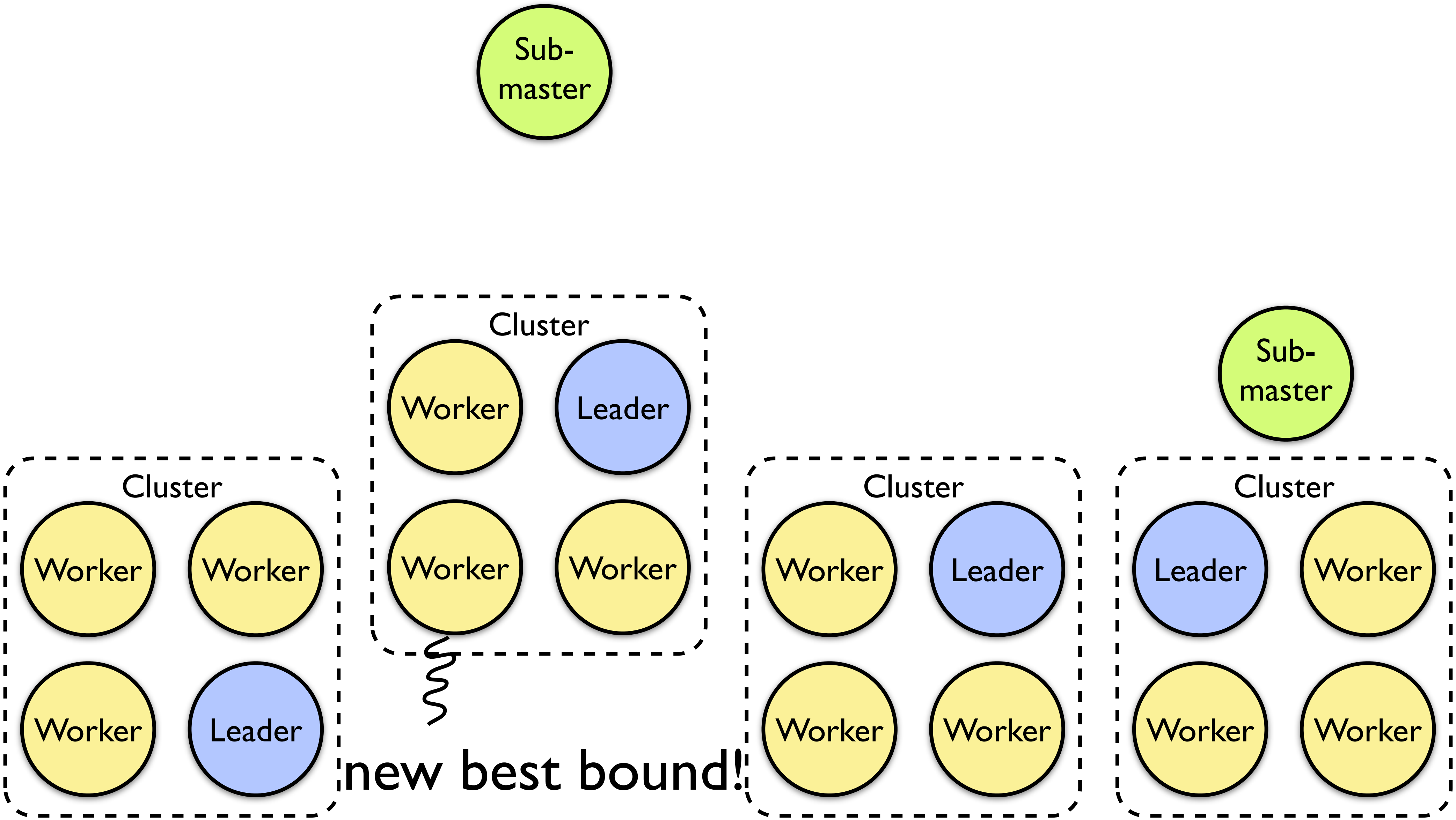
BnB framework - Communications for Sharing Bound



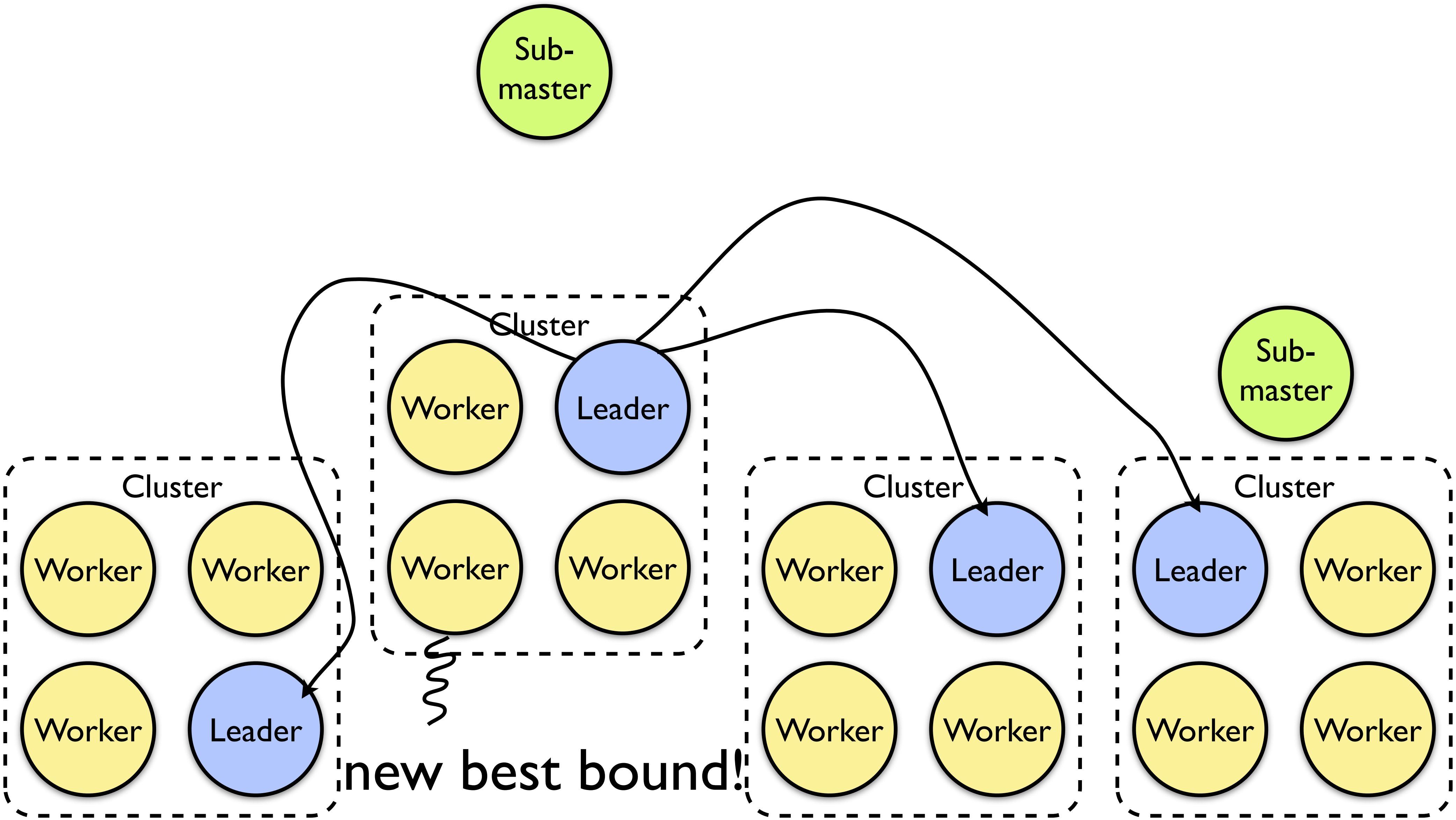
BnB framework - Communications for Sharing Bound



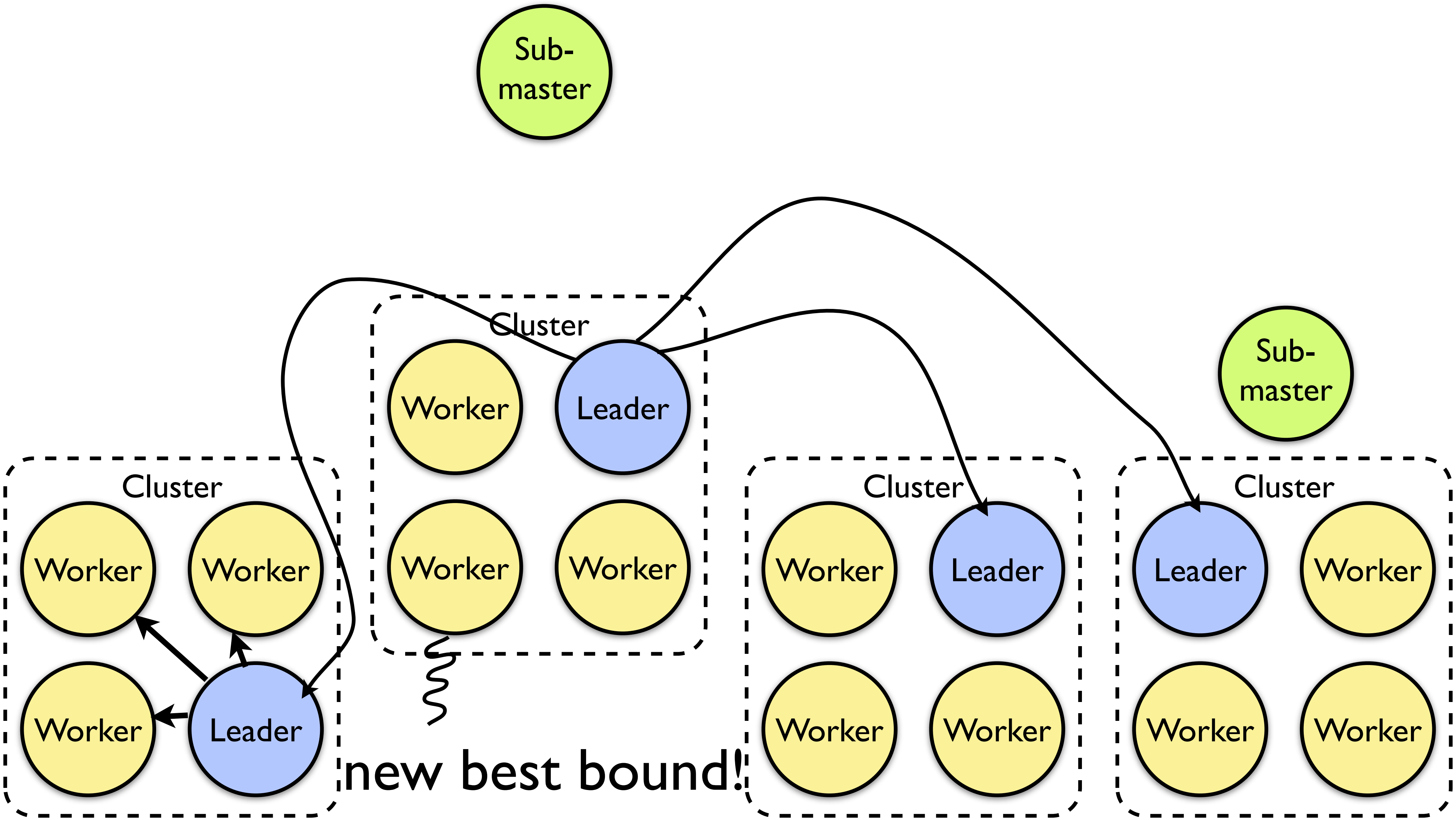
BnB framework - Communications for Sharing Bound



BnB framework - Communications for Sharing Bound

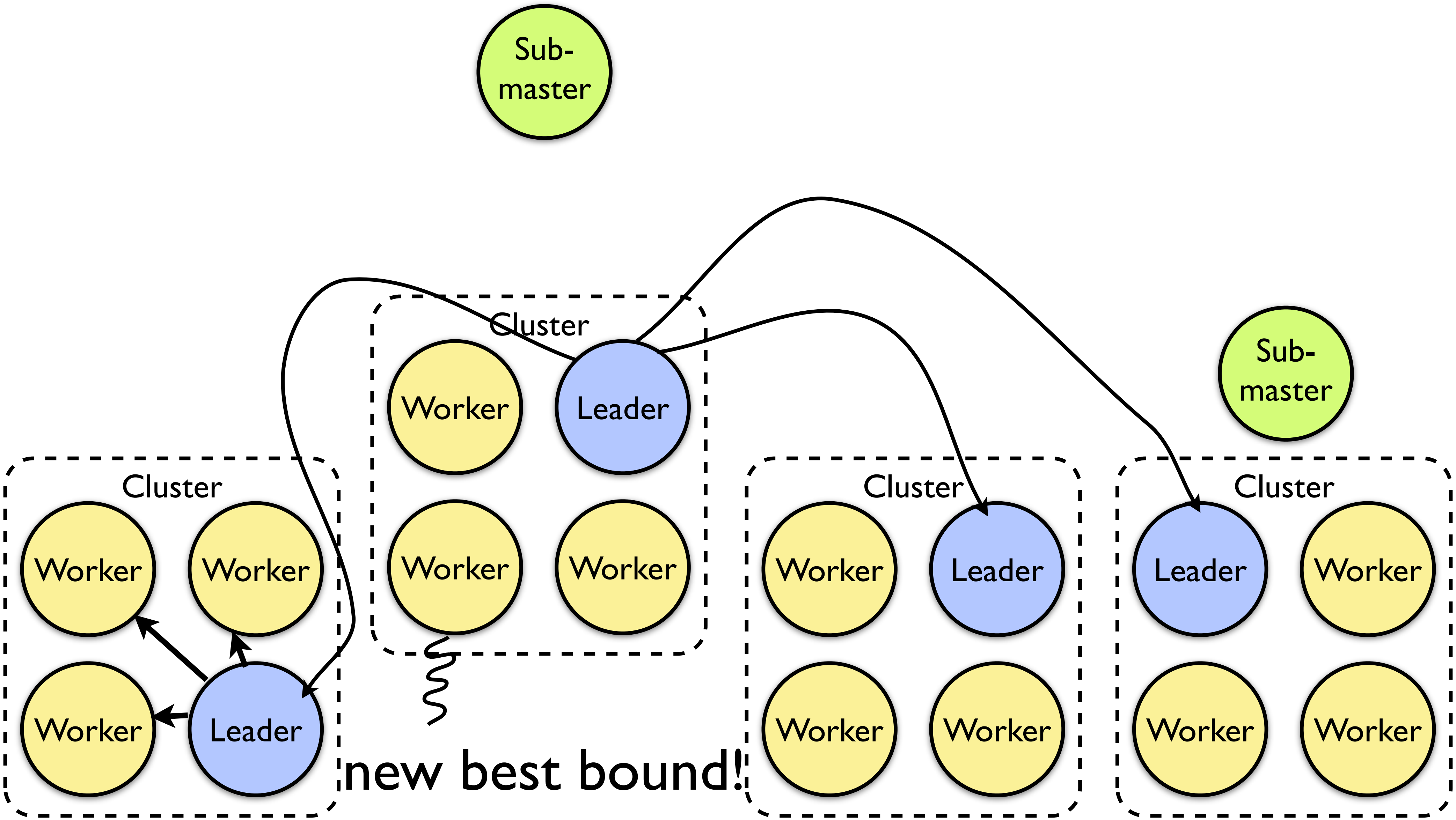


BnB framework - Communications for Sharing Bound



BnB framework - Communications for Sharing Bound

Grid middleware provides communications



BnB Search Strategies

- The improvement of the Bound
  1. depends of how it is shared (communication)
  2. depends of how the search-tree is generated:
     - Classical
       - Depth-First Search
       - Breadth-First Search
     - Contribution
       - First-In-First-Out (FIFO)
       - Priority
       - open API ...

BnB Framework: Fault-Tolerance

BnB Framework: Fault-Tolerance

- Manage user exception ➟ computation stopped

BnB Framework: Fault-Tolerance

- Manage user exception ➟ computation stopped

For us, a fault is a Failed Stop

BnB Framework: Fault-Tolerance

- Manage user exception ➟ computation stopped

For us, a fault is a Failed Stop

- Worker fault ➟ handled by (sub-)master

BnB Framework: Fault-Tolerance

- Manage user exception ➟ computation stopped

For us, a fault is a Failed Stop

- Worker fault ➟ handled by (sub-)master
- Leader fault ➟ master choose new one

BnB Framework: Fault-Tolerance

- Manage user exception ➟ computation stopped

For us, a fault is a Failed Stop

- Worker fault ➟ handled by (sub-)master
- Leader fault ➟ master choose new one
- Sub-master fault ➟ manager change a worker to sub-master

BnB Framework: Fault-Tolerance

- Manage user exception ➟ computation stopped

For us, a fault is a Failed Stop

- Worker fault ➟ handled by (sub-)master
- Leader fault ➟ master choose new one
- Sub-master fault ➟ manager change a worker to sub-master
- Master fault ➟ back-up file

BnB Framework: Fault-Tolerance

- Manage user exception ➟ computation stopped

For us, a fault is a Failed Stop

- Worker fault ➟ handled by (sub-)master
- Leader fault ➟ master choose new one
- Sub-master fault ➟ manager change a worker to sub-master
- Master fault ➟ back-up file

- Load-Balancing is natural with master-worker

BnB Framework: Fault-Tolerance

- Manage user exception ➟ computation stopped

For us, a fault is a Failed Stop

- Worker fault ➟ handled by (sub-)master
- Leader fault ➟ master choose new one
- Sub-master fault ➟ manager change a worker to sub-master
- Master fault ➟ back-up file

- Load-Balancing is natural with master-worker
  - the framework provides a function to get the number of free Workers ➟ users use it to decide branching

Grid'BnB [HiPC07] Features

Grid'BnB [HiPC07] Features

Design

Grid'BnB [HiPC07] Features

- Asynchronous communications
- Hierarchical master-worker with com.
- Dynamic task splitting
- Efficient communications with groups
- Fault-tolerance

Design

Grid'BnB [HiPC07] Features

- Asynchronous communications
- Hierarchical master-worker with com.
- Dynamic task splitting
- Efficient communications with groups
- Fault-tolerance

Design

Users

Grid'BnB [HiPC07] Features

- Asynchronous communications
- Hierarchical master-worker with com.
- Dynamic task splitting
- Efficient communications with groups
- Fault-tolerance

Design

- Hidden parallelism and Grid difficulties
- API for COPs
- Ease of deployment
- Principally tree-based
- Implementing and testing search strategies
- Focus on objective function

Users

Grid'BnB [HiPC07] Features

- Asynchronous communications
- Hierarchical master-worker with com.
- Dynamic task splitting
- Efficient communications with groups

Design

Validate and Test Grid'BnB by experiments

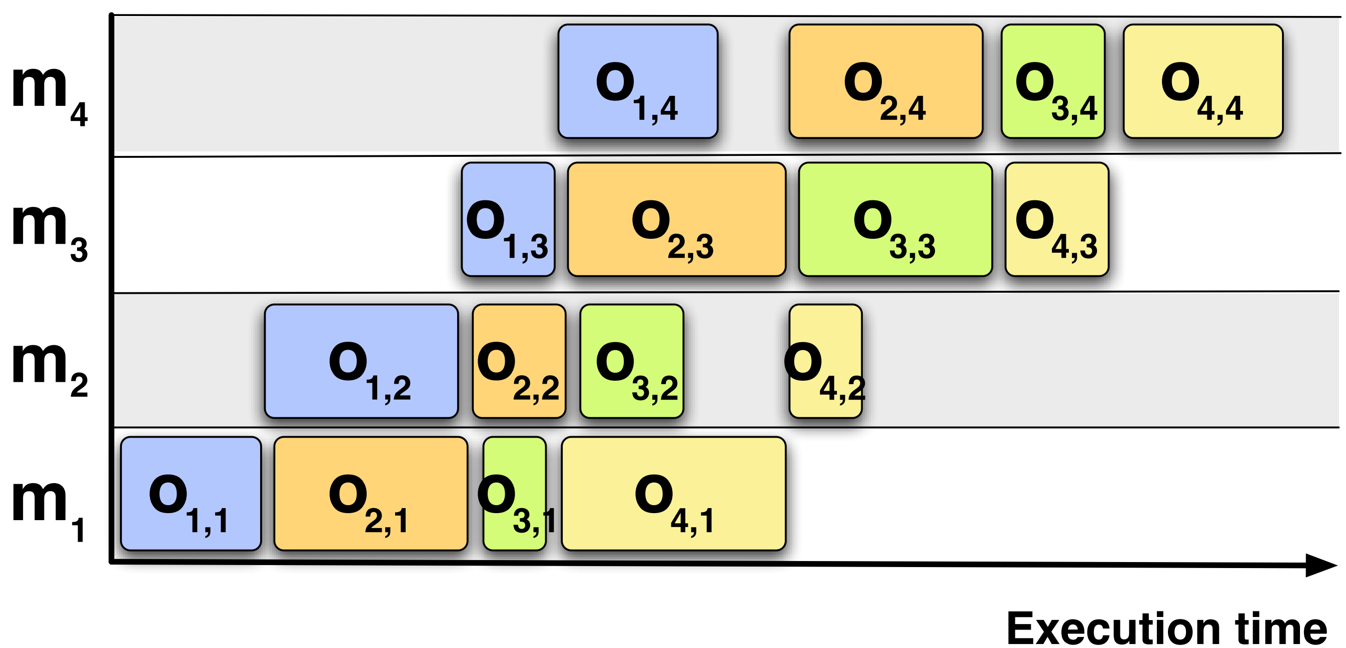
- Hidden parallelism and Grid difficulties
- API for COPs
- Ease of deployment
- Principally tree-based
- Implementing and testing search strategies
- Focus on objective function

Users

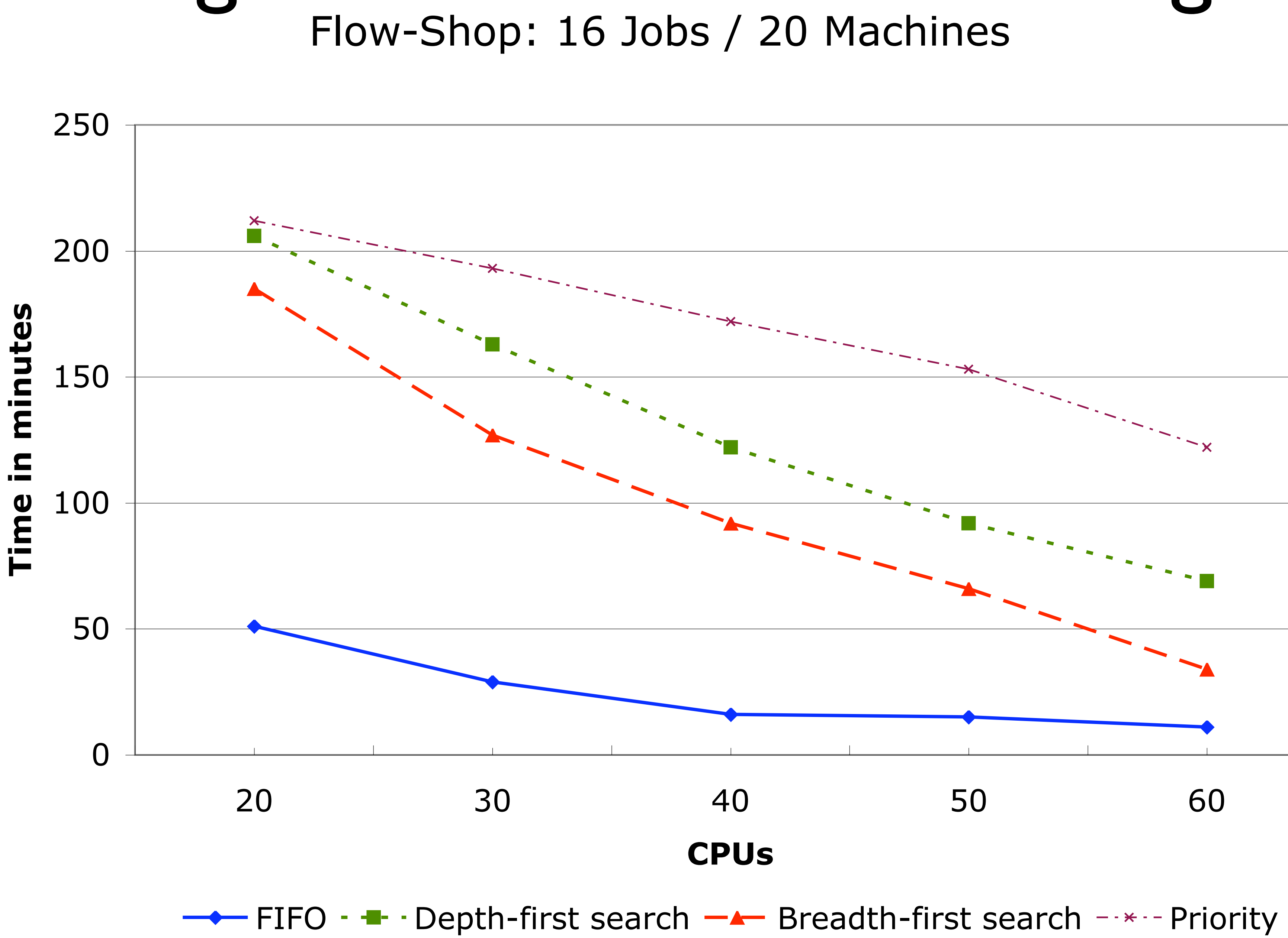
Flow-Shop Experiments

- NP-hard permutation optimization problem

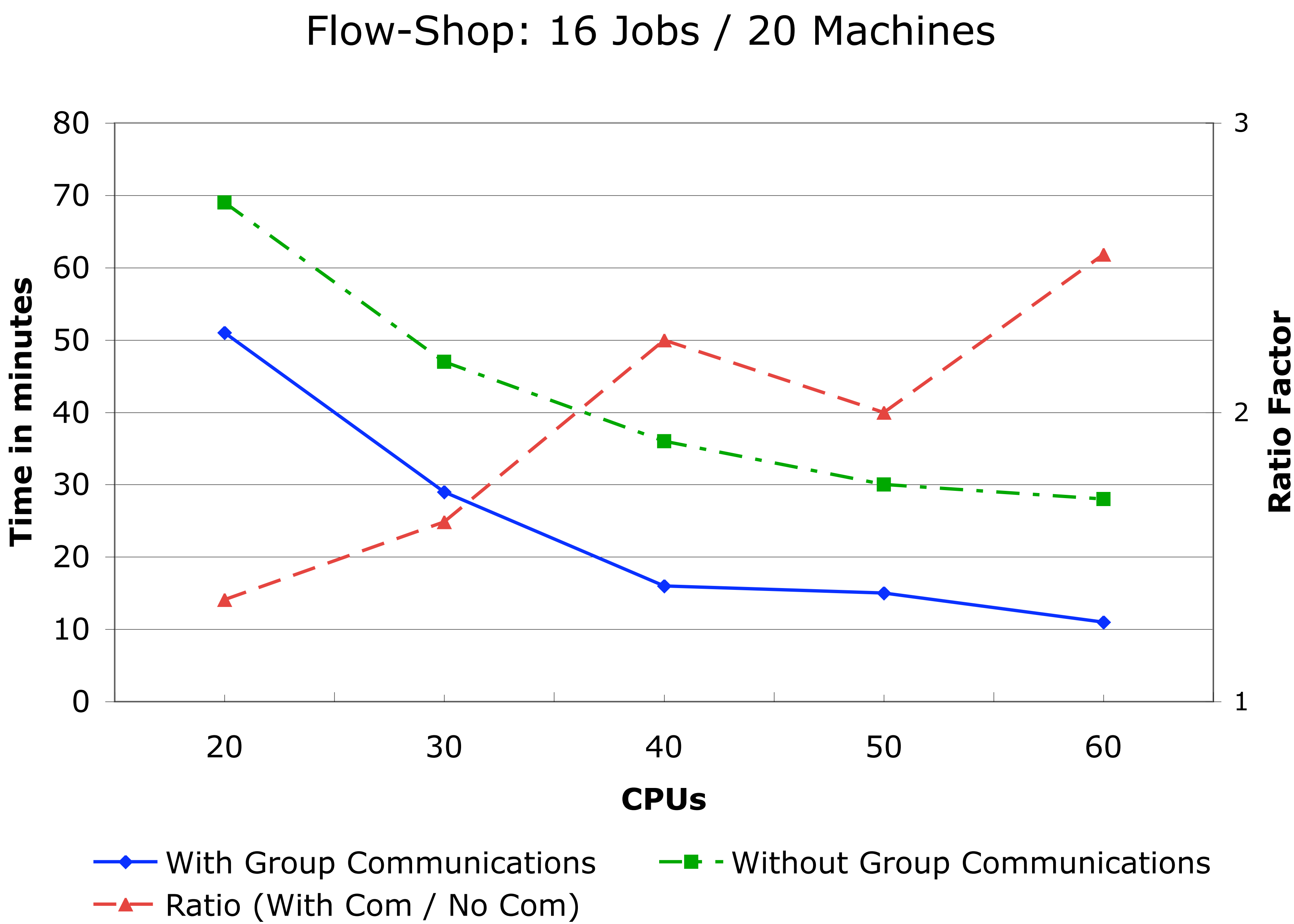
$$J = \{j1, j2, . . . jn\}$$
$$ji = \{ oi1, oi2, . . . oim \}$$
$$M = \{m1, m2, . . . mm\}$$



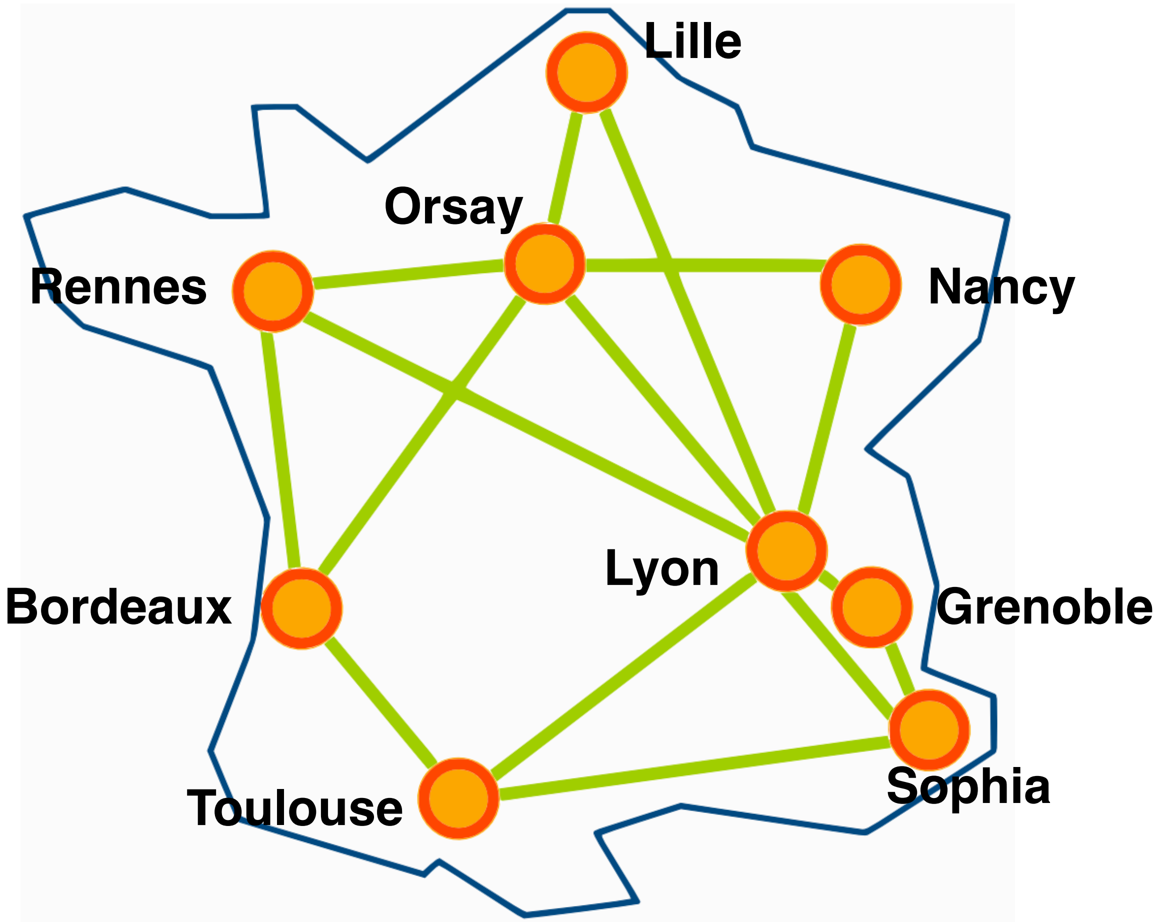
Single Cluster: Search Strategies



Single Cluster: Communications



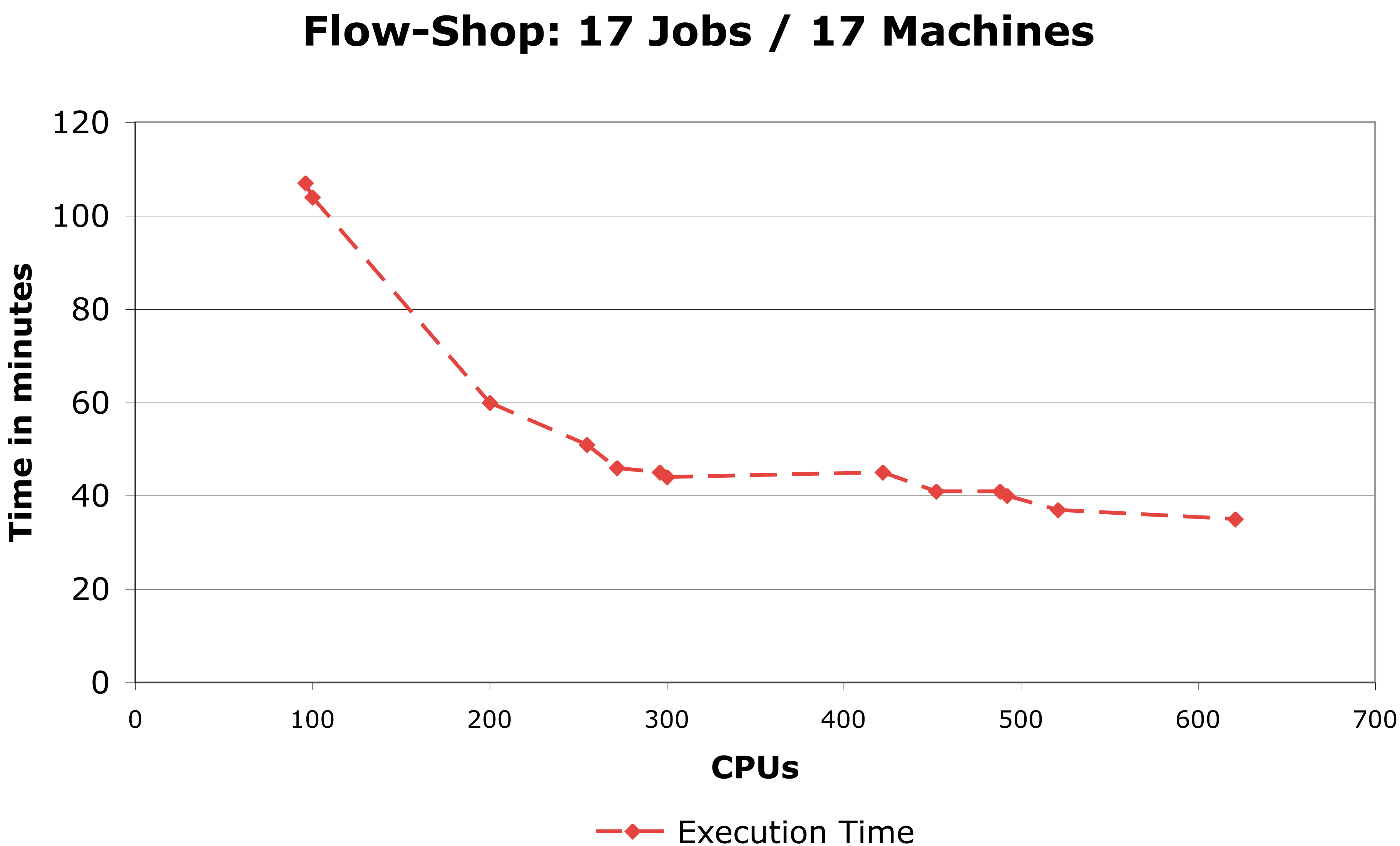
Grid'5000: the French Grid



9 sites - 14 clusters - 3586 CPUs

Grid Experimentations

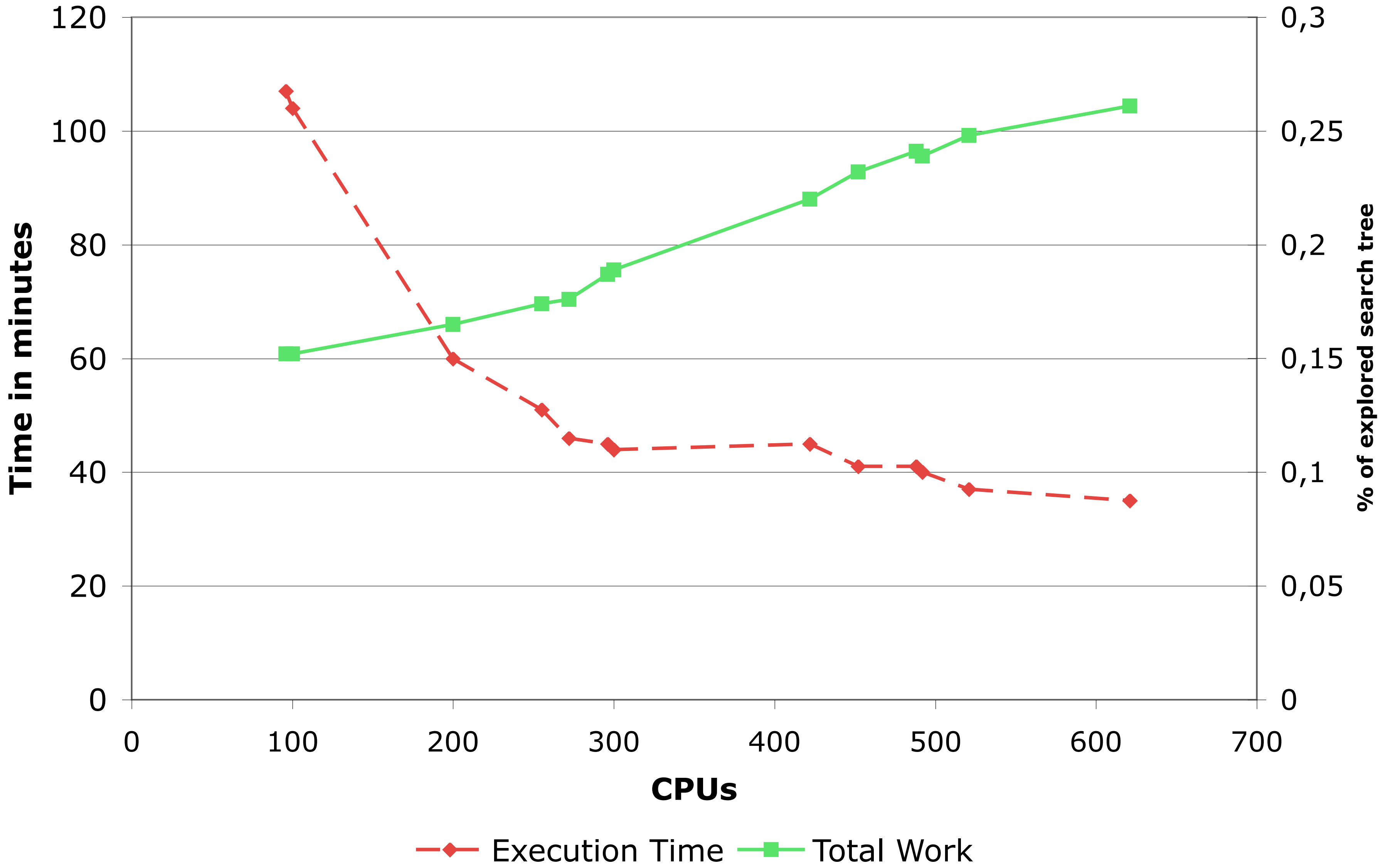
up to 621 CPUs on 5 sites



Grid Experimentations

up to 621 CPUs on 5 sites

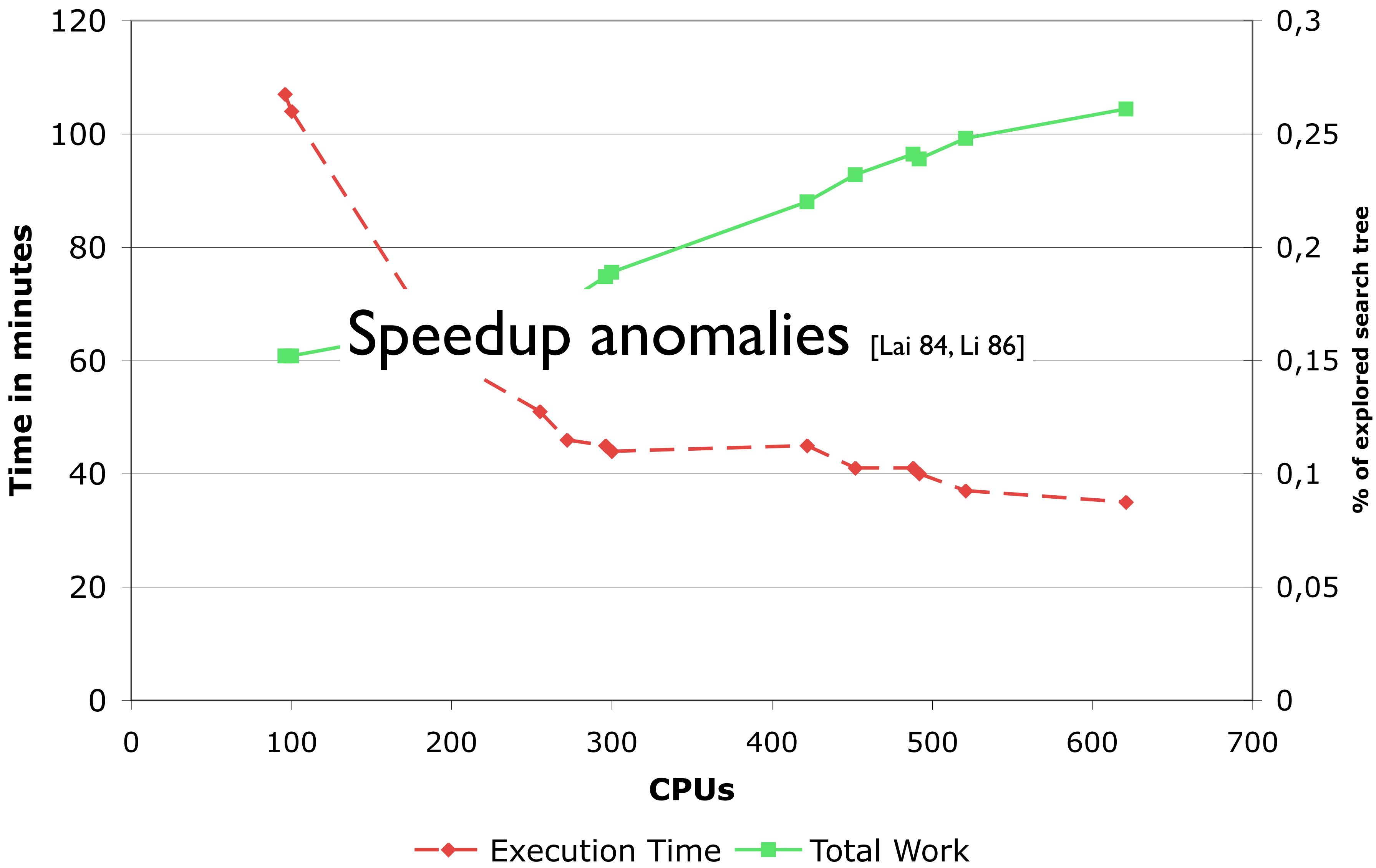
Flow-Shop: 17 Jobs / 17 Machines



Grid Experimentations

up to 621 CPUs on 5 sites

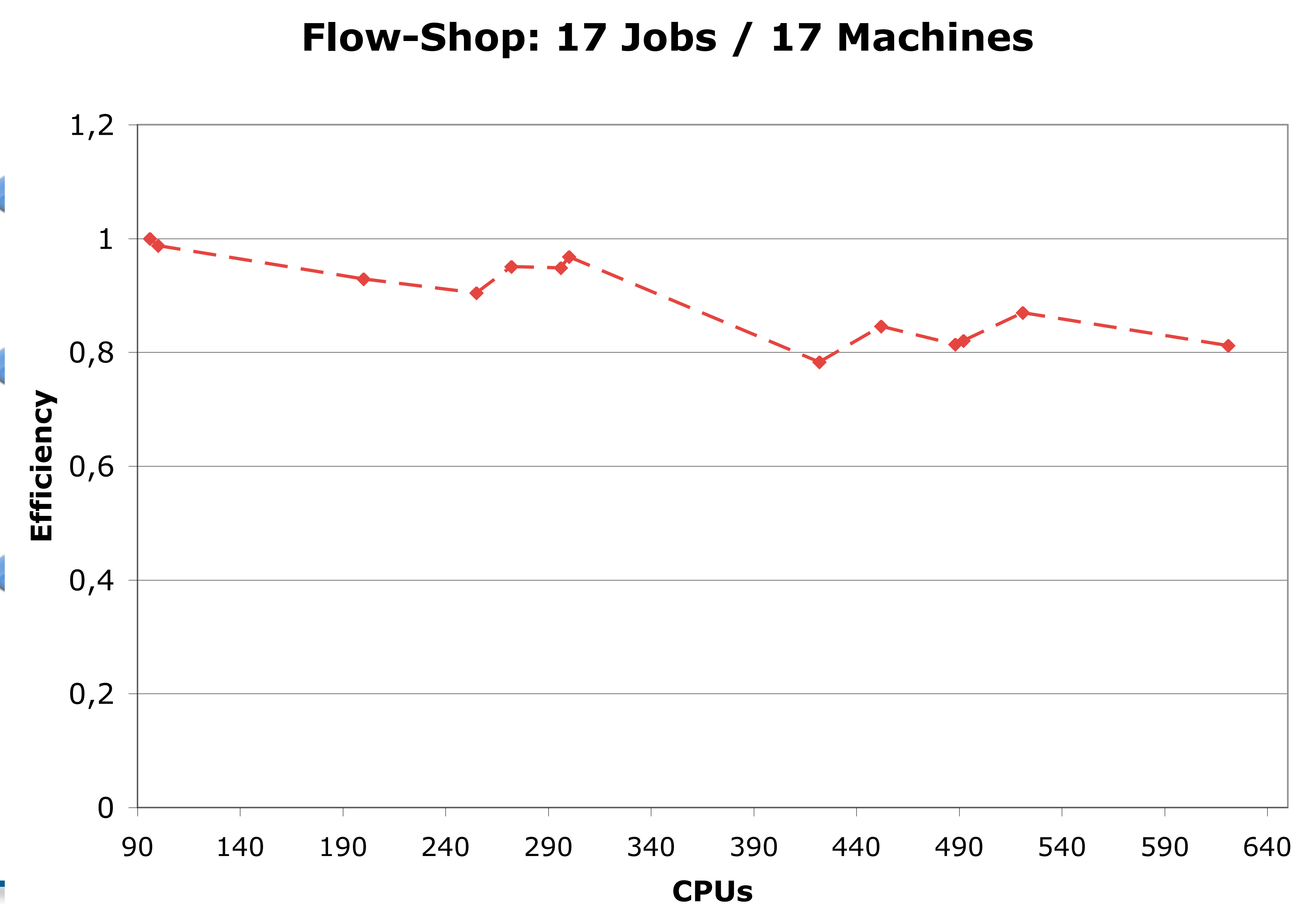
Flow-Shop: 17 Jobs / 17 Machines



Speedup Anomalies & Efficiency

- Parallel tree-based speedup may sometimes quite spectacular (> or < linear) [Mans 95]
- Speedup Anomalies in BnB [Roucairol 87, Lai 84, Li 86]
  - speedup depends of how the tree is dynamically built
- Efficiency: is a related measure computed as the speedup divided by the number of processors.

Speedup Anomalies & Efficiency



Grid'BnB: Results

Grid'BnB: Results

- Experimentally validate our BnB framework for Grids
  - validity of organizing communications
  - scalability on Grid (up to 621 CPUs on 5 sites)

Grid'BnB: Results

- Experimentally validate our BnB framework for Grids
  - validity of organizing communications
  - scalability on Grid (up to 621 CPUs on 5 sites)

Grid'BnB: Results

- Experimentally validate our BnB framework for Grids
  - validity of organizing communications
  - scalability on Grid (up to 621 CPUs on 5 sites)

- Problems:
  - deployment on Grids is difficult
  - dynamically acquiring new resources is difficult
  - popularity of Grid'5000
  - cannot mix Grid'5000 and under-utilized lab desktops

Grid'BnB: Results

- Experimentally validate our BnB framework for Grids
  - validity of organizing communications
  - scalability on Grid (up to 621 CPUs on 5 sites)

Grid middleware needs a better supporting of dynamic infrastructure

  - deployment on Grids is difficult
  - dynamically acquiring new resources is difficult
  - popularity of Grid'5000
  - cannot mix Grid'5000 and under-utilized lab desktops

Agenda

- Context, Problem, and Related Work
- Contributions
  - Branch-and-Bound Framework for Grids
  - Desktop Grid with Peer-to-Peer
  - Mixing Desktops & Clusters
- Perspectives & Conclusion

Peer-to-Peer as Grid Middleware

- Grid Computing and Peer-to-Peer share a common goal:
  - sharing resources [Foster 03, Goux 00]
- Grid related work: [Globus]
  - ✓ providing computational resources
  - \- installing/deploying Grid middleware is difficult
- P2P related work: [Gnutella, Freenet, DHT]
  - \- focusing on sharing data & mono-application
  - ✓ dynamic & easy to deploy

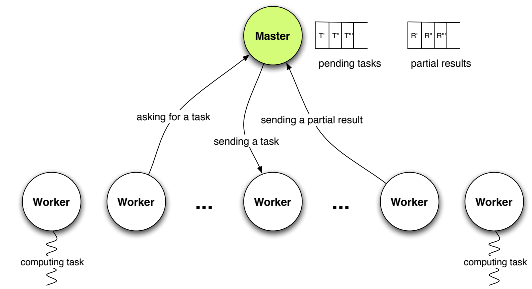
Peer-to-Peer as Grid Middleware

- Grid Computing and Peer-to-Peer share a common goal:
  - sharing resources [Foster 03, Goux 00]
- Grid related work: [Globus]
  - ✓ providing computational resources
  - \- installing/deploying Grid middleware is difficult
- P2P related work: [Gnutella, Freenet, DHT]
  - \- focusing on sharing data & mono-application
  - ✓ dynamic & easy to deploy

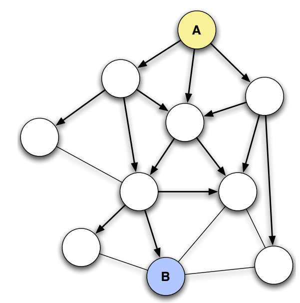
Objective: provide a P2P infrastructure for Grids and sharing computational resources

Which P2P Architecture?

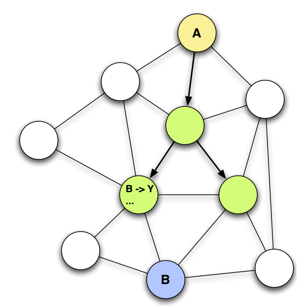
- Master-Worker (SETI@home)
  - centralized
  - targets only desktops
  - good for embarrassingly parallel



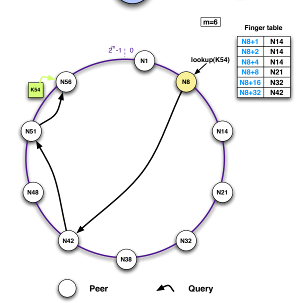
- Pure/Unstructured Peer-to-Peer (Gnutella)
  - flooding problems
  - ✓ supports high-churn (good for Desktop Grids)
  - ✓ supports many kind of application (data, computational)



- Hybrid Peer-to-Peer (JXTA)
  - uses central servers
  - ✓ limits the flooding
  - has to manage churn

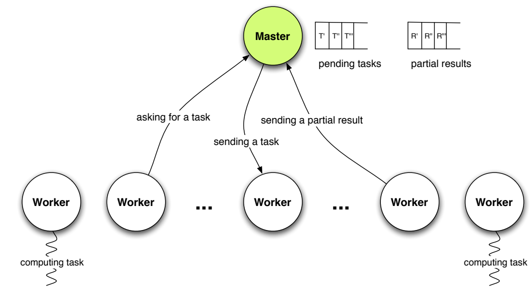


- Structured Peer-to-Peer (Chord)
  - high cost for managing churn
  - efficient for data sharing (Distributed Hash Table)

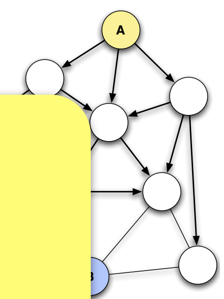


Which P2P Architecture?

- Master-Worker (SETI@home)
  - • centralized
  - - targets only desktops
  - • good for embarrassingly parallel

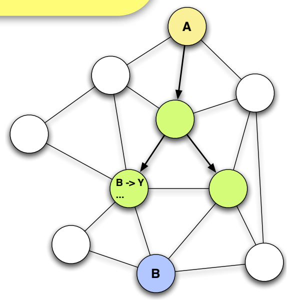


- Pure/Unstructured Peer-to-Peer (Gnutella)

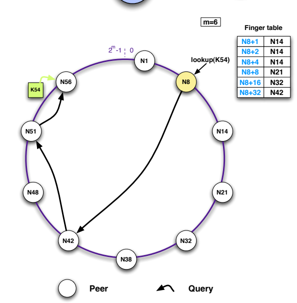


Pure Peer-to-Peer is the most adapted
Needs to avoid the flooding problem

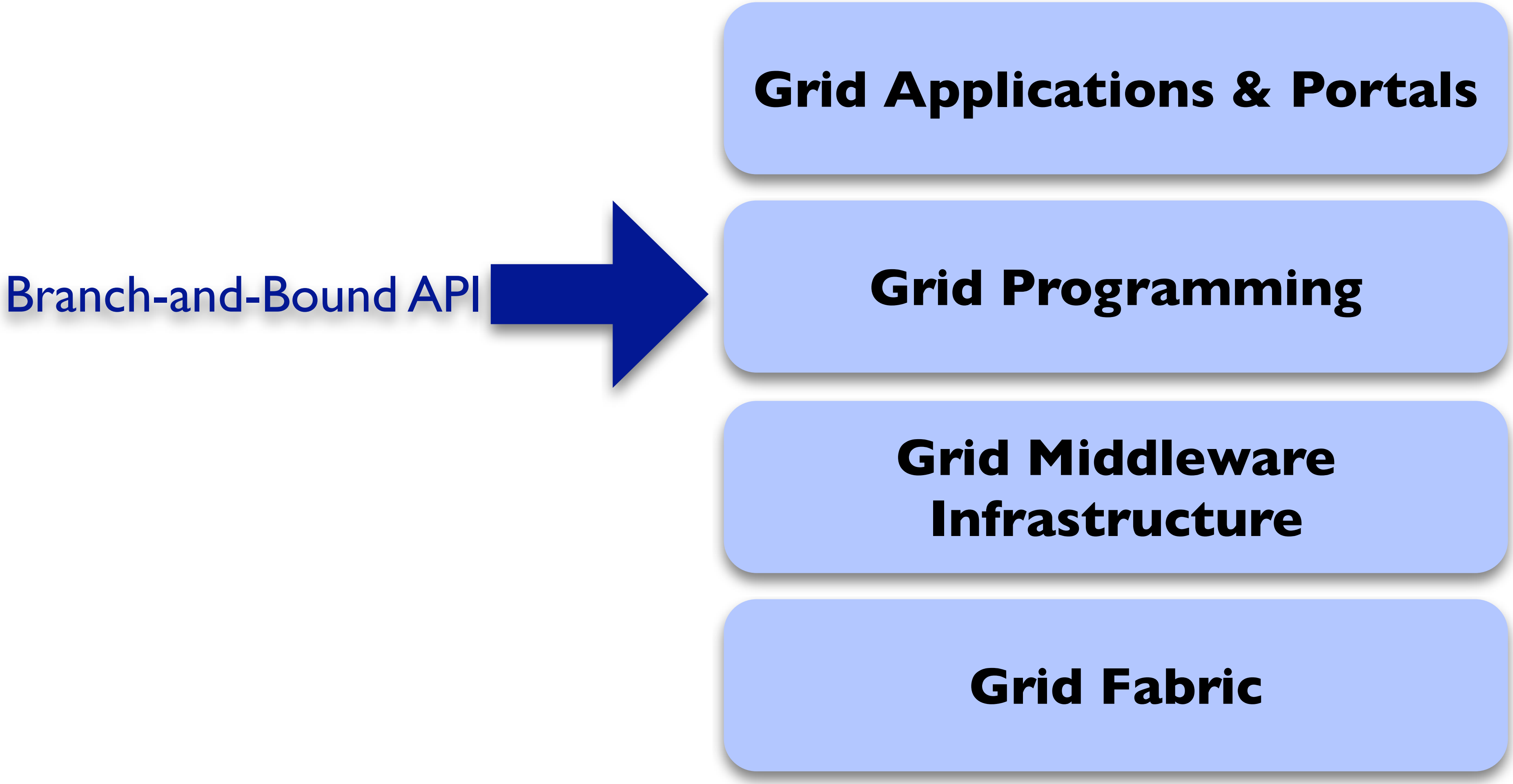
- Hybrid Peer-to-Peer (JXTA)
  - • uses central servers
  - ✓ limits the flooding
  - - has to manage churn



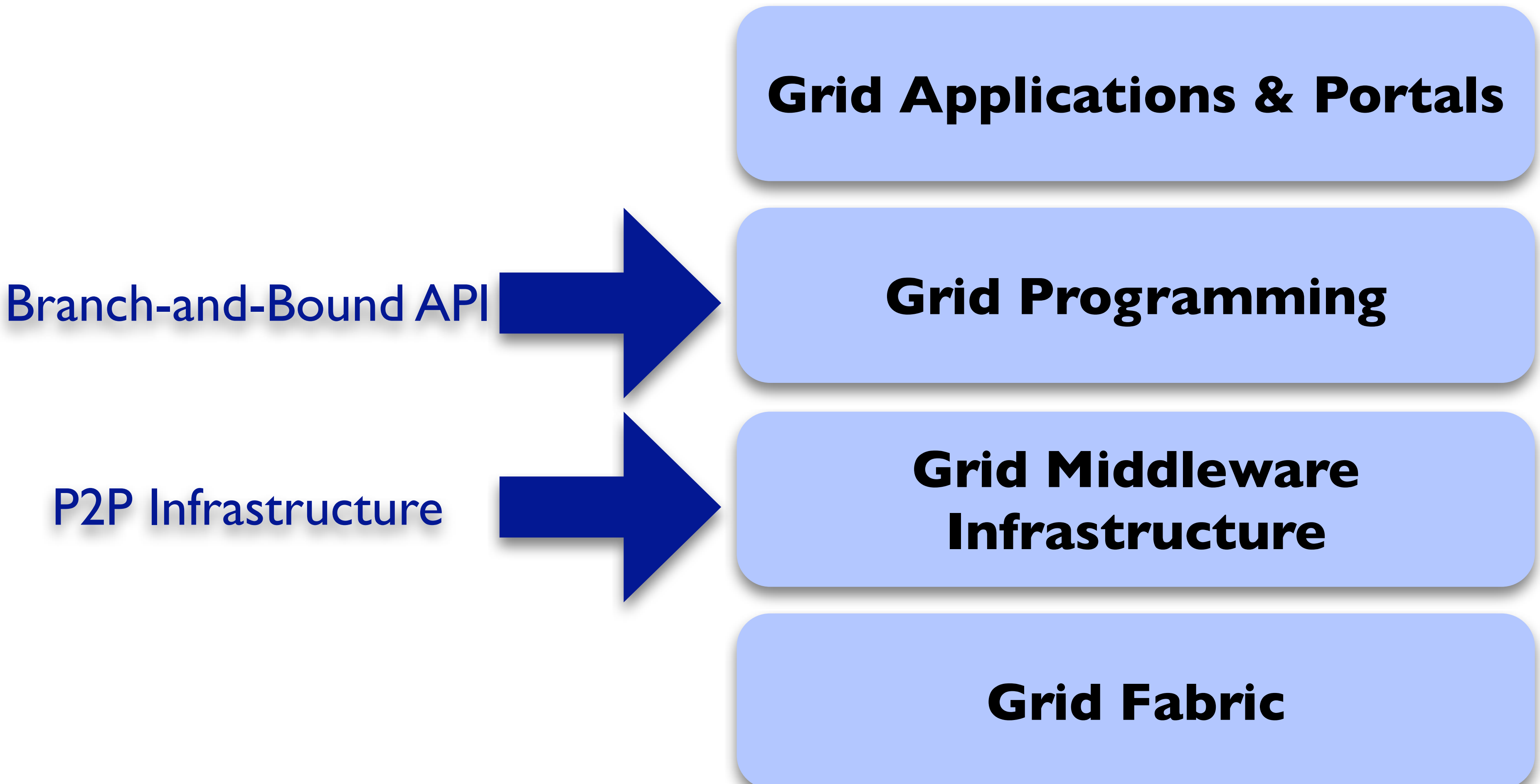
- Structured Peer-to-Peer (Chord)
  - - high cost for managing churn
  - • efficient for data sharing (Distributed Hash Table)



Contribution & Positioning



Contribution & Positioning



The Peer-to-Peer Infrastructure [CMST06]

The Peer-to-Peer Infrastructure [CMST06]

- Pure Peer-to-Peer overlay network
- Using it as Grid middleware infrastructure

- The proposed solution to avoid the flooding problem:
  - 3 node-request protocols:
    - 1 node: Random walk algorithm
    - n nodes: Breadth-First-Search (BFS) algorithm with acknowledgement
    - max nodes: BFS without acknowledgement

- Best-effort

INRIA Sophia P2P Desktop Grid

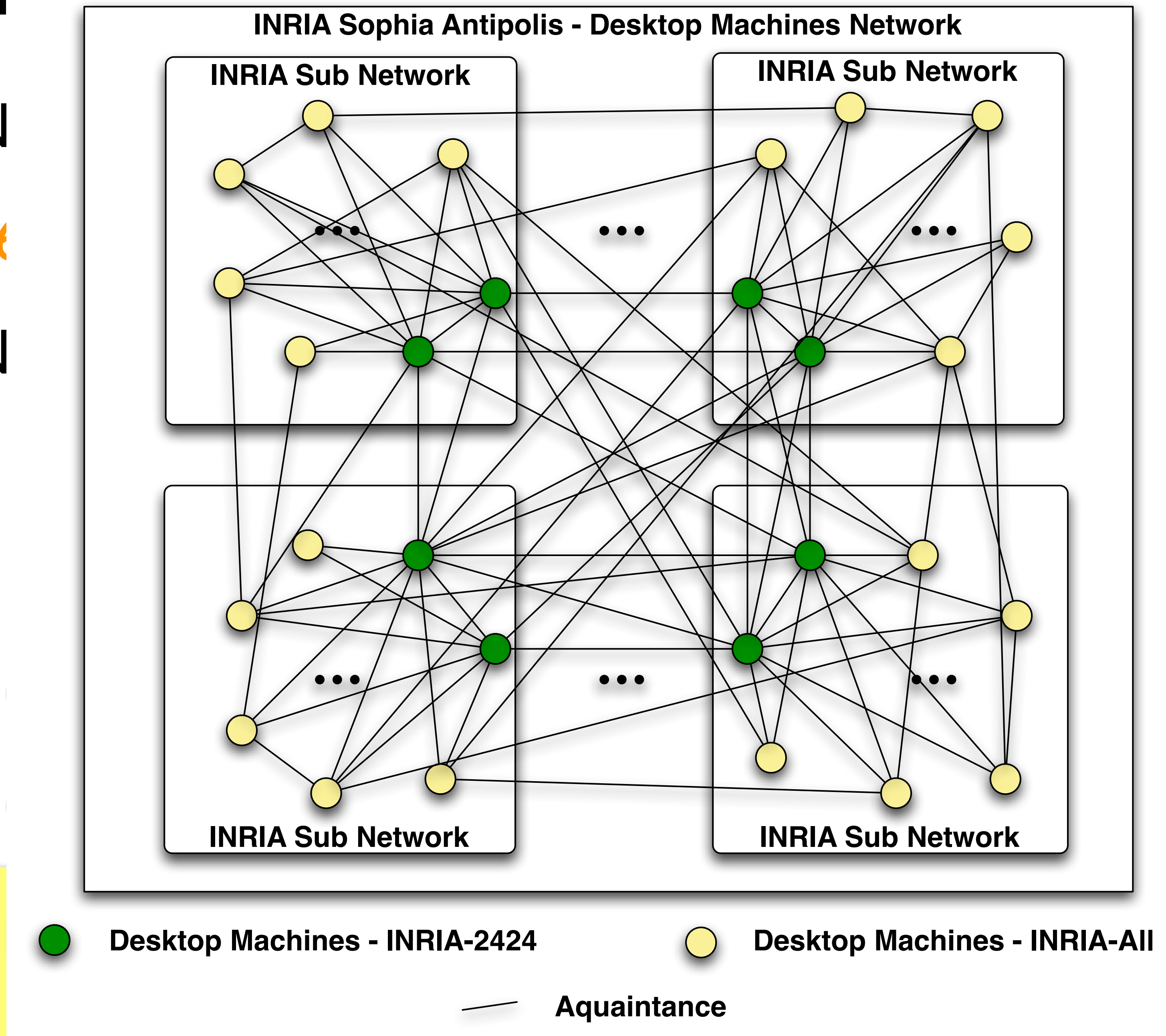
- Need to validate the infrastructure with desktops
- 260 desktops at INRIA Sophia lab
- No disturbing normal users:
  - running in low priority
  - working schedules:
    - 24/24 ≈ 50 machines (INRIA-2424)
    - night/weekend ≈ 260 machines (INRIA-ALL)

INRIA Sophia P2P Desktop Grid

- Need to validate the infrastructure with desktops
- 260 desktops at INRIA Sophia lab
- No disturbing normal users:
  - running in low priority
  - working schedules:
    - 24/24 ≈ 50 machines (INRIA-2424)
    - night/weekend ≈ 260 machines (INRIA-ALL)

Deployed our P2P infrastructure as permanent Desktop Grid at INRIA Sophia

INRIA Sophia P2P Desktop Grid



Long Running Experiments with the P2P Desktop Grid

- Context: ETSI Grid Plugtests contest ➟ n-queens
- n-queens:
  - embarrassingly parallel / CPU intensive / master-worker

Long Running Experiments with the P2P Desktop Grid

- Context: ETSI Grid Plugtests contest ➟ n-queens
- n-queens:
  - embarrassingly parallel / CPU intensive / master-worker

How many solution for 25 queens ?

n-queens Experiment Results

Total # of Tasks	**12,125,199**
Task Computation	**≈ 138"**
Computation Time	**≈ 185 days**
Cumulated Time	**≈ 53 years**
# of Desktop Machines	**260**
Total of Solution Found	2,207,893,435,808,352 **≈ 2 quadrillions**

- World Record [Sloane Integers Sequence A000170]
- What we learn from this experiments:
  - validate the workability of the infrastructure
  - validate the robustness of the infrastructure
  - hard to forecast machine's performances

Agenda

- Context, Problem, and Related Work
- Contributions
  - Branch-and-Bound Framework for Grids
  - Desktop Grid with Peer-to-Peer
  - Mixing Desktops & Clusters
- Perspectives & Conclusion

Mixing Desktops & Clusters [PARCO07]

Grid'BnB
Parallel BnB Framework for Grid

P2P Infrastructure
as Grid Midlleware

Mixing Desktops & Clusters [PARCO07]

Grid'BnB
Parallel BnB Framework for Grid

+

P2P Infrastructure
as Grid Midlleware

API for solving COPs &
Dynamic Grid Infrastructure

Mixing Desktops & Clusters [PARCO07]

Grid'BnB
Parallel BnB Framework for Grid

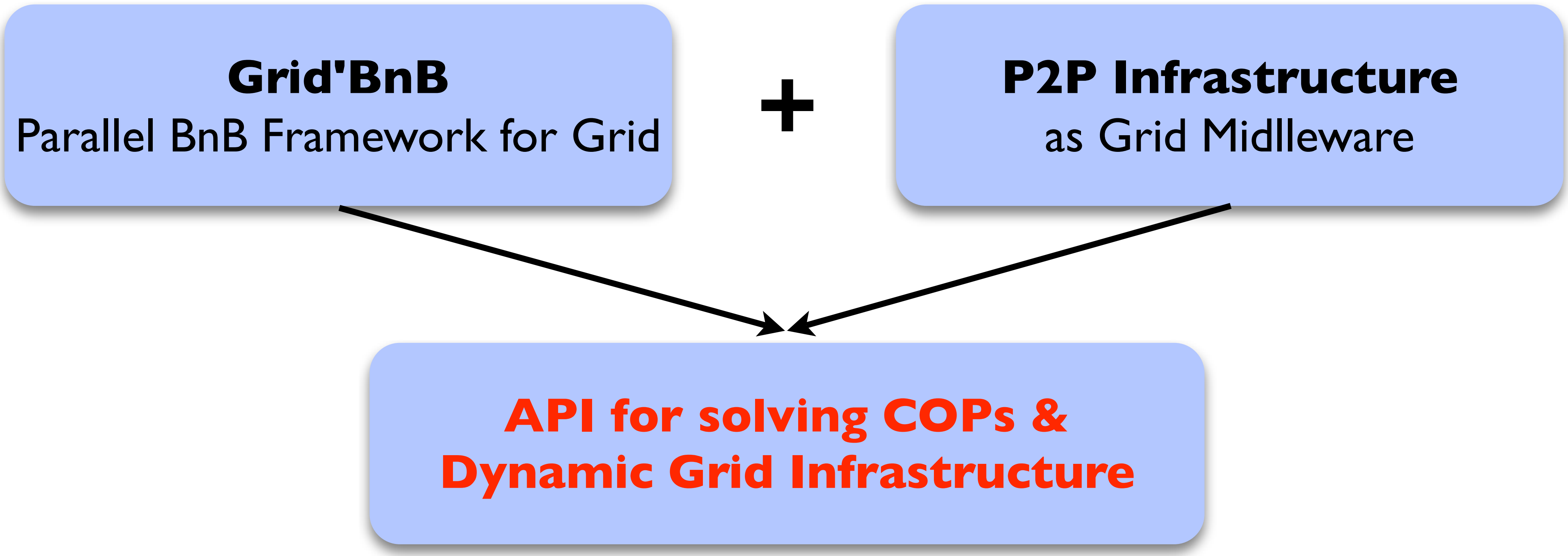
+

P2P Infrastructure
as Grid Midlleware

API for solving COPs &
Dynamic Grid Infrastructure

Validate by experiments on a Grid of Desktops & Clusters

Mixing Desktops & Clusters [PARCO07]



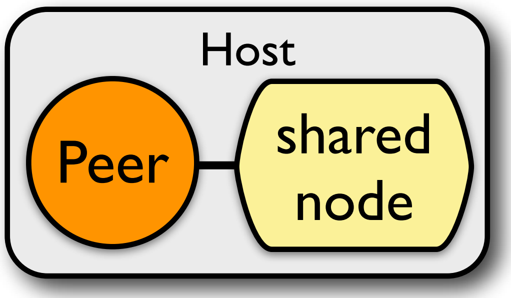
Validate by experiments on a Grid of Desktops & Clusters

New Problems:

firewalls ➟ forwarder

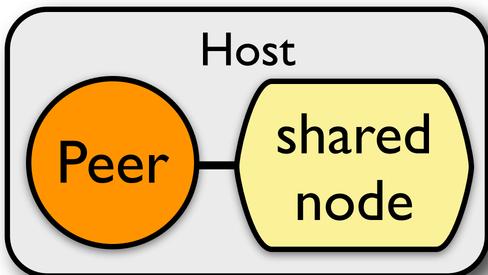
sharing clusters with the P2P infrastructure

Sharing Cluster's Nodes with P2P

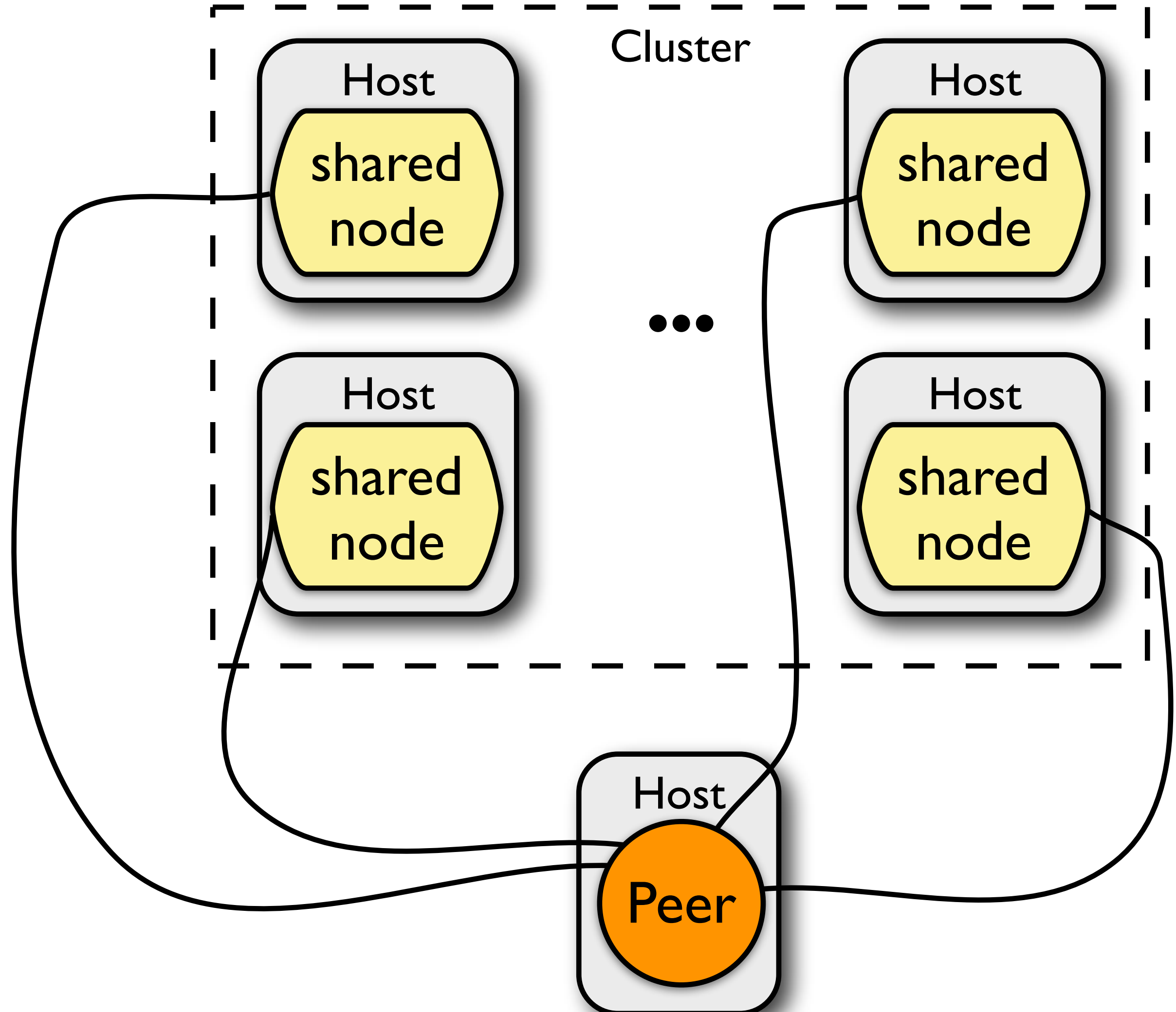


Peer sharing local node

Sharing Cluster's Nodes with P2P

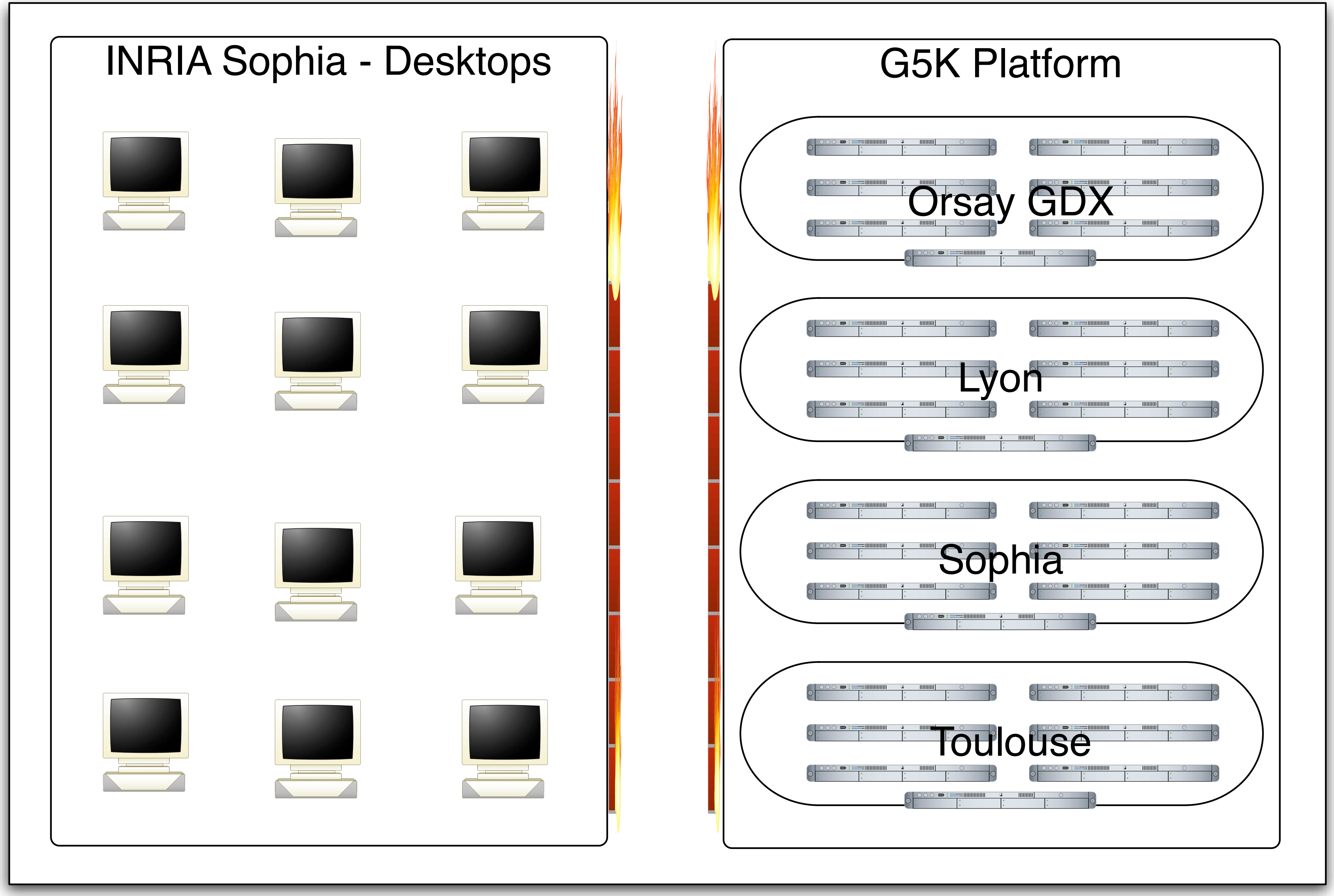


Peer sharing local node



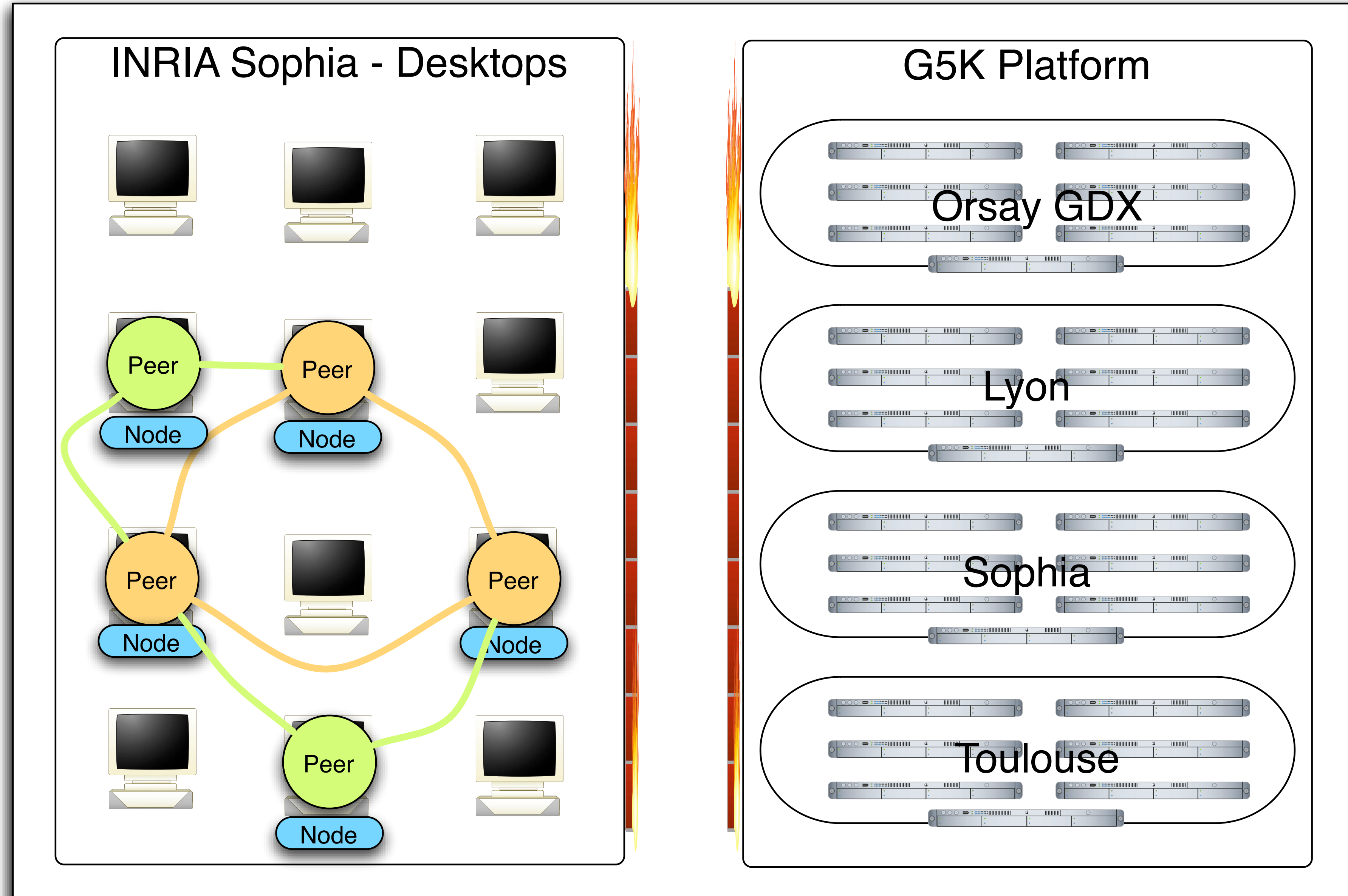
Peer sharing remote nodes

Testbed



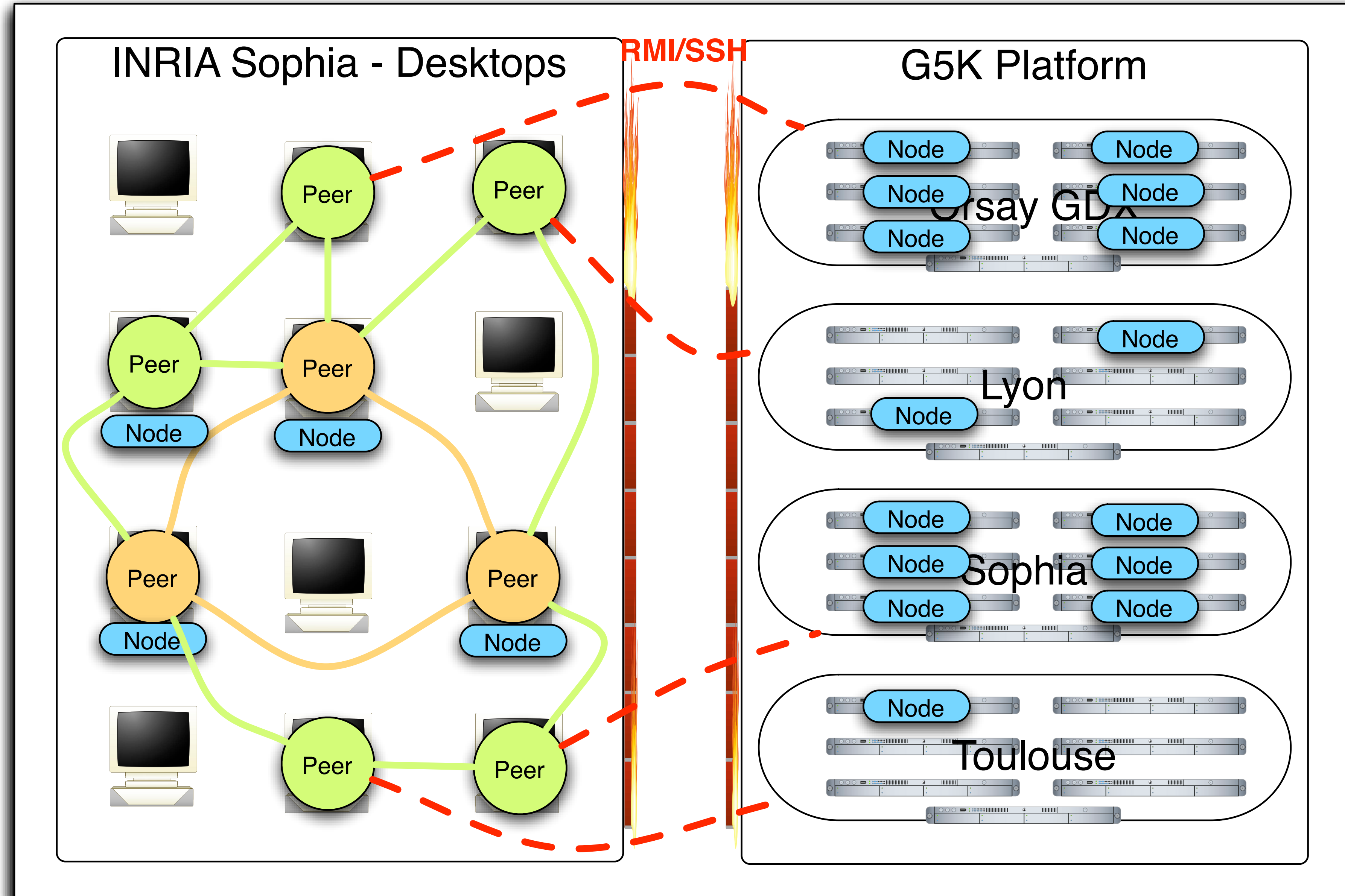
INRIA Sophia - Desktops
G5K Platform
Orsay GDX
Lyon
Sophia
Toulouse

Testbed



INRIA Sophia - Desktops
Peer
Node
Peer
Node
Peer
Node
Peer
Node
Peer
Node
G5K Platform
Orsay GDX
Lyon
Sophia
Toulouse

Testbed

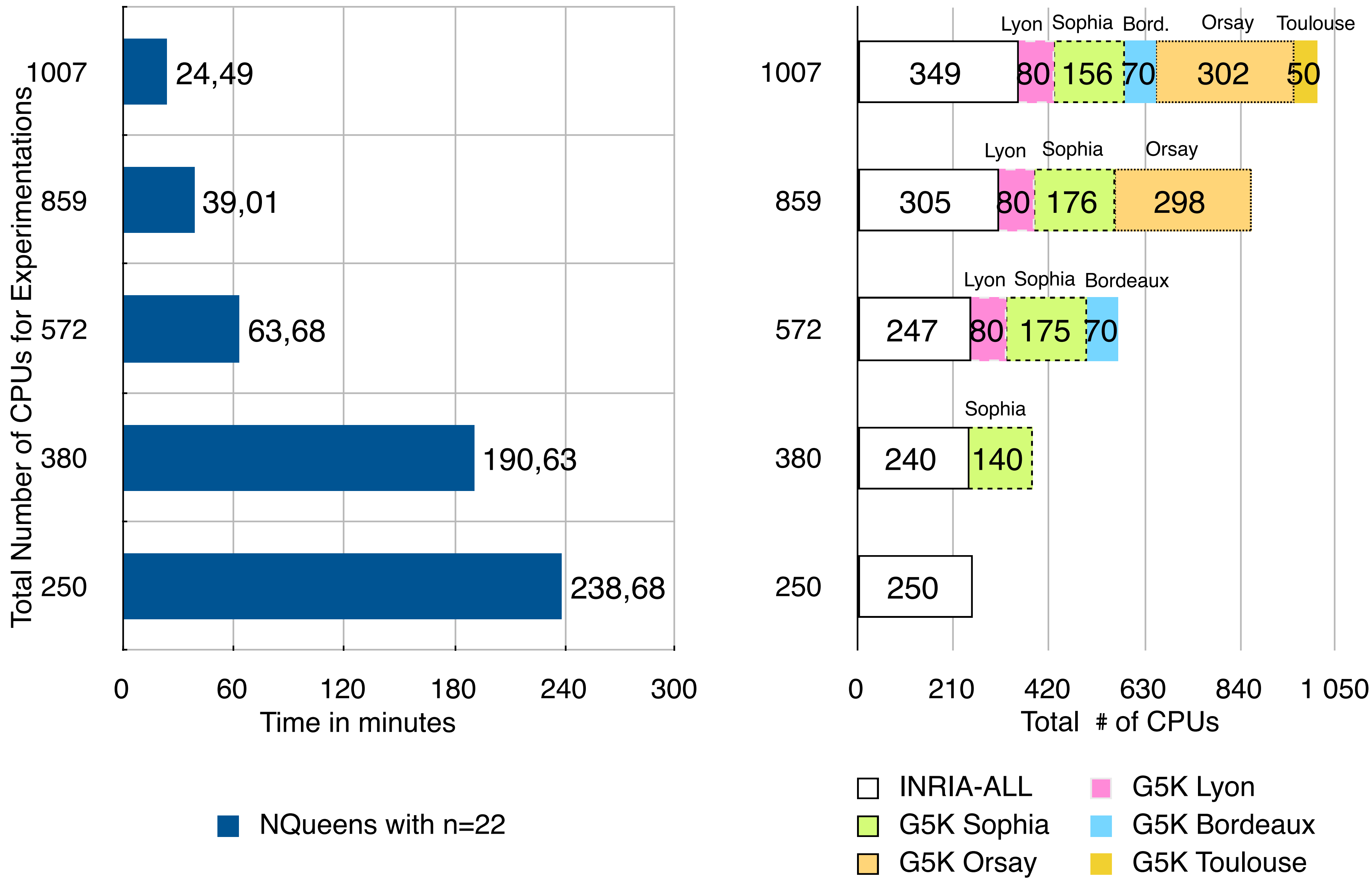


INRIA Sophia - Desktops
RMI/SSH
G5K Platform
Peer
Peer
Peer
Peer
Node
Peer
Node
Peer
Node
Peer
Node
Peer
Peer
Node
Orsay GDX
Node
Lyon
Sophia
Toulouse

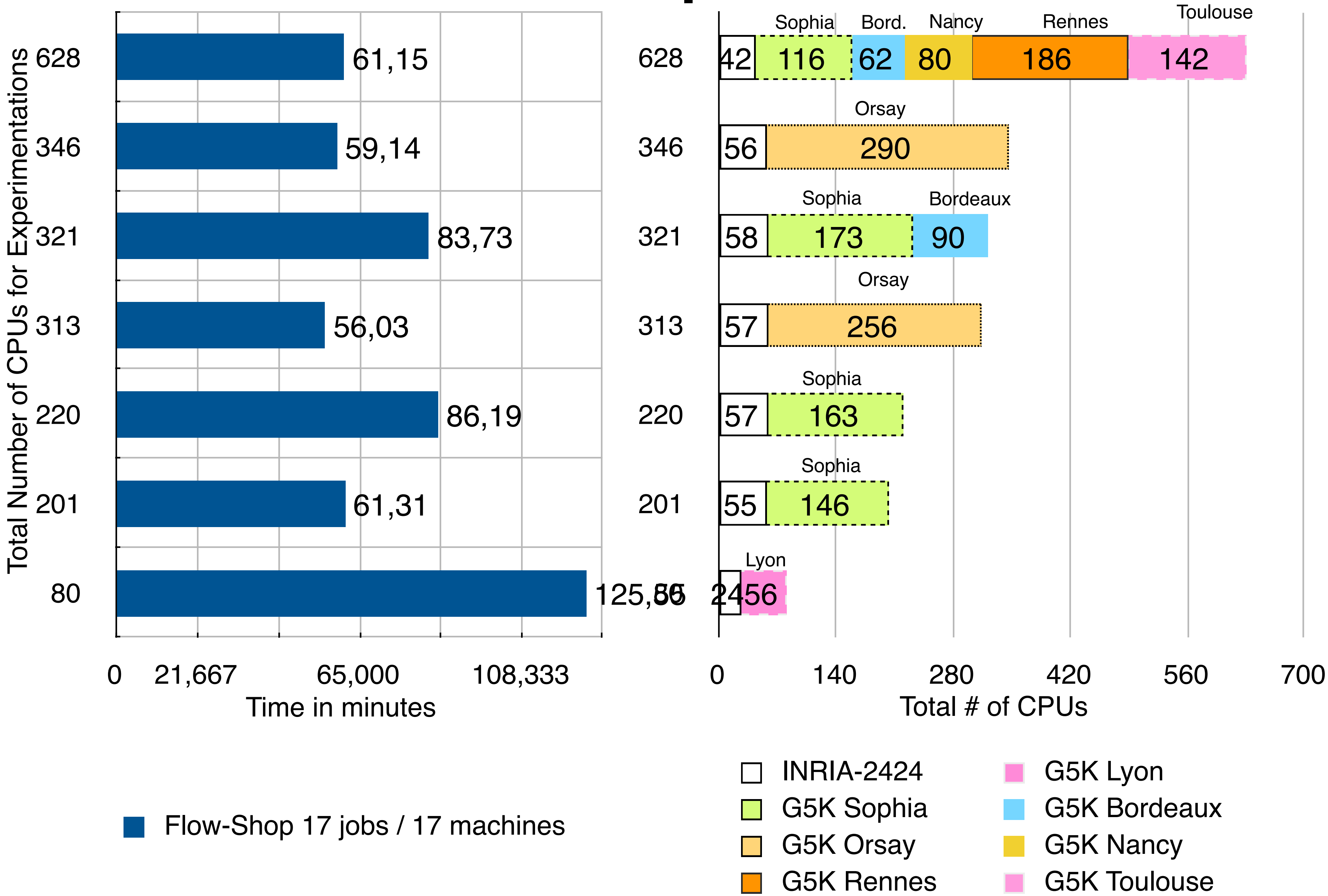
Large-Scale Experiments

- **Goal**: validate the infrastructure by experiments
- With **n-queens**:
  - no communication between workers
  - test the workability of the infrastructure
- With **flow-shop**:
  - communications between workers
  - test solving COPs

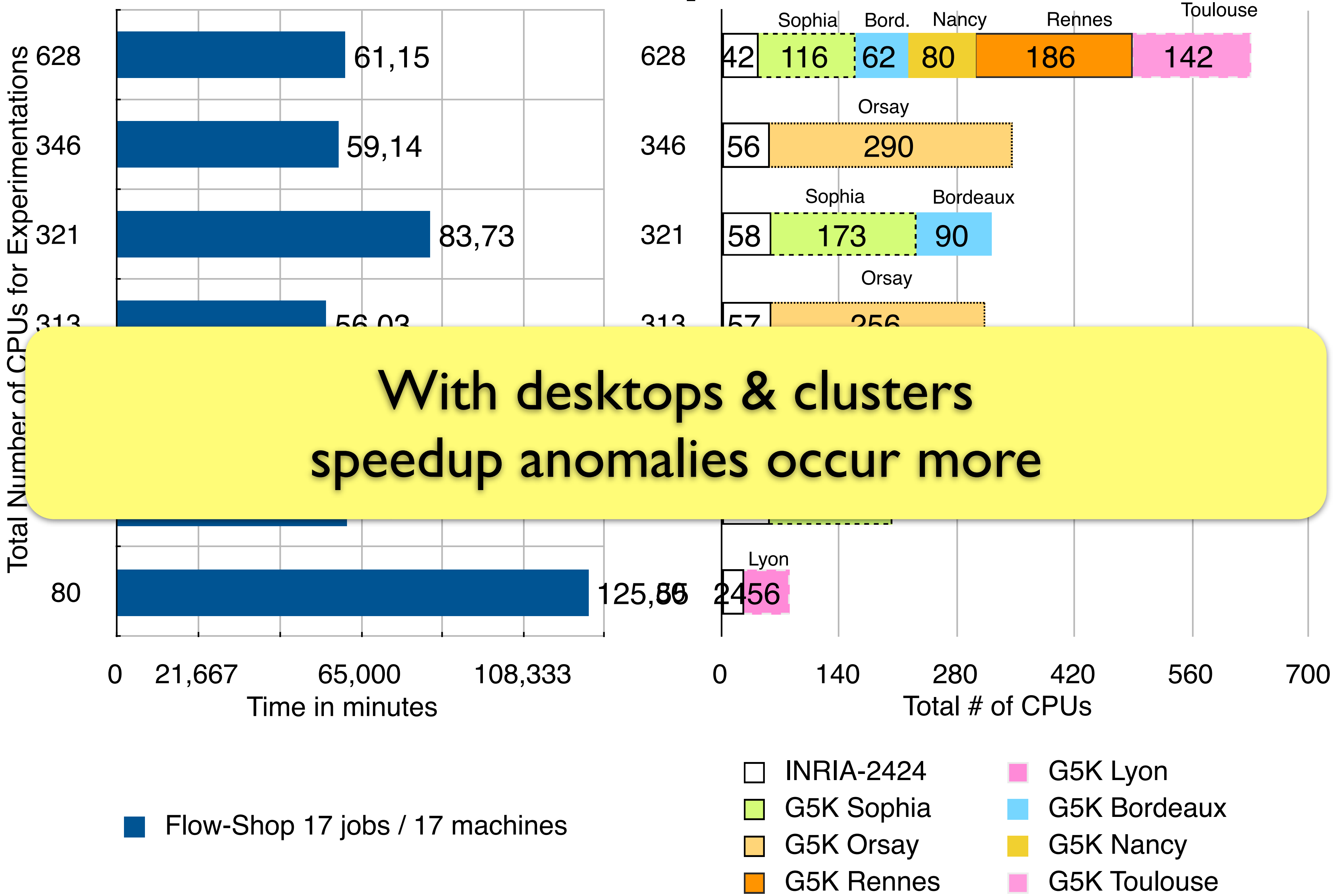
N-Queens Results



Flow-Shop Results



Flow-Shop Results



With desktops & clusters speedup anomalies occur more

Mixing - Analysis

- N-Queens problem scales well up to 1007 CPUs
  - 349 Desktops + 5 Clusters
- Flow-Shop up to 628 CPUs
  - 42 Desktops + 5 Clusters
  - worse performances than using only clusters (anomalies are more frequents)
- Experimented in closed environments -- security

- Grid'5000 platform's success ➟ hard for running long exp.

P2P as a Meta-Grid infrastructure

P2P as a Meta-Grid infrastructure

- **Observation** from Grid'5000:
  - 300 nodes available for 2 minutes
  - provide best-effort queue

P2P as a Meta-Grid infrastructure

- **Observation** from Grid'5000:
  - 300 nodes available for 2 minutes
  - provide best-effort queue
- **Idea**: take benefit from these nodes [Condor]
  - by hand: not easy
  - with the P2P infrastructure:
    - deploying peers in best-effort queue

P2P as a Meta-Grid infrastructure

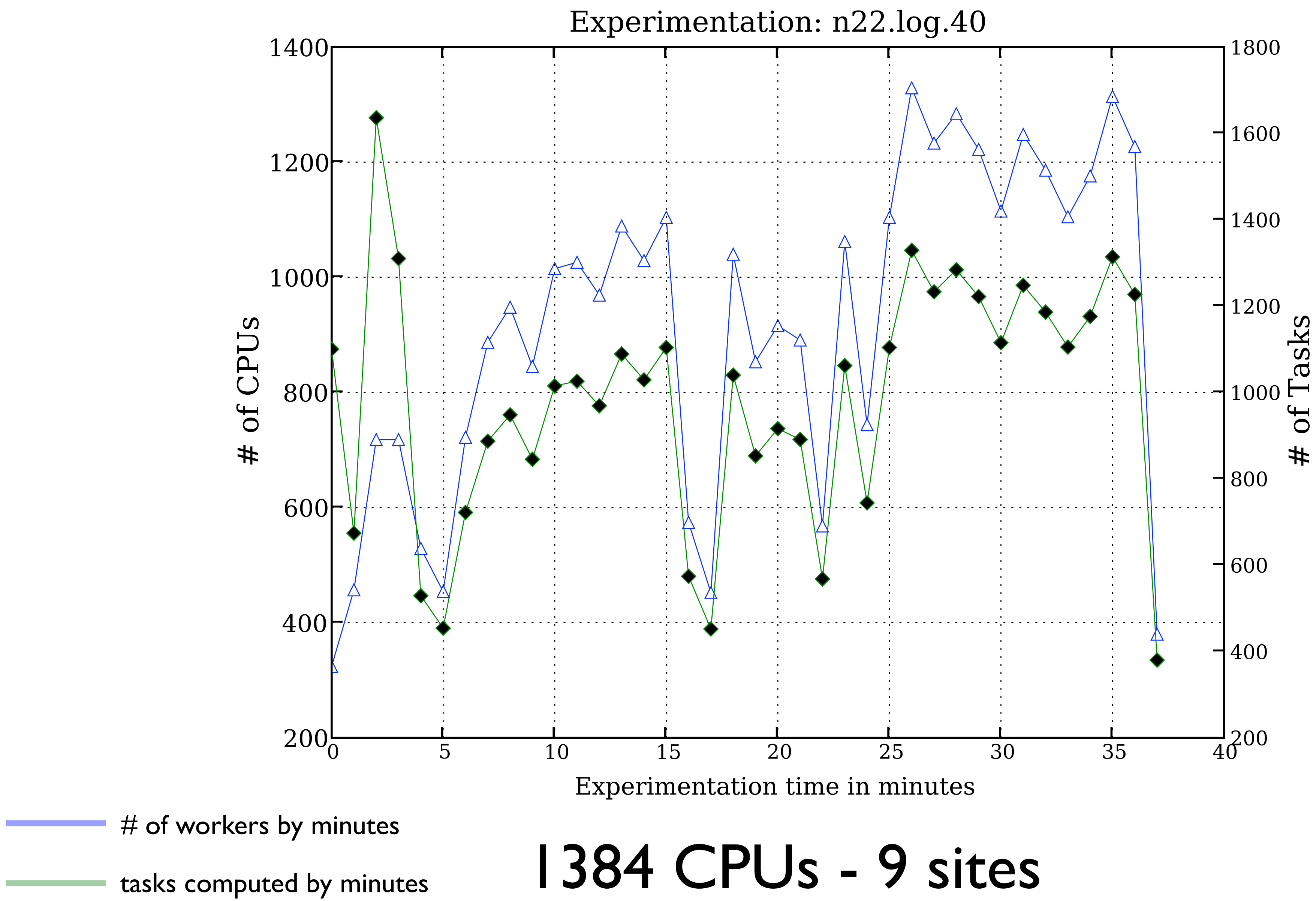
- **Observation** from Grid'5000:
  - 300 nodes available for 2 minutes
  - provide best-effort queue
- **Idea**: take benefit from these nodes [Condor]
  - by hand: not easy
  - with the P2P infrastructure:
    - deploying peers in best-effort queue
- **Result**: a permanent P2P infrastructure over Grid'5000

P2P as a Meta-Grid infrastructure

- **Observation** from Grid'5000:
  - 300 nodes available for 2 minutes
  - provide best-effort queue
- **Idea**: take benefit from these nodes [Condor]
  - by hand: not easy
  - with the P2P infrastructure:
    - deploying peers in best-effort queue
- **Result**: a permanent P2P infrastructure over Grid'5000

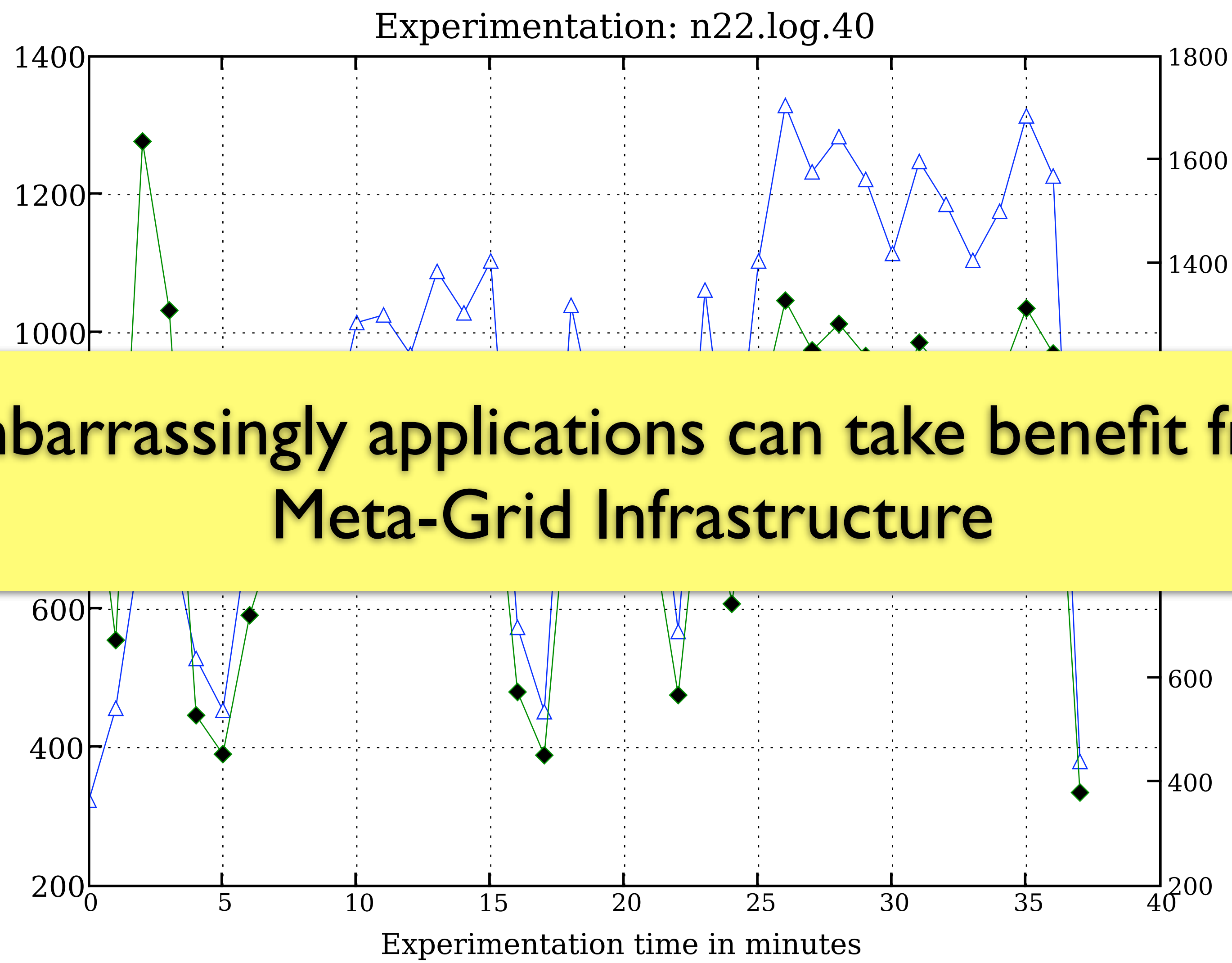
P2P infrastructure as a dynamic Grid middleware

P2P with N-Queens on Grid'5000



1384 CPUs - 9 sites

P2P with N-Queens on Grid'5000



Embarrassingly applications can take benefit from Meta-Grid Infrastructure

# of workers by minutes

tasks computed by minutes

1384 CPUs - 9 sites

Agenda

- Context, Problem, and Related Work
- Contributions
  - Branch-and-Bound Framework for Grids
  - Desktop Grid with Peer-to-Peer
  - Mixing Desktops & Clusters
- Perspectives & Conclusion

Summary

- Grid'BnB: a BnB framework for Grids
  - communication between workers
  - organizing workers in groups
- Grid infrastructure
  - based on P2P architecture
  - mixing desktops and clusters
  - deployed at INRIA Sophia lab

Perspectives

- Peer-to-Peer Infrastructure:
  - Job Scheduler
  - Resource Localization (PhD)
- Large-Scale Experiments:
  - International Grid: France, Japan, and Netherlands
  - Grid Pugtests
- Deployment:
  - Contracts in Grids (GCM Standard)
- Industrialization:
  - P2P ➟ Desktop Resource Virtualization
- CPER - P2P
  - 1M€/4years to professionalize the INRIA Grid

Conclusion

Conclusion

- Branch-and-Bound for Grids
  - solve COPs
  - hide Grid difficulties
  - communication between workers

Conclusion

- Branch-and-Bound for Grids
  - solve COPs
  - hide Grid difficulties
  - communication between workers

Conclusion

- Branch-and-Bound for Grids
  - solve COPs
  - hide Grid difficulties
  - communication between workers
- Peer-to-Peer as Grid infrastructure
  - mixing desktops and clusters

Conclusion

- Branch-and-Bound for Grids
  - solve COPs
  - hide Grid difficulties
  - communication between workers
- Peer-to-Peer as Grid infrastructure
  - mixing desktops and clusters

Conclusion

- Branch-and-Bound for Grids
  - solve COPs
  - hide Grid difficulties
  - communication between workers
- Peer-to-Peer as Grid infrastructure
  - mixing desktops and clusters
- Tested and experimented

Conclusion

- Branch-and-Bound for Grids
  - solve COPs
  - hide Grid difficulties
  - communication between workers
- Peer-to-Peer as Grid infrastructure
  - mixing desktops and clusters
- Tested and experimented
- Available in open source

Conclusion

- Branch-and-Bound for Grids
    - solve COPs
    - hide Grid difficulties
    - communication between workers
- Peer-to-Peer as Grid infrastructure
    - mixing desktops and clusters
- Tested and experimented
- Available in open source

➟ **Provides framework and infrastructure to hide Grid difficulties**

[SCCC05] Balancing Active Objects on a Peer to Peer Infrastructure
Javier Bustos-Jimenez, Denis Caromel, Alexandre di Costanzo, Mario Leyton and Jose M. Piquer. Proceedings of the XXV International Conference of the Chilean Computer Science Society (SCCC 2005), Valdivia, Chile, November 2005.

[HIC06] Executing Hydrodynamic Simulation on Desktop Grid with ObjectWeb ProActive
Denis Caromel, Vincent Cavé, Alexandre di Costanzo, Céline Brignolles, Bruno Grawitz, and Yann Viala. HIC2006: Proceedings of the 7th International Conference on HydroInformatics, Nice, France, September 2006.

[HiPC07] A Parallel Branch & Bound Framework for Grids
Denis Caromel, Alexandre di Costanzo, Laurent Baduel, and Satoshi Matsuoka. Grid'BnB: HiPC'07, Goa, India, December 2007.

[CMST06] ProActive: an Integrated platform for programming and running applications on grids and P2P systems
Denis Caromel, Christian Delbé, Alexandre di Costanzo, and Mario Leyton. Journal on Computational Methods in Science and Technology, volume 12, 2006.

[PARCO07] Peer-to-Peer for Computational Grids: Mixing Clusters and Desktop Machines
Denis Caromel, Alexandre di Costanzo, and Clément Mathieu. Parallel Computing Journal on Large Scale Grid, 2007.

[FGCS07] Peer-to-Peer and Fault-tolerance: Towards Deployment-based Technical Services
Denis Caromel, Alexandre di Costanzo, and Christian Delbé. Journal Future Generation Computer Systems, to appear, 2007.

and Workshops, Master report, ...