LaPIe: Collective Communications adapted to Grid Environments

Luiz Angelo Barchet-Estefanel

Thesis Supervisor: M Denis TRYSTRAM
Co-Supervisor: M Grégory MOUNIE

ID-IMAG Laboratory
Grenoble - France

Introduction to Parallel Processing

Fact

The demand for computing power will always grow up

There are two options to increase the available computer power:

Introduction to Parallel Processing

Fact

The demand for computing power will always grow up

There are two options to increase the available computer power:

- Buy a bigger computer - $$$$$

Introduction to Parallel Processing

Fact

The demand for computing power will always grow up

There are two options to increase the available computer power:

- Buy a bigger computer - $$$$$
- Use several computers

Parallel Processing

- Divide a problem into multiple fragments that can be executed in parallel

Introduction to Grids/Metacomputing

Definition

- Aggregation of geographically distributed computers
- Mainly clusters of computers

Fact

The Grid hardware already exists

- Interconnexion of several clusters and NOWs

The Grid software only emerges

- Most difficulties come from the resource heterogeneity

Communications in a Grid

Influence of resource heterogeneity

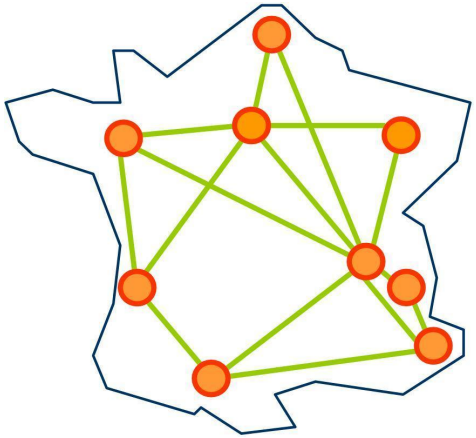
Geographically distributed systems

- Different communication latencies

Heterogeneous communication infrastructures

- Transfer bandwidth

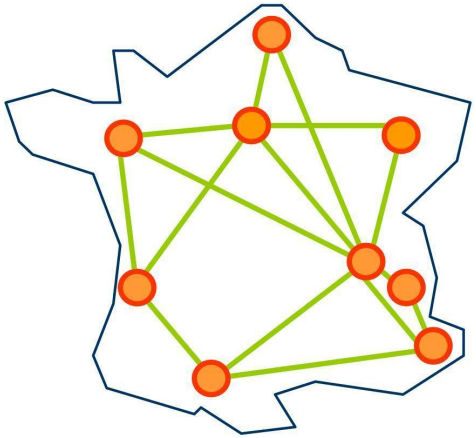
Example: GRID 5000



	Latency	Bandwidth*
Myrinet	10 μs	250 MB/s
Giga Ethernet	50 μs	**120 MB/s**

average bandwidth for a 32MB message sent with MPI

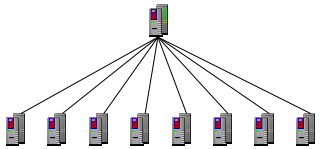
Example: GRID 5000

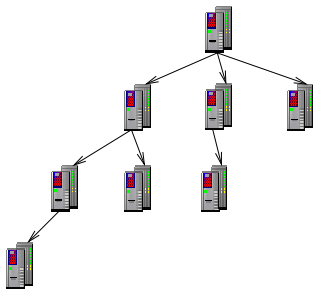


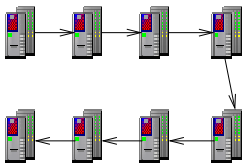
	Latency	Bandwidth
Myrinet	10 μs	250 MB/s
Giga Ethernet	50 μs	**120 MB/s**
WAN Connection	5000 μs	**6-120 MB/s**

average bandwidth for a 32MB message sent with MPI

Collective Communications



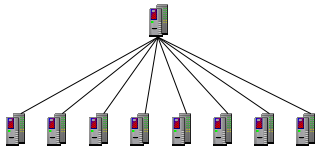


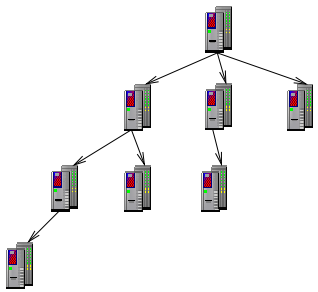


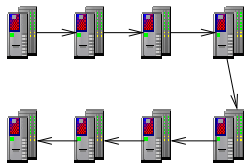
Definition

- Collective communication is defined as communication that involves a group of processes
  - Different communication patterns

Collective Communications







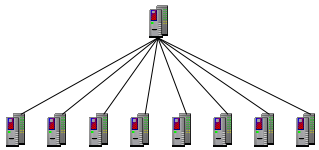
Definition

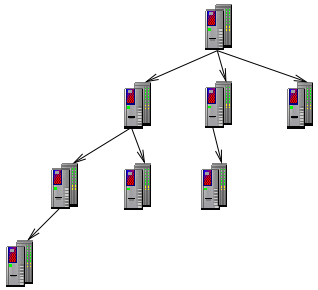
- Collective communication is defined as communication that involves a group of processes
  - Different communication patterns

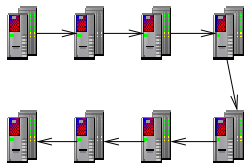
Most programming environments include collective communication primitives

- PVM, MPI, Athapascan, etc.
- Consensus, Group Membership, etc.

Collective Communications



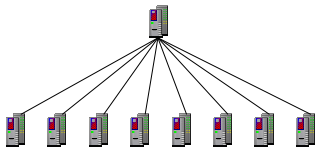


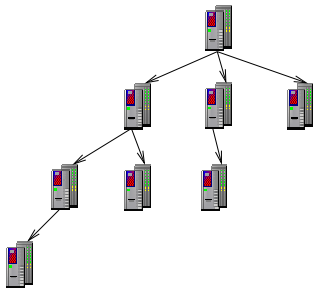


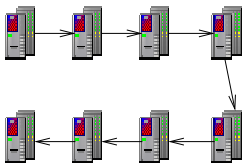
Impact of communication heterogeneity

- Absence of a single efficient strategy

Collective Communications







Impact of communication heterogeneity

- Absence of a single efficient strategy
- Performance depends on:
  - communication pattern
  - network characteristics
  - operation parameters (# of nodes, message size, etc.)

Overview of this work

Our goal: improve communication scheduling on grid environments through the use of an hierarchical network modelling

- provide efficient grid-aware collective communication operations

What we need:

- qualitative knowledge of the network topology
  - detect network heterogeneity
- quantitative knowledge of the network interconnexions
  - identify latency and bandwidth among different nodes

Overview of this work

Our approach: use hybrid algorithms

- dynamic scheduling of inter-cluster communication
- efficient ”static” algorithms for intra-cluster communication

Technical validation: evaluation through synthetic experiences

- performances are close to those experienced by real applications
- fast prototyping

Outline

Optimising Collective Communications

Objective: minimise the overall execution time

- improve data distribution
- reduce communications through slow links

Optimising Collective Communications

Objective: minimise the overall execution time

- improve data distribution
- reduce communications through slow links

Heterogeneous Systems - Grids

- communication scheduling according to the network characteristics

Optimising Collective Communications

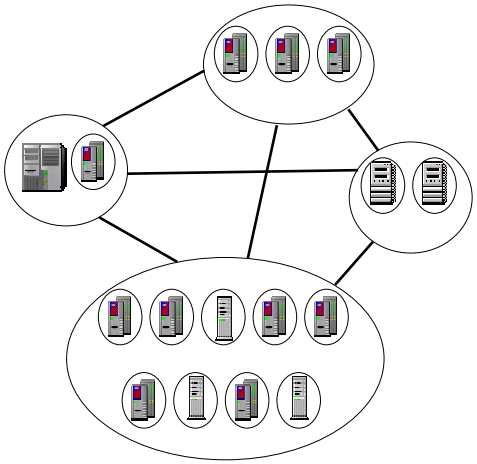
Objective: minimise the overall execution time

- improve data distribution
- reduce communications through slow links

Heterogeneous Systems - Grids

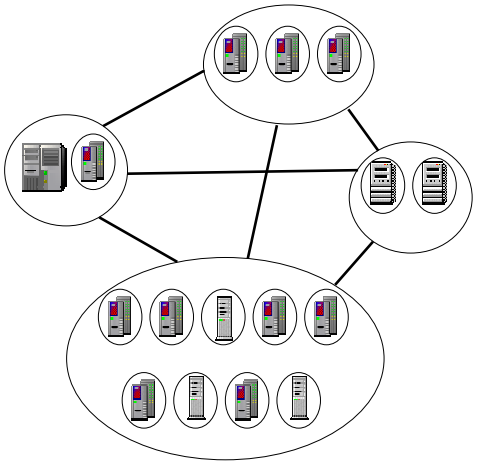
- communication scheduling according to the network characteristics
- **NP-Complete**
  - no accurate analytical models are available

Hierarchical Structure



”Flat Tree” approach

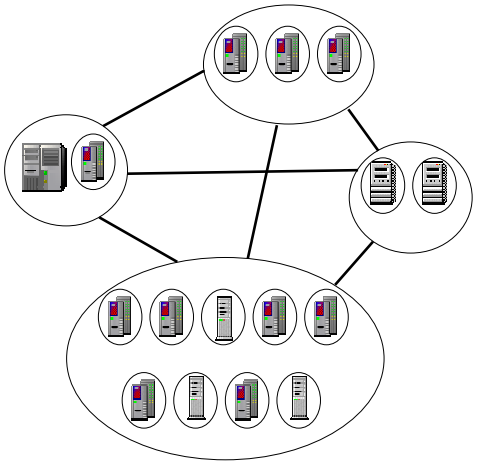
Hierarchical Structure



”Flat Tree” approach

- Objective: minimise distant communications

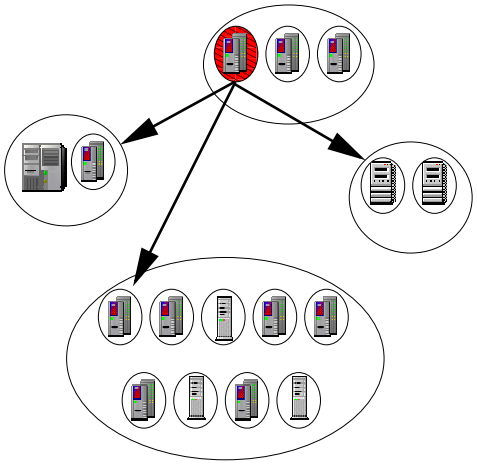
Hierarchical Structure



”Flat Tree” approach

- Objective: minimise distant communications
- Communication is divided in two layers

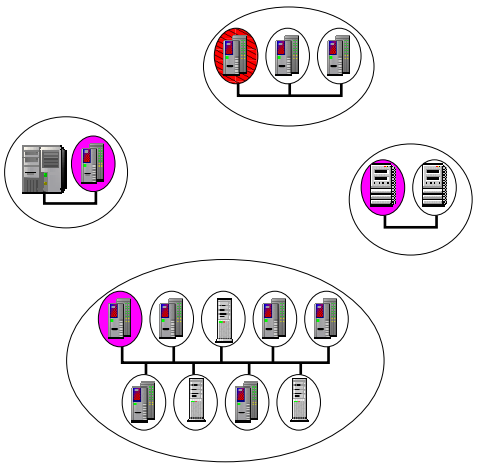
Hierarchical Structure



”Flat Tree” approach

- Objective: minimise distant communications
- Communication is divided in two layers
  - Distant nodes

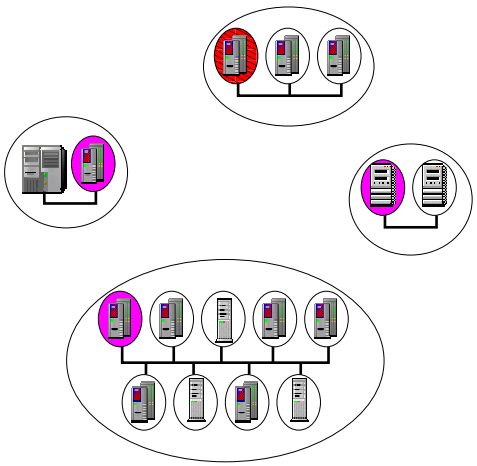
Hierarchical Structure



”Flat Tree” approach

- Objective: minimise distant communications
- Communication is divided in two layers
  - Distant nodes
  - Local nodes

Hierarchical Structure



”Flat Tree” approach

- Objective: minimise distant communications
- Communication is divided in two layers
  - Distant nodes
  - Local nodes

ECO (Lowekamp 96) - PVM library
MagPIe (Kielmann 99) - MPI library

Analysis of this approach

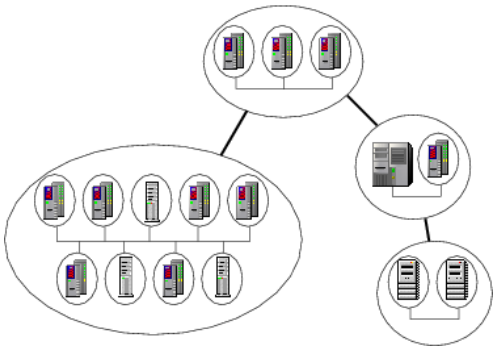
Advantages

- Easy to implement
- Minimises communication across slow links

Limitations

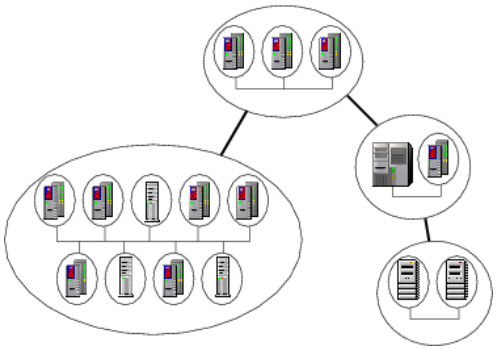
- Too tight scheduling
  - communication hierarchy does not make difference between links capacities/latencies
- The root process handles all long distance transmissions
  - does not explore parallel transmissions

Multi-layered Hierarchy



Multi-layered communications

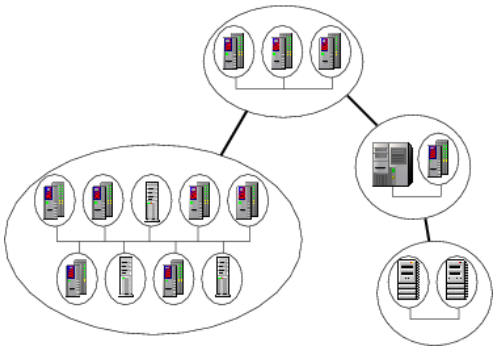
Multi-layered Hierarchy



Multi-layered communications

- Structured according to the relative performance of each layer
  - WAN > MAN > LAN > SMP

Multi-layered Hierarchy



Multi-layered communications

- Structured according to the relative performance of each layer
  - WAN > MAN > LAN > SMP

MPICH-G2 (Karonis 02) - MPI library

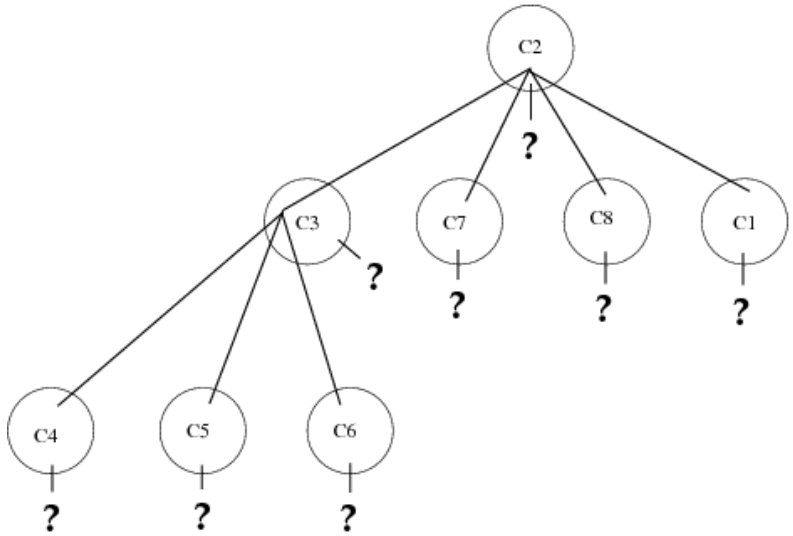
Analysis of this approach

Advantages

- More flexible structure
- Based on the relative communication performance

Limitation

- Hierarchy does not takes into account the communication cost inside each cluster



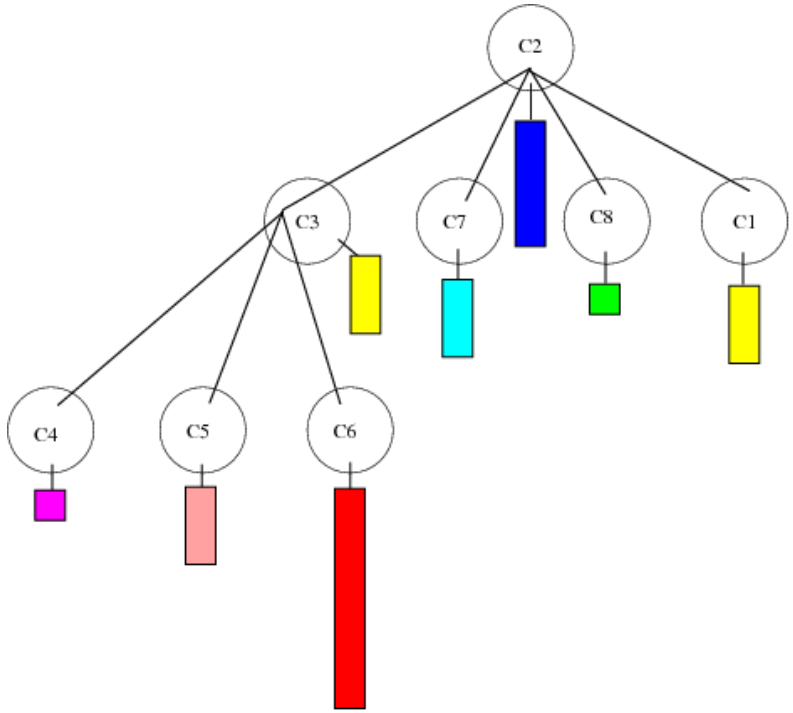
Analysis of this approach

Advantages

- More flexible structure
- Based on the relative communication performance

Limitation

- Hierarchy does not takes into account the communication cost inside each cluster



How to improve Grid communications

Is it possible to better schedule communications in a grid environment?

- Dynamically generated hierarchy
  - network parameters, message size and communication pattern
- Fully Grid-aware
  - includes the communication cost inside each cluster

Our Approach

- Simplify the network description
  - focus on topology discovery and clustering
- Augment the information about clusters' performance
  - performance models to predict the communication cost
- Improve the usage of multi-layered hierarchy
  - grid-aware scheduling heuristics

Outline

Identifying Logical Clusters

Approaches

- Locality of the nodes
- User-defined mappings
- Network discovery tools

Identifying Logical Clusters

Approaches

- Locality of the nodes
- User-defined mappings
- Network discovery tools

Locality of the nodes

- Simple
- Does not express clusters' internal heterogeneity
- Does not consider interconnection parameters

Identifying Logical Clusters

Approaches

- Locality of the nodes
- User-defined mappings
- Network discovery tools

User defined topology

- Expensive and hard to do
- Sufficiently accurate (?)
- Normally falls back to the locality of the nodes

Identifying Logical Clusters

Approaches

- Locality of the nodes
- User-defined mappings
- Network discovery tools

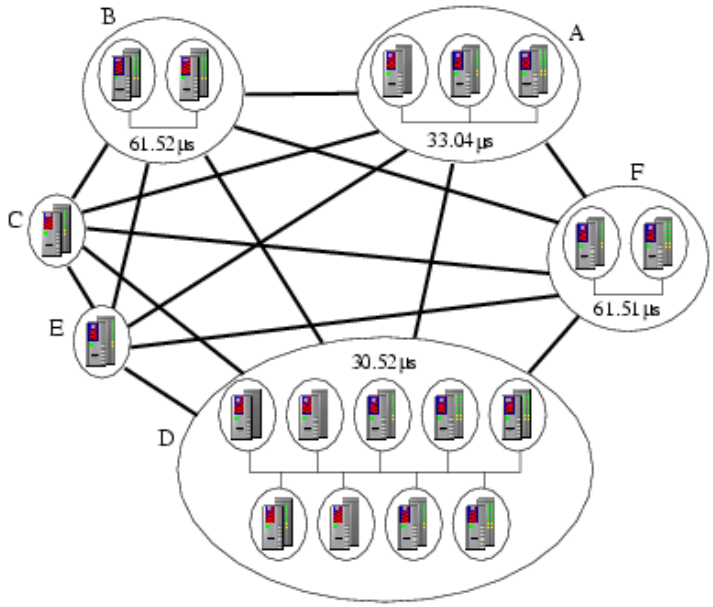
Some network tools

- NWS - measures latency and bandwidth between nodes
- REMOS - uses SNMP to construct a low-level topology
- TopoMon - identifies shared links

What we need

Application-level topology discovery

- identification of homogeneous ”islands”
- fast deployment



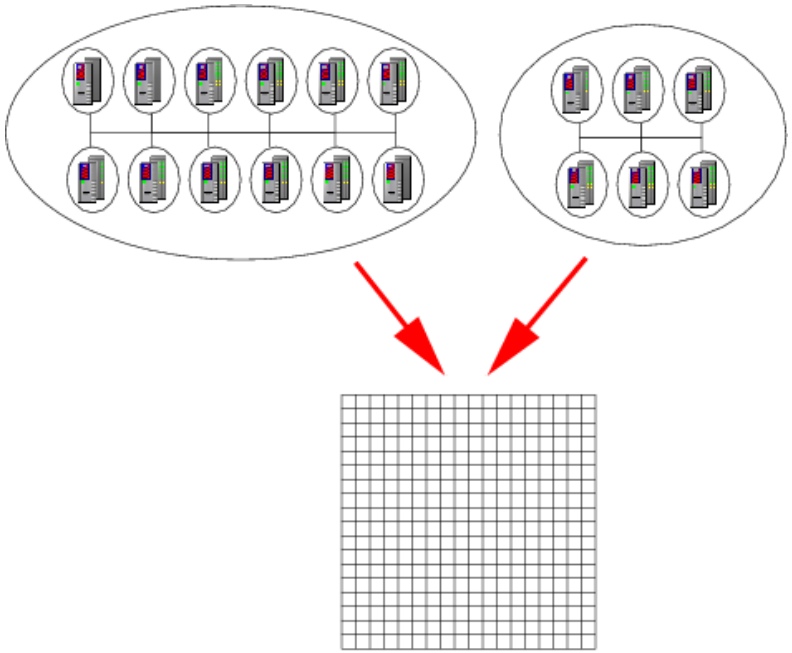
Latency between Subnets(µs)

	A	B	C	D	E	F
A	0	61.53	105.25	37.99	103.45	69.96
B	61.53	0	224.98	61.52	139.04	137.52
C	105.25	224.98	0	61.49	207.98	129.45
D	37.99	61.52	61.49	0	66.49	61.51
E	103.45	139.04	207.98	66.49	0	123.97
F	69.96	137.52	129.45	61.51	123.97	0

Topology Discovery

First Phase: identify network heterogeneity

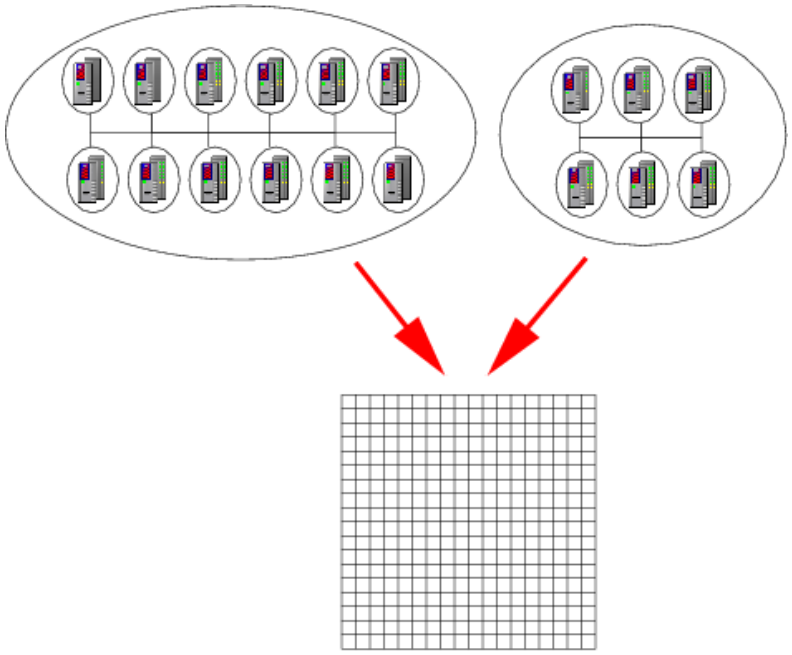
- use of NWS-like tools



Topology Discovery

First Phase: identify network heterogeneity

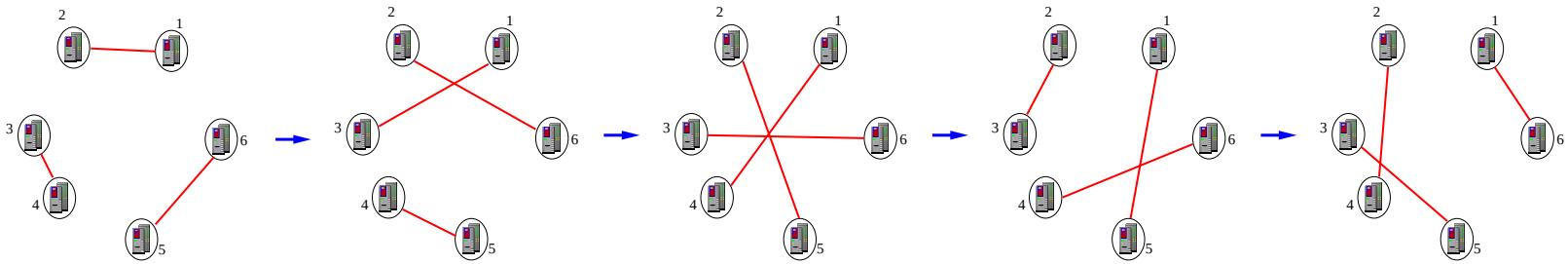
- use of NWS-like tools
- construct a $n \times n$ distance matrix
  - latency



Details

How to minimise the probing time

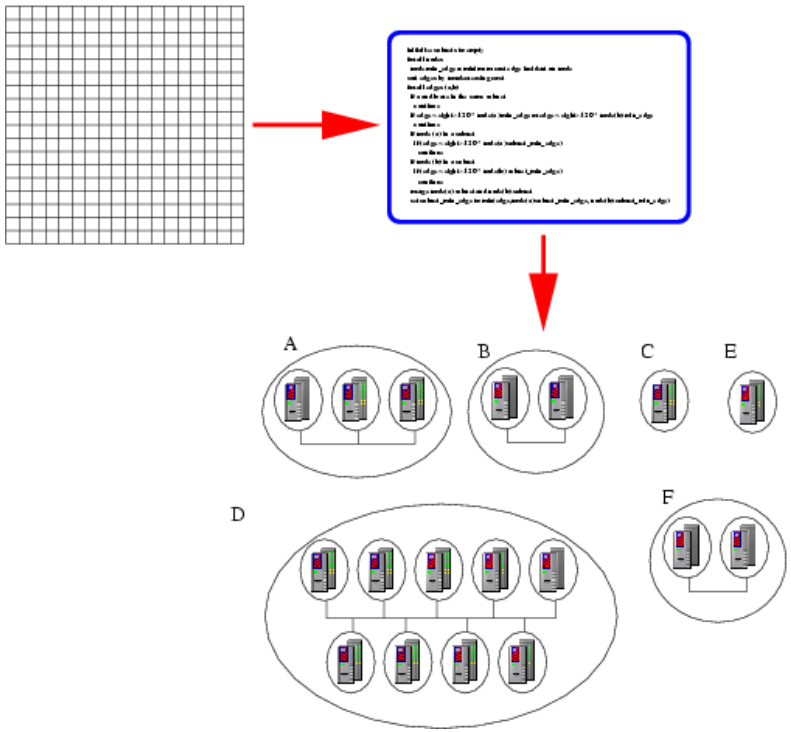
- Latency measure is short enough to not disturb the network
- Schedule parallel probes among independent pairs



Topology Discovery

Second Phase: clustering

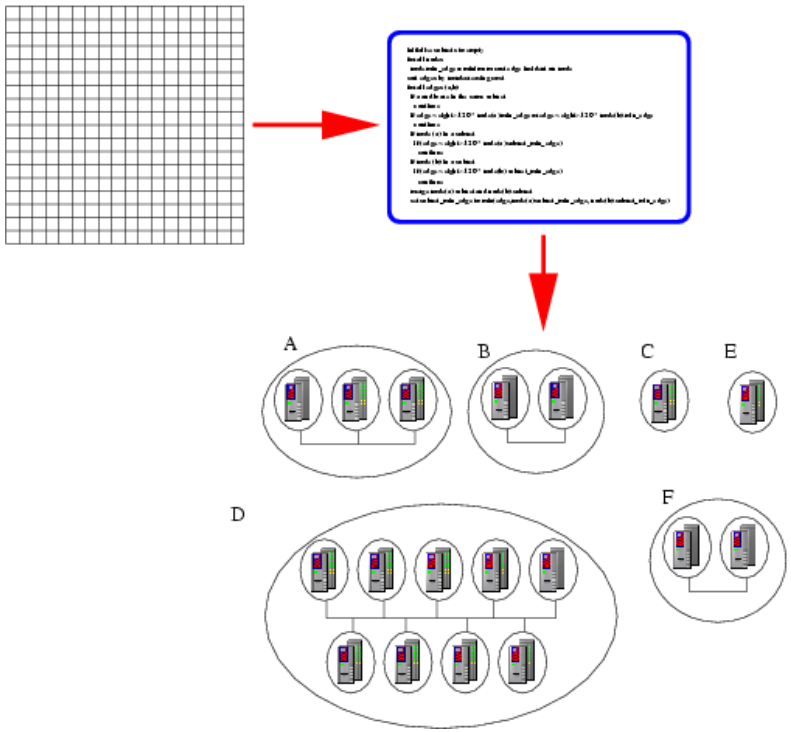
- use of a clustering algorithm (ECO)
  - Tolerance factor $\rho = 30\%$



Topology Discovery

Second Phase: clustering

- use of a clustering algorithm (ECO)
  - Tolerance factor $\rho = 30\%$
- Formatted output (*magpie_clusters* file)



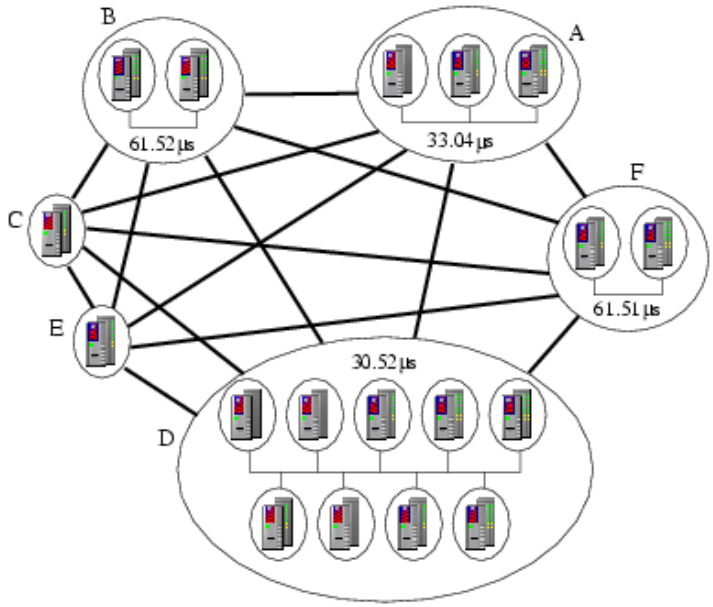
Topology Description

```
cluster 0
process 0 48 49 50 51 52 53 54 55 56 57 58 59 60 61
62 63 65 66 67 68 69 70 71 72 73 74 75 76 77
cluster 1
process 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 64 79 80 81 82 83 84 85 86 87
cluster 2
process 20 21 22 23 25 26
cluster 3
process 24
cluster 4
process 27
cluster 5
process 28 29 30 31 32 33 34 35 36 37 38 39 40 41
42 43 44 45 46 47
```

Topology Discovery

Third Phase: obtaining network parameters

- Reduced set of measures
  - one node from each cluster



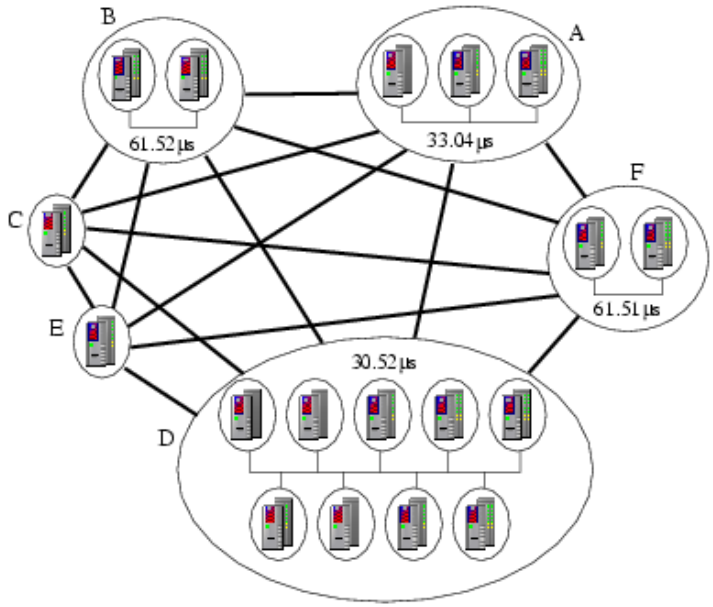
Latency between Subnets(µs)

	A	B	C	D	E	F
A	0	61.53	105.25	37.99	103.45	69.96
B	61.53	0	224.98	61.52	139.04	137.52
C	105.25	224.98	0	61.49	207.98	129.45
D	37.99	61.52	61.49	0	66.49	61.51
E	103.45	139.04	207.98	66.49	0	123.97
F	69.96	137.52	129.45	61.51	123.97	0

Topology Discovery

Third Phase: obtaining network parameters

- Reduced set of measures
  - one node from each cluster
  - $O(C^2)$ measures



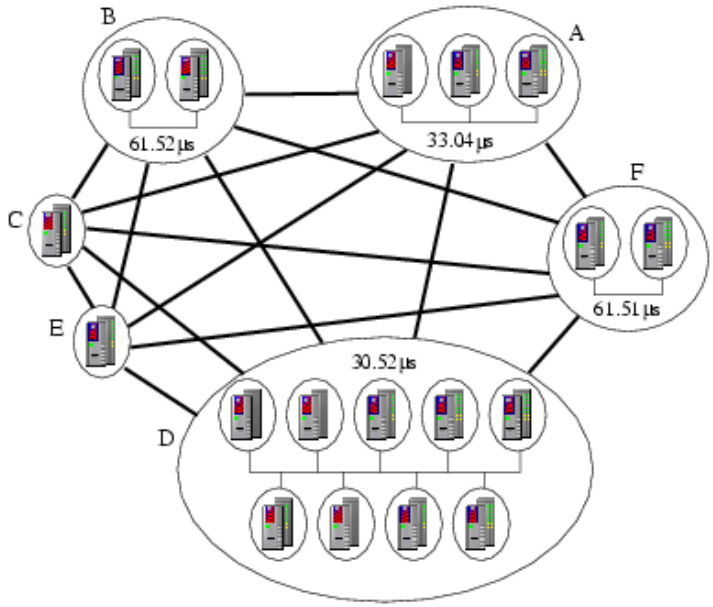
Latency between Subnets(s)

	A	B	C	D	E	F
A	0	61.53	105.25	37.99	103.45	69.96
B	61.53	0	224.98	61.52	139.04	137.52
C	105.25	224.98	0	61.49	207.98	129.45
D	37.99	61.52	61.49	0	66.49	61.51
E	103.45	139.04	207.98	66.49	0	123.97
F	69.96	137.52	129.45	61.51	123.97	0

Topology Discovery

Third Phase: obtaining network parameters

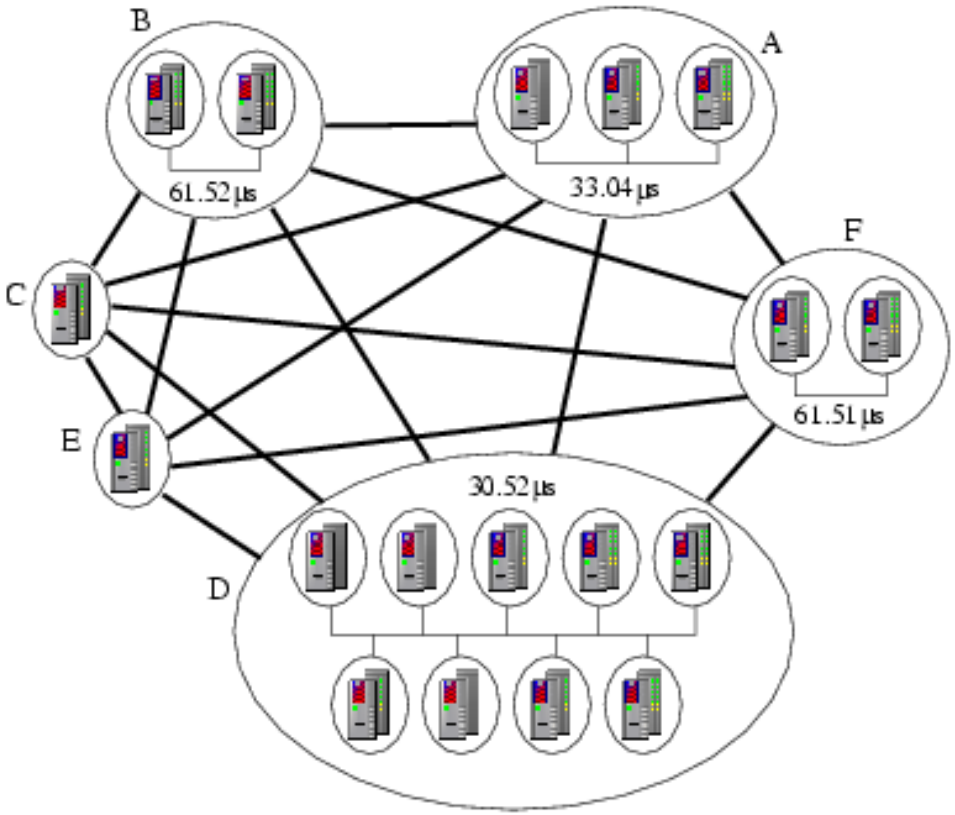
- Reduced set of measures
  - one node from each cluster
  - $O(C^2)$ measures
- Merge of this information with network topology



Latency between Subnets(µs)

	A	B	C	D	E	F
A	0	61.53	105.25	37.99	103.45	69.96
B	61.53	0	224.98	61.52	139.04	137.52
C	105.25	224.98	0	61.49	207.98	129.45
D	37.99	61.52	61.49	0	66.49	61.51
E	103.45	139.04	207.98	66.49	0	123.97
F	69.96	137.52	129.45	61.51	123.97	0

Example: the IDPOT cluster



Latency between Subnets(µs)

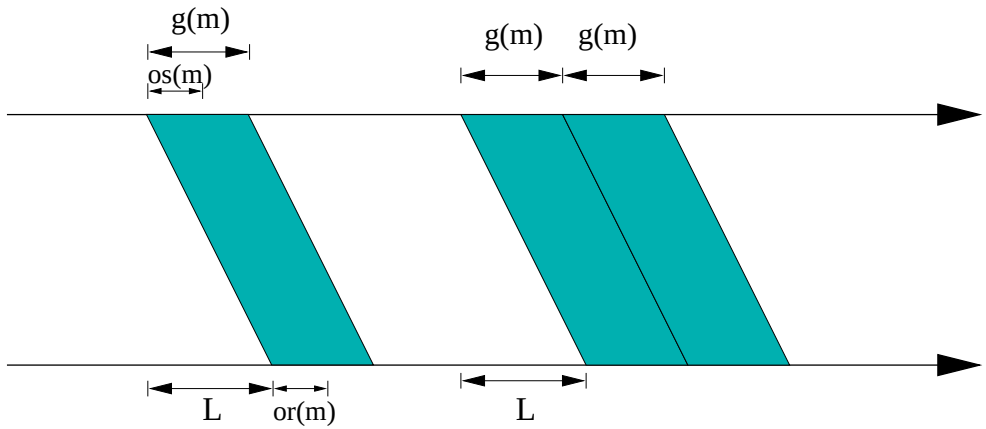
	A	B	C	D	E	F
A	0	61.53	105.25	37.99	103.45	69.96
B	61.53	0	224.98	61.52	139.04	137.52
C	105.25	224.98	0	61.49	207.98	129.45
D	37.99	61.52	61.49	0	66.49	61.51
E	103.45	139.04	207.98	66.49	0	123.97
F	69.96	137.52	129.45	61.51	123.97	0

Outline

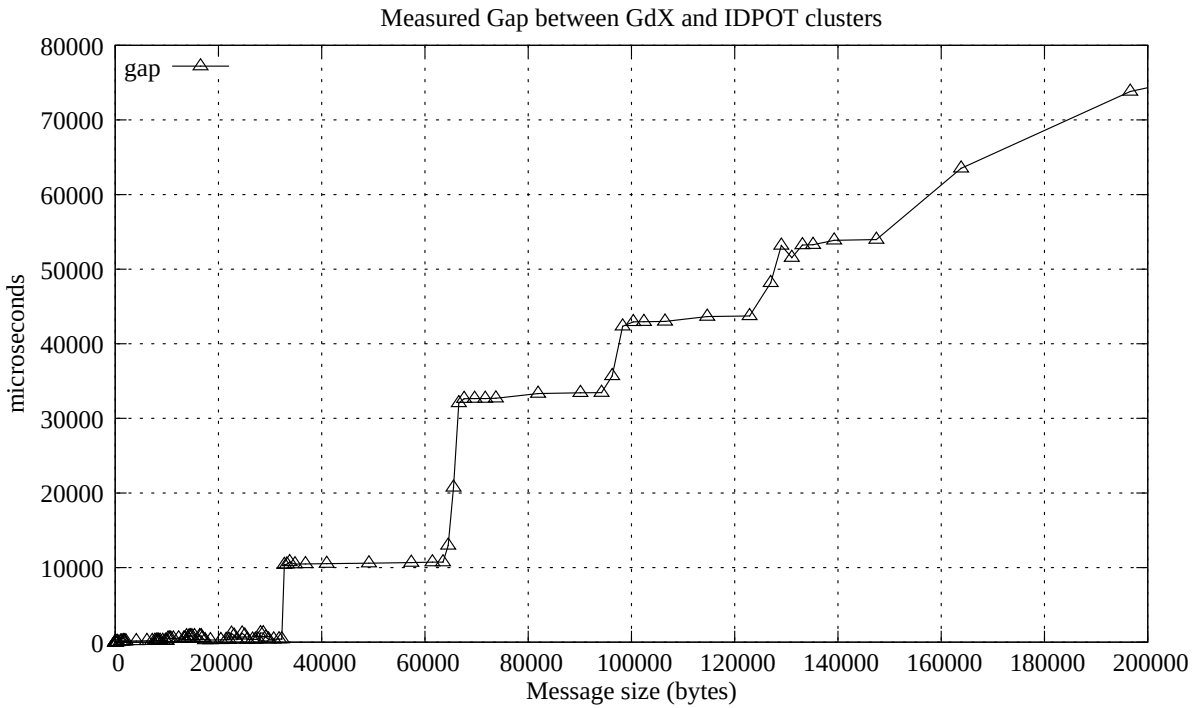
Modelling Collective Communications

We use pLogP cost model (Kielmann *et al.*)

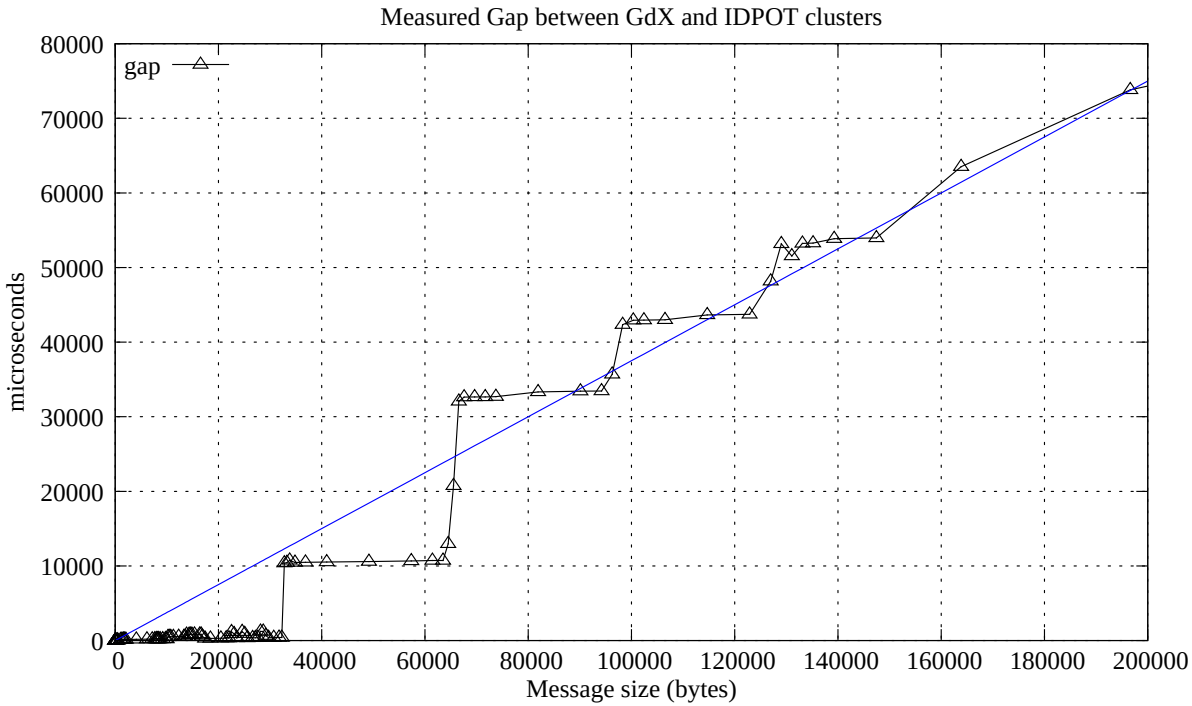
- Number of processes - *P*
- Latency - *L*
- Communication *gap* - *g(m)*
  - Send and receive overhead - *os(m)*, *or(m)*



Advantages of pLogP



Comparing with the Hockney model



pLogP allows a theoretical modelling that is close to the reality

Example: modelling MPI_Bcast

Definition

- One process (root) send the same message to every process in the group

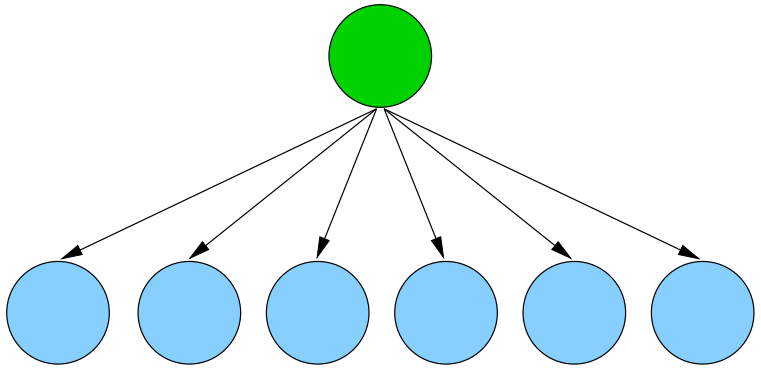
Example: modelling MPI_Bcast

Definition

- One process (root) send the same message to every process in the group

Strategies

- Flat Tree
- Binary Tree
- Binomial Tree
- Chain (pipeline)
- ...



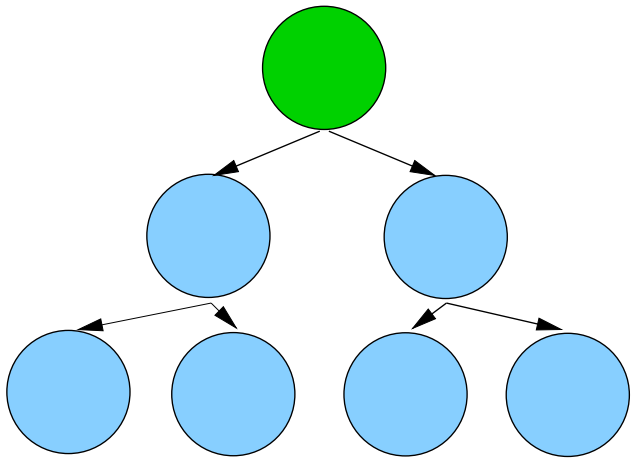
Example: modelling MPI_Bcast

Definition

- One process (root) send the same message to every process in the group

Strategies

- Flat Tree
- Binary Tree
- Binomial Tree
- Chain (pipeline)
- ...



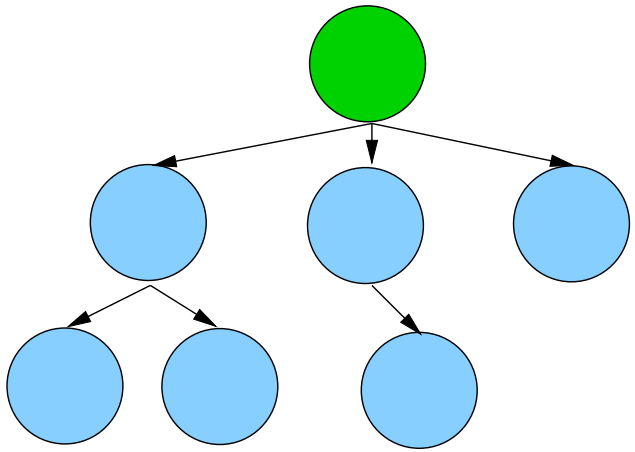
Example: modelling MPI_Bcast

Definition

- One process (root) send the same message to every process in the group

Strategies

- Flat Tree
- Binary Tree
- Binomial Tree
- Chain (pipeline)
- ...



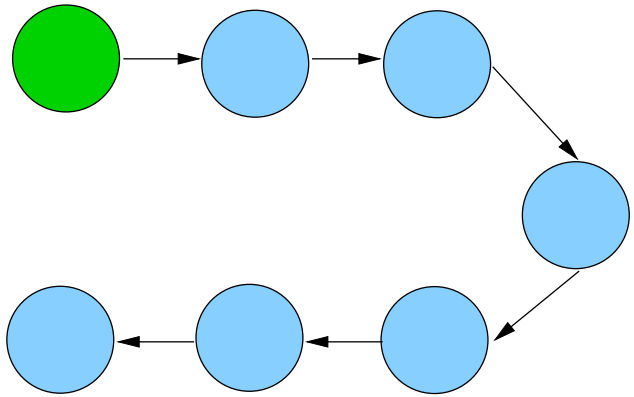
Example: modelling MPI_Bcast

Definition

- One process (root) send the same message to every process in the group

Strategies

- Flat Tree
- Binary Tree
- Binomial Tree
- Chain (pipeline)
- ...



MPI_Bcast Modelling on Homogeneous Clusters

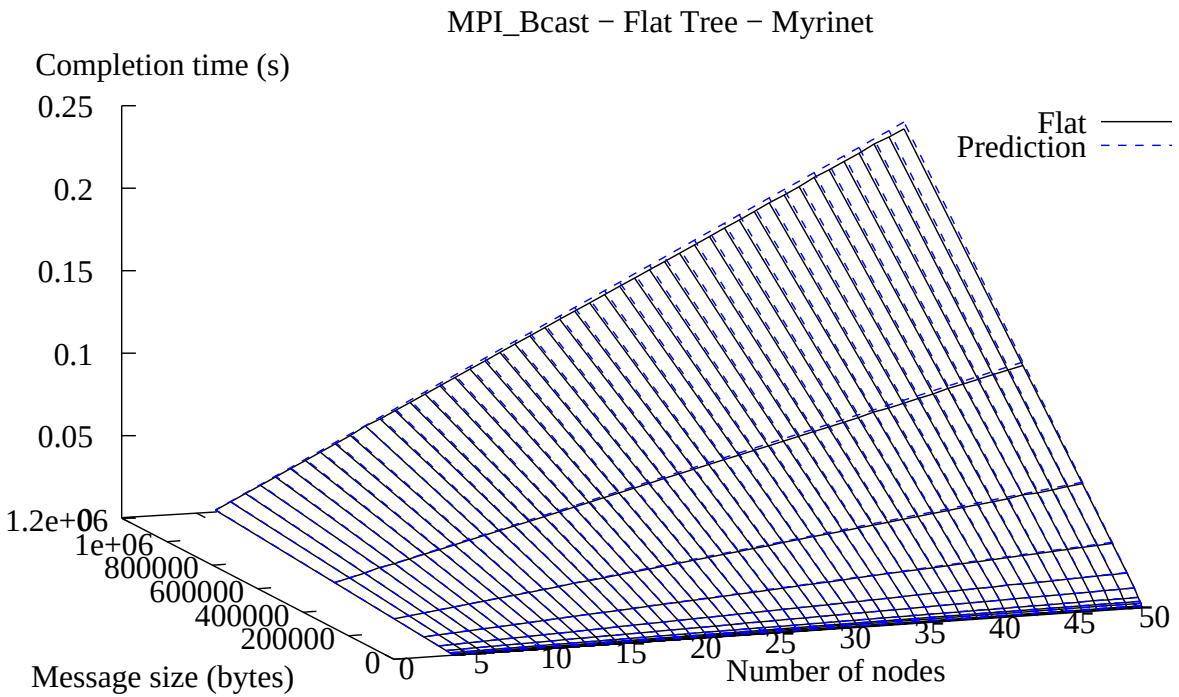
Implementation Strategy	Communication Model
Flat Tree	$(P-1) \times g(m) + L$
Flat Tree with Rendez-vous	$(P-1) \times g(m) + 2 \times g(1) + 3 \times L$
Segmented Flat Tree	$(P-1) \times (g(s) \times k) + L$
Binomial Tree	$\lfloor log_2 P \rfloor \times g(m) + \lceil log_2 P \rceil \times L$
Binomial Tree with Rendez-vous	$\lfloor log_2 P \rfloor \times g(m) + \lceil log_2 P \rceil \times (2 \times g(1) + 3 \times L)$
Segmented Binomial Tree	$\lfloor log_2 P \rfloor \times g(s) \times k + \lceil log_2 P \rceil \times L$
Binary Tree	$\leq \lceil log_2 P \rceil \times (2 \times g(m) + L)$
Chain	$(P-1) \times (g(m) + L)$
Chain with Rendez-vous	$(P-1) \times (g(m) + 2 \times g(1) + 3 \times L)$
Segmented Chain (*Pipeline*)	$(P-1) \times (g(s) + L) + (g(s) \times (k-1))$

MPI_Bcast Modelling on Homogeneous Clusters

Implementation Strategy	Communication Model
Flat Tree	$(P-1) \times g(m) + L$
Flat Tree with Rendez-vous	$(P-1) \times g(m) + 2 \times g(1) + 3 \times L$
Segmented Flat Tree	$(P-1) \times (g(s) \times k) + L$
Binomial Tree	$\lfloor log_2 P \rfloor \times g(m) + \lceil log_2 P \rceil \times L$
Binomial Tree with Rendez-vous	$\lfloor log_2 P \rfloor \times g(m) + \lceil log_2 P \rceil \times (2 \times g(1) + 3 \times L)$
Segmented Binomial Tree	$\lfloor log_2 P \rfloor \times g(s) \times k + \lceil log_2 P \rceil \times L$
Binary Tree	$\leq \lceil log_2 P \rceil \times (2 \times g(m) + L)$
Chain	$(P-1) \times (g(m) + L)$
Chain with Rendez-vous	$(P-1) \times (g(m) + 2 \times g(1) + 3 \times L)$
Segmented Chain (*Pipeline*)	$(P-1) \times (g(s) + L) + (g(s) \times (k-1))$

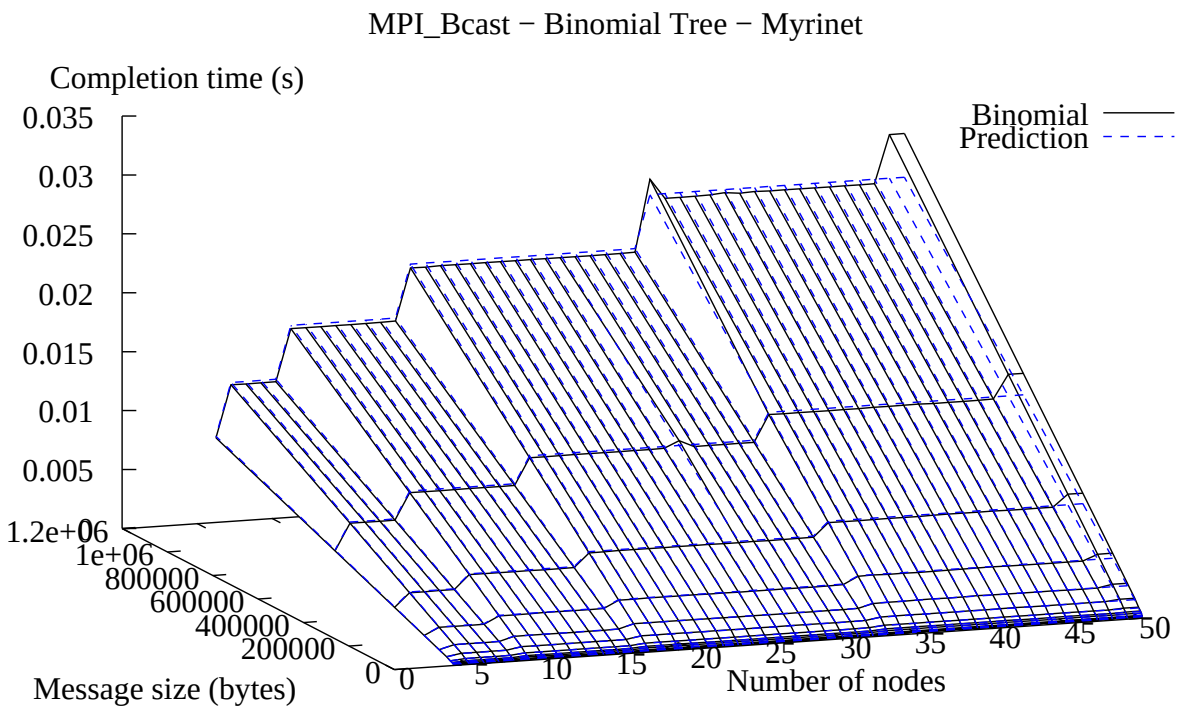
Flat Tree Broadcast

- The simplest one - $(P-1) \times g(m) + L$
- normally used with a few nodes (bad performance)
  - prediction error < 2%



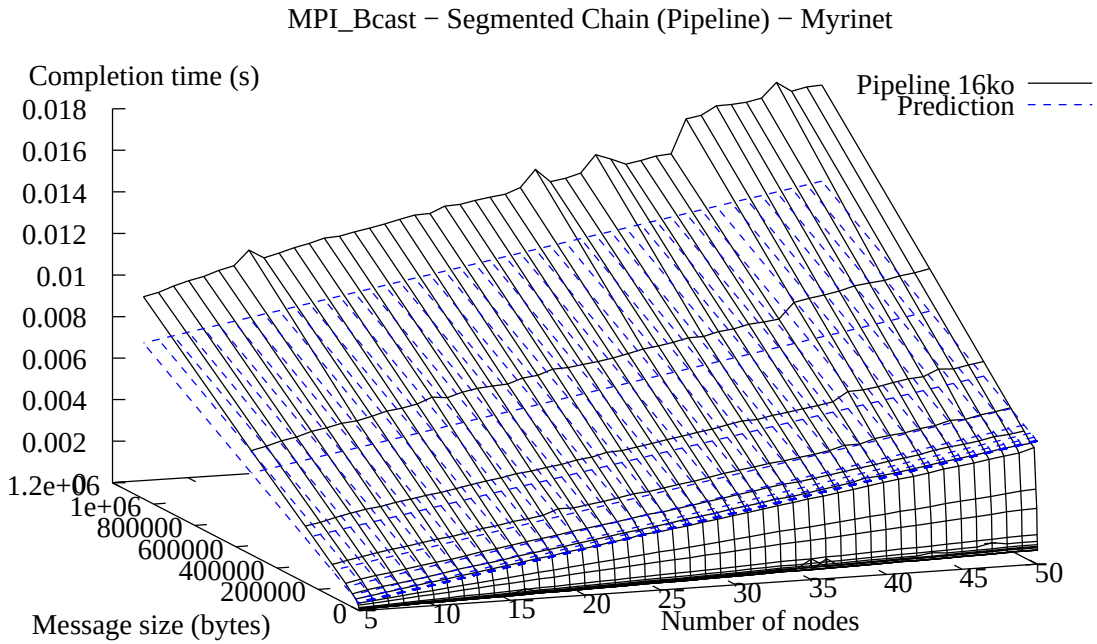
Binomial Tree Broadcast

- $\lfloor log_2 P \rfloor \times g(m) + \lceil log_2 P \rceil \times L$
  - prediction error < 5%

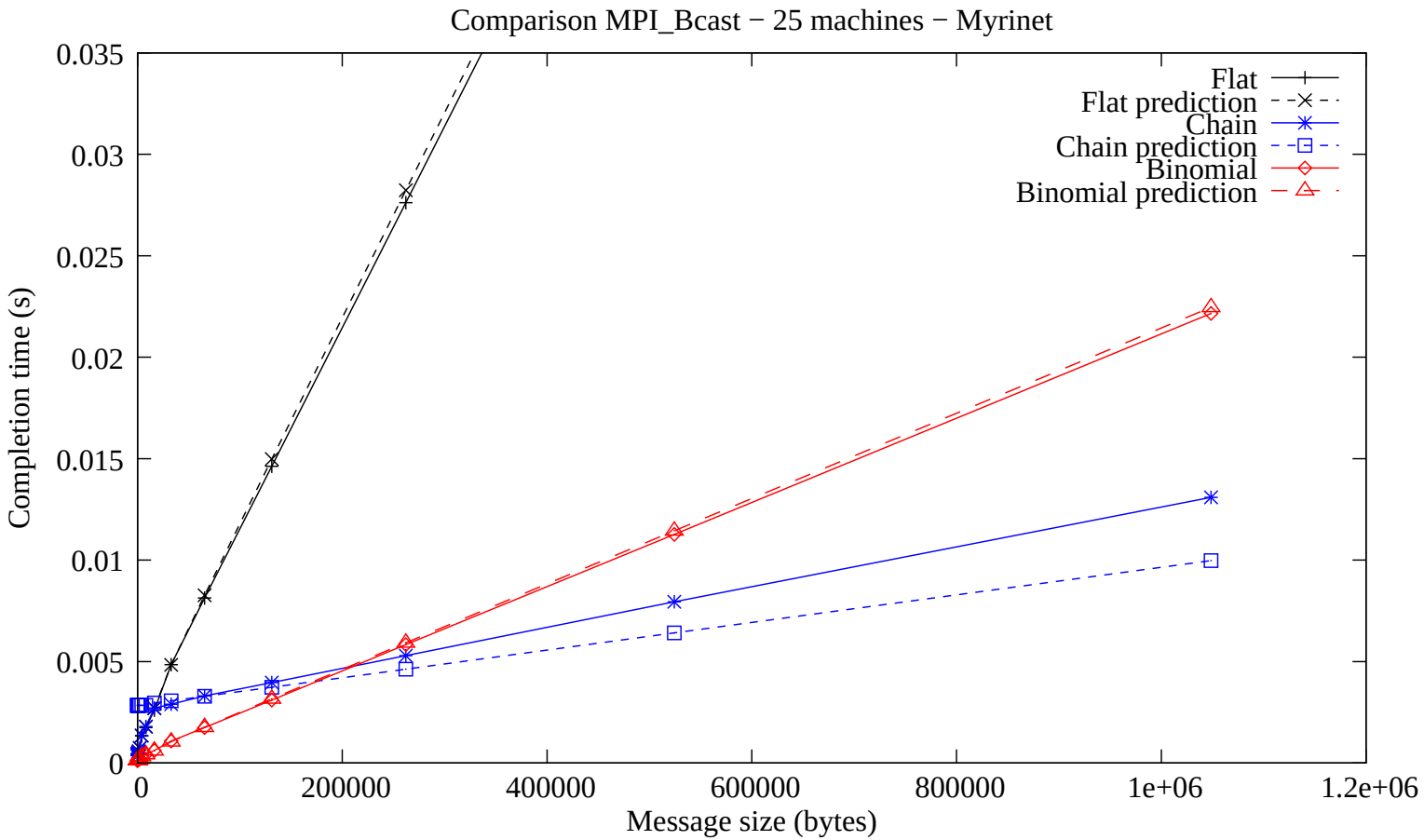


Segmented Chain Broadcast

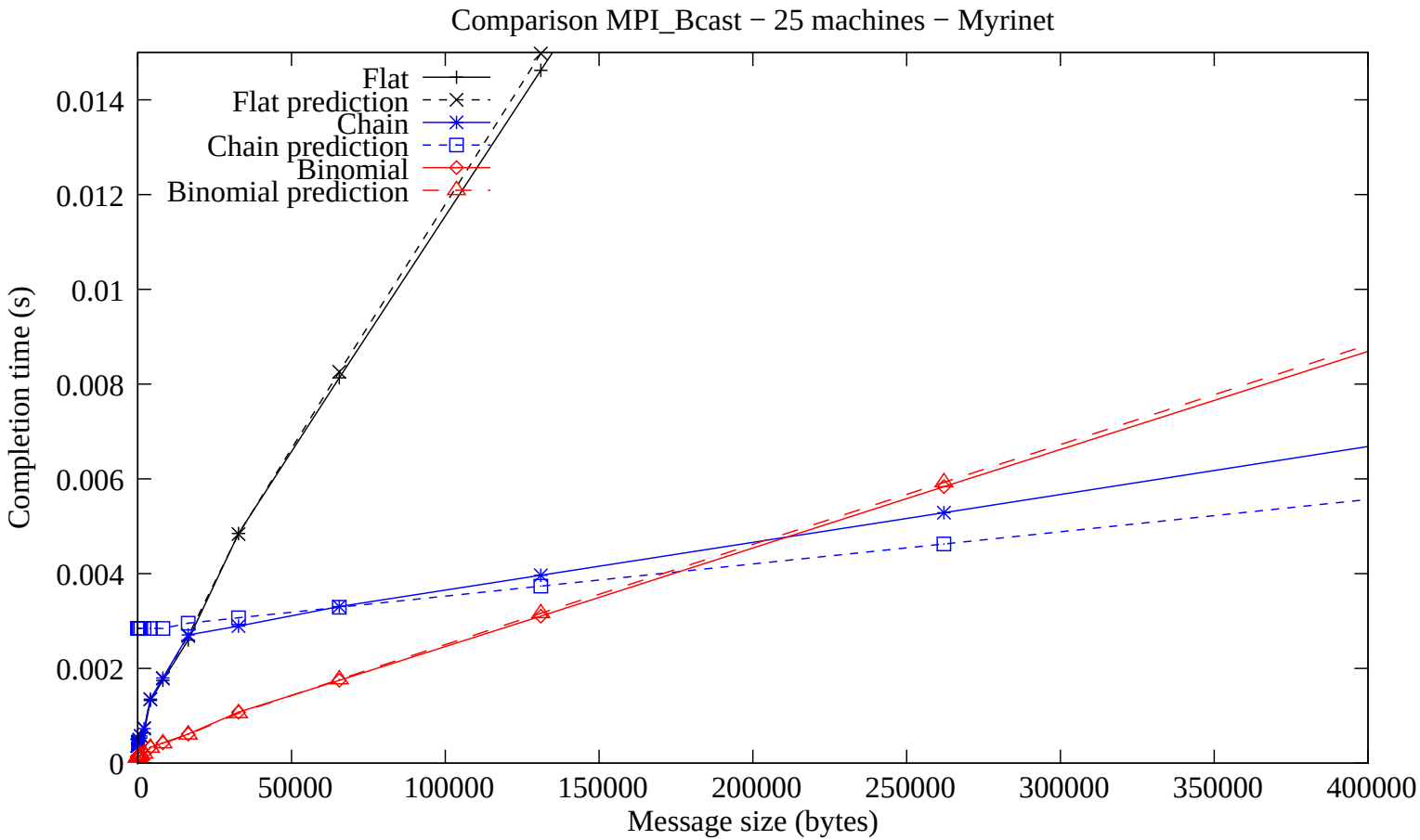
- $(P-1) \times (g(s)+L) + (g(s) \times (k-1))$
- Performance depends on the segment size
- Dependent on the performance of **all** nodes



Choosing the best strategy



Choosing the best strategy - small messages



Outline

Grid-Aware Collective Communication

Scheduling Communications in a Heterogeneous Environment

- exhaustive search
  - genetic algorithms (Vorakosit)
  - simulated annealing (Vadhiyar)

Grid-Aware Collective Communication

Scheduling Communications in a Heterogeneous Environment

- exhaustive search
  - genetic algorithms (Vorakosit)
  - simulated annealing (Vadhiyar)
- operation specific optimisations
  - pipelined broadcasts (Beaumont *et al.*)
  - balanced trees (Burger *et al.*).

Grid-Aware Collective Communication

Scheduling Communications in a Heterogeneous Environment

- exhaustive search
  - genetic algorithms (Vorakosit)
  - simulated annealing (Vadhiyar)
- operation specific optimisations
  - pipelined broadcasts (Beaumont *et al.*)
  - balanced trees (Burger *et al.*).
- optimisation heuristics
  - FEF and ECEF (Bhat)

Grid-Aware Collective Communication

Why to use an hierarchical scheduling

- reduces the search space

Grid-Aware Collective Communication

Why to use an hierarchical scheduling

- reduces the search space
- each cluster may use different strategies
  - binomial, chain, etc.

Grid-Aware Collective Communication

Why to use an hierarchical scheduling

- reduces the search space
- each cluster may use different strategies
  - binomial, chain, etc.
- this approach may be employed also with other communication patterns

Broadcast - Optimisation Heuristics

Fastest Edge First -FEF (Bhat)

- objective: select the sender that can reach a new receiver earlier
- strategy: find the edge with the minimum latency

$$\min_{i \in A, j \in B} L_{i,j}$$

Drawback

this strategy may overload a single sender

Broadcast - Optimisation Heuristics

Earliest Completing Edge First - ECEF (Bhat)

- objective: select the fastest available sender to reach a new receiver
  - strategy: take into account the *Ready Time* and the transfer time

$$\min_{i \in A, j \in B} (RT_i + g_{i,j}(m) + L_{i,j})$$

Weakness (?)

Can the receiver contribute to the broadcast?

Broadcast - Optimisation Heuristics

Earliest Completing Edge First with lookahead - ECEFLA (Bhat)

- objective: select the fastest available sender to reach a **good** receiver
  - a node that can contribute with message diffusion
- strategy: use a lookahead function to evaluate the usefulness of a receiver

$$\min_{i \in A, j \in B} (RT_i + g_{i,j}(m) + L_{i,j} + F_j) \ ;$$

Broadcast - Optimisation Heuristics

Earliest Completing Edge First with lookahead - ECEFLA (Bhat)

- objective: select the fastest available sender to reach a **good** receiver
  - a node that can contribute with message diffusion
- strategy: use a lookahead function to evaluate the usefulness of a receiver

$$\min_{i \in A, j \in B} (RT_i + g_{i,j}(m) + L_{i,j} + F_j) \;\; ; \;\; F_j = \min_{P_k \in B} (g_{j,k}(m) + L_{j,k})$$

Broadcast - Optimisation Heuristics

Common characteristics of these heuristics

- Give priority to fast links
  - maximise the number of potential senders

Question:

Can a previous knowledge on intra-cluster communications improve the efficiency of these heuristics?

- T_k - communication time inside a cluster

Specific Heuristics

ECEFLA-*t*

- simple extension of the ECEFLA heuristic
- objective: select the fastest available sender to reach a **good** receiver
  - a cluster contacted by this node may finish in the smallest time
  - quickly reduces the number of clusters to contact

$$\min_{i \in A, j \in B} (RT_i + g_{i,j}(m) + L_{i,j} + F_j) \;\; ; \;\; F_j = \min_{P_k \in B} (g_{j,k}(m) + L_{j,k} + T_k)$$

Drawbacks in a Grid System

All these strategies always try to contact first the fastest clusters/nodes

- Communications to distant/slow clusters are delayed
- This extra delay may augment the makespan

Balance communication:

- Give some priority to slow clusters
- Still keep trying to reach the largest number of nodes
  - maximise the number of data sources

Specific Heuristics

ECEFLA-T - tries to balance the scheduling

- objective: select a receiver whose cost to contact the slowest cluster is still reduced
  - sender is the fastest one that can reach the slowest cluster
- strategy: the lookahead function maximises the search

$$\min_{i \in A, j \in B} (RT_i + g_{i,j}(m) + L_{i,j} + F_j) \;\; ; \;\; F_j = \max_{P_k \in B} (g_{j,k}(m) + L_{j,k} + T_k)$$

Drawback

slow clusters will be contacted only after no fast cluster remains

Specific Heuristics

Bottom-Up

- gives priority to slow clusters
- objective: prevent a supplementary delay for the slow clusters
- strategy: search for the slowest cluster still not contacted

$$\max_{P_j \in B} \left(\min_{P_i \in A} \left(g_{i,j}(m) + L_{i,j} + T_j \right) \right)$$

Drawback

does not improve the number of data sources

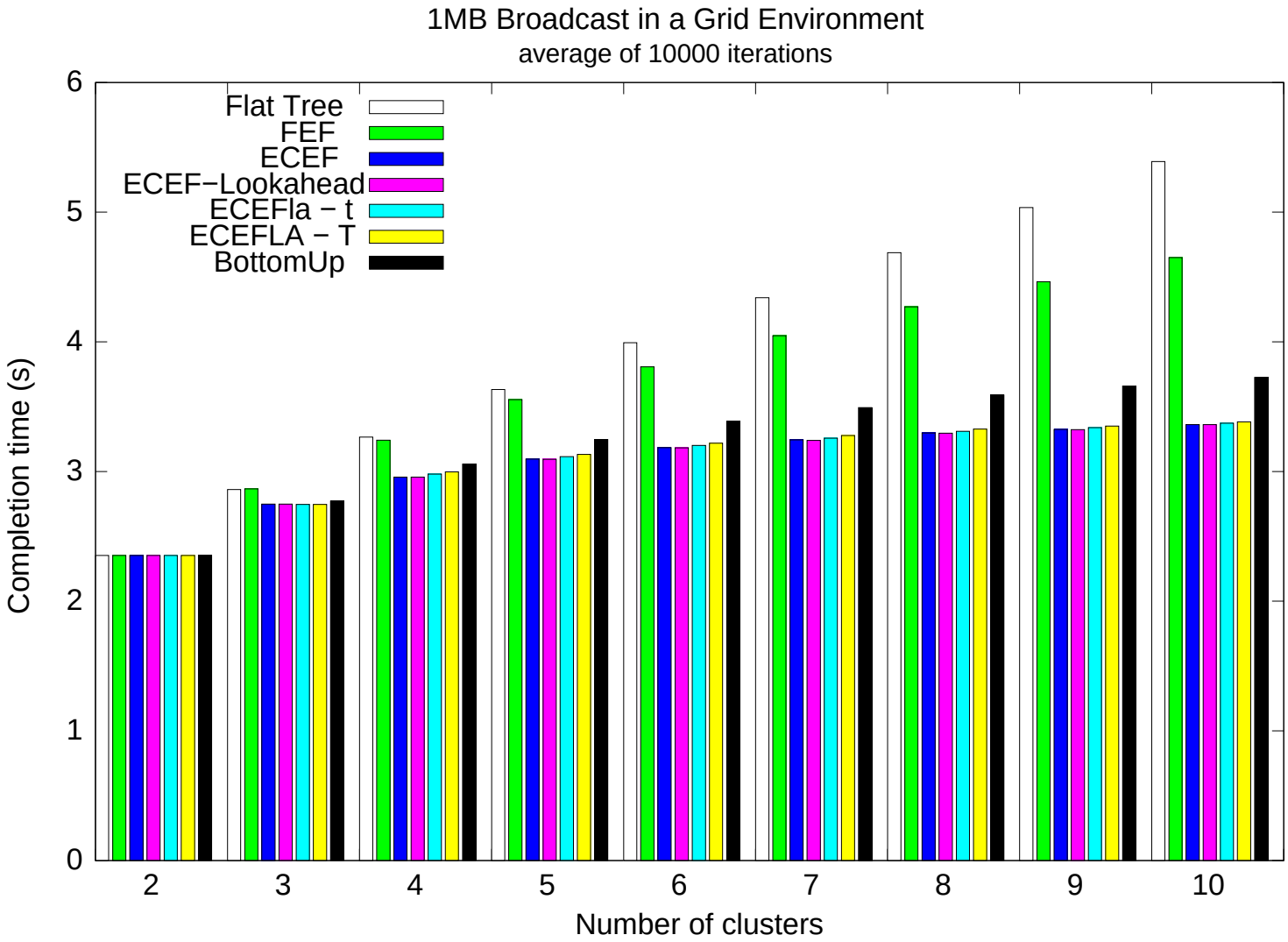
Comparing Strategies

Simulations

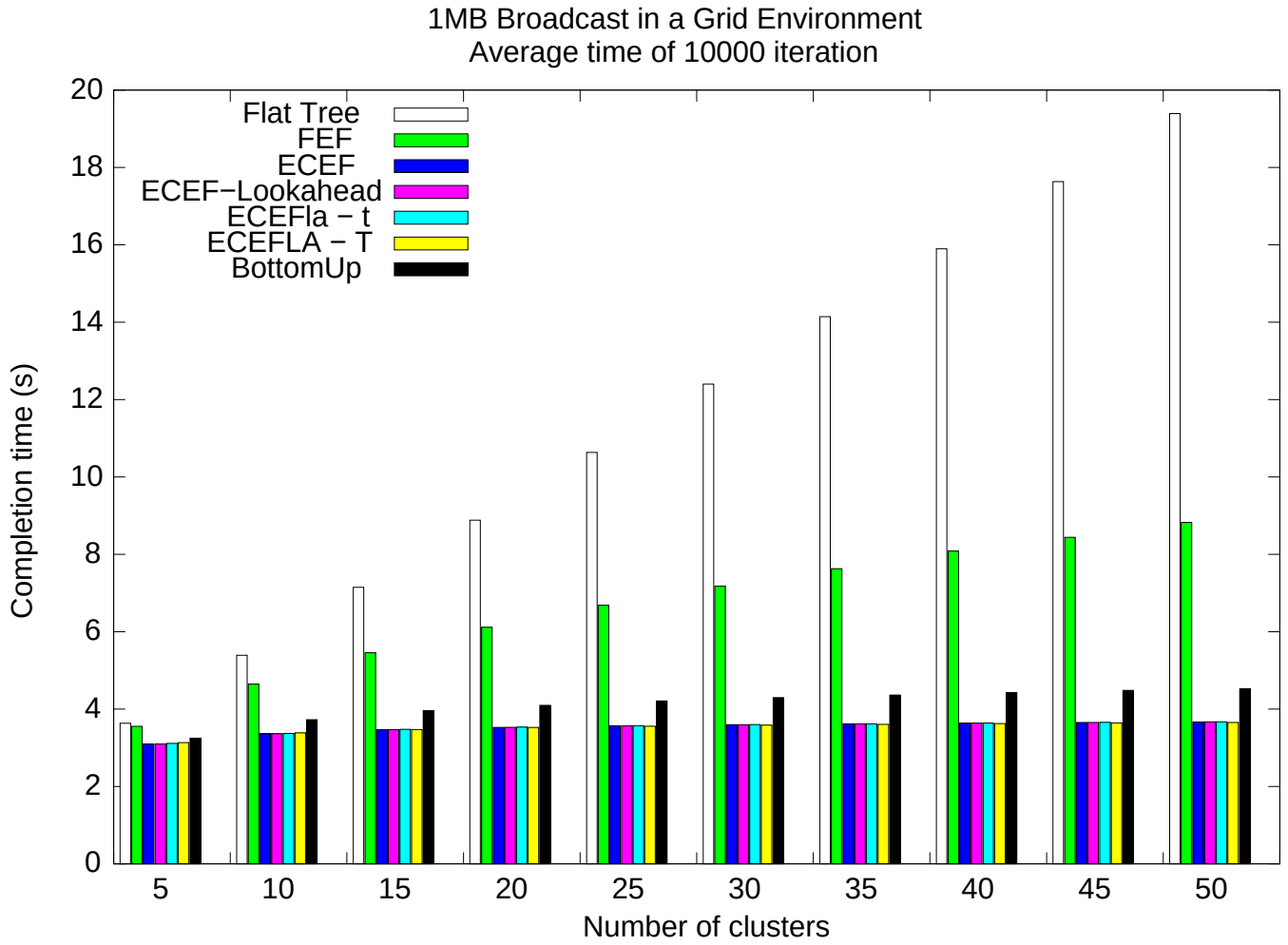
- Use of simulations to obtain the average performance of each strategy
  - average of 10000 runs
- Random values between:

	minimum⇔maximum	
$gap_{i,j}$	0.10 s⇔0.60 s	IDPOT-icluster2⇔IDPOT-GdX
$latency_{i,j}$	0.001 s⇔0.015 s	IDPOT-icluster2⇔GdX-Rennes
T_i	0.02 s⇔3 s	1 MB Myrinet⇔1 MB Fast Ethernet

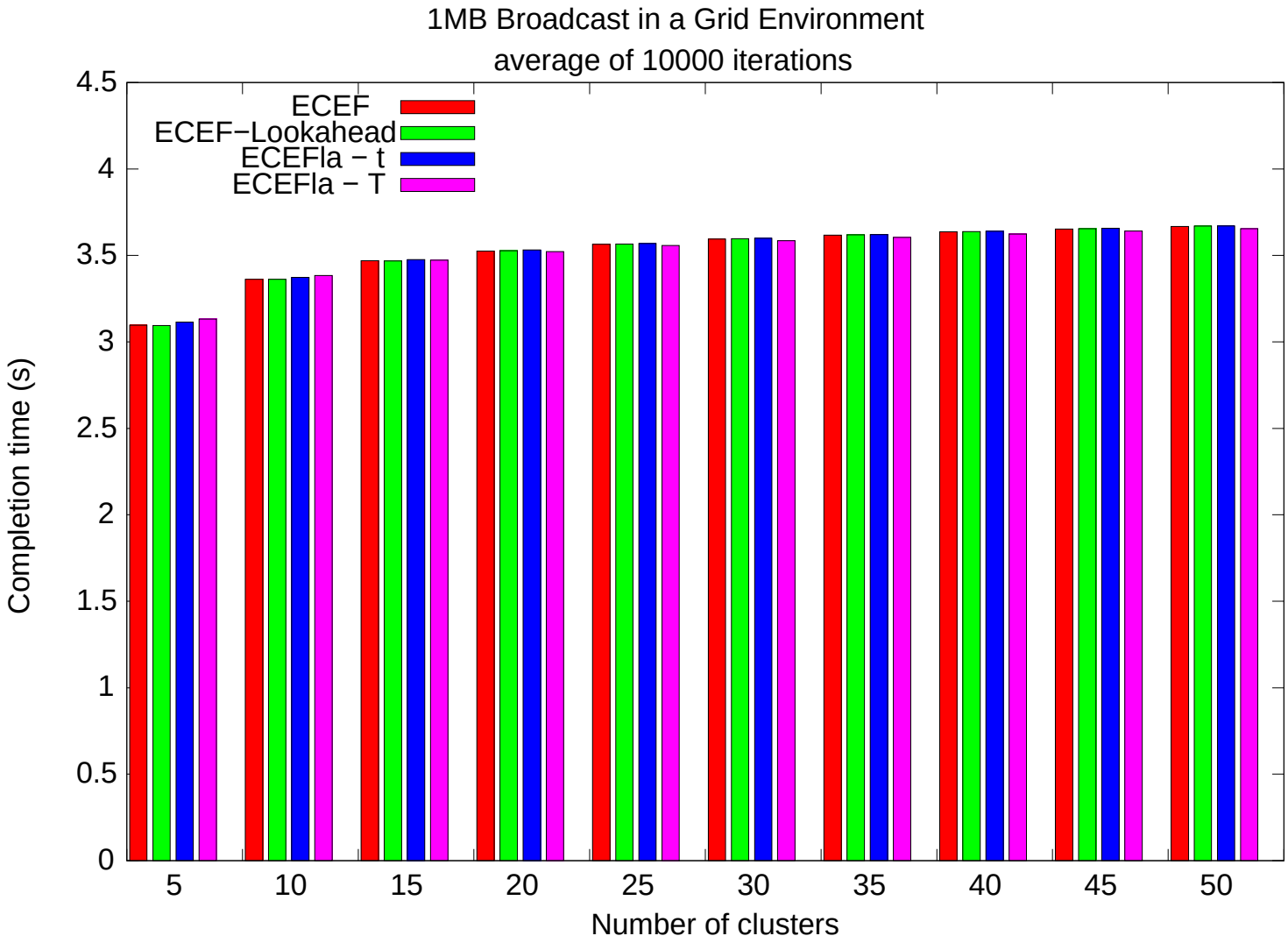
Comparing Strategies



A Large Scale Grid

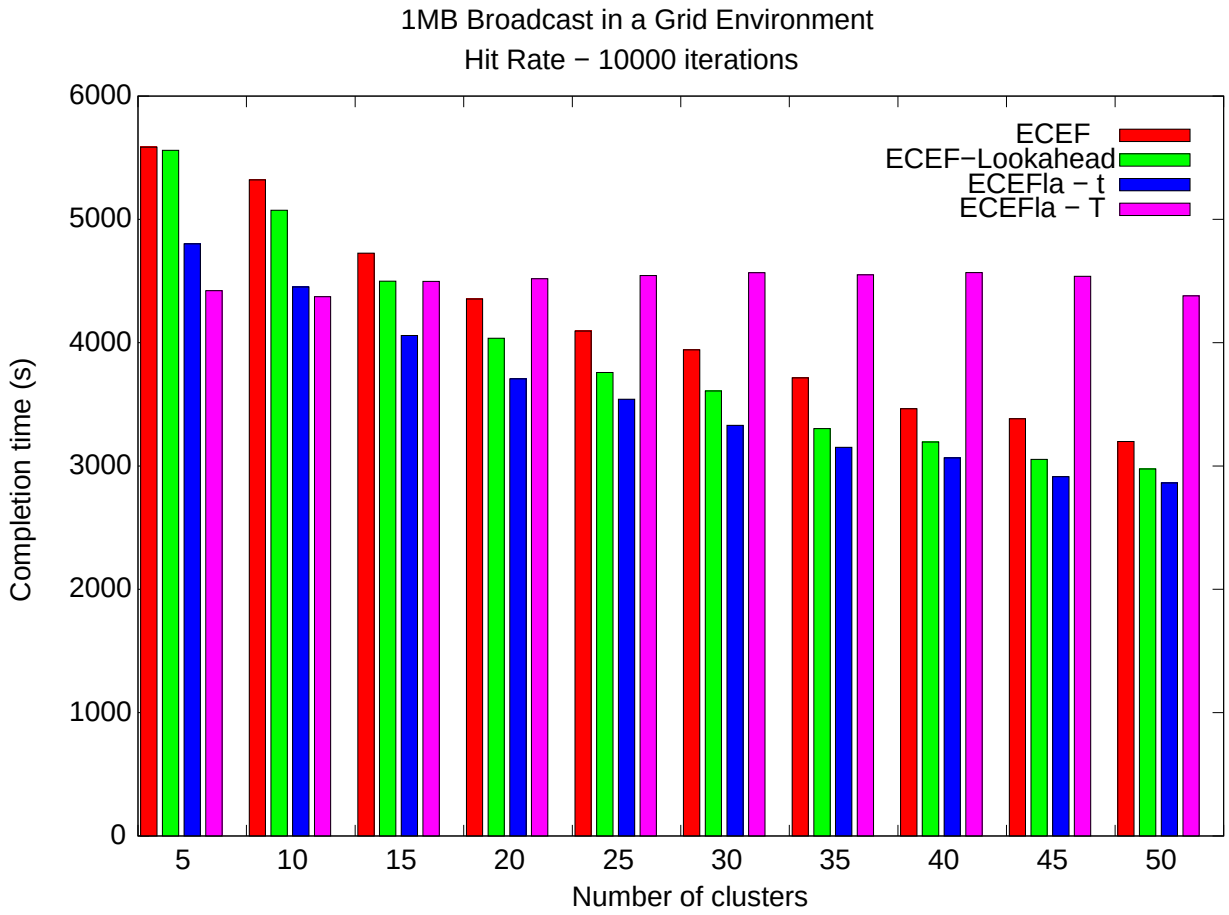


A Close Look


1MB Broadcast in a Grid Environment
average of 10000 iterations
ECEF
ECEF−Lookahead
ECEFla − t
ECEFla − T
Completion time (s)
Number of clusters
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
10
15
20
25
30
35
40
45
50

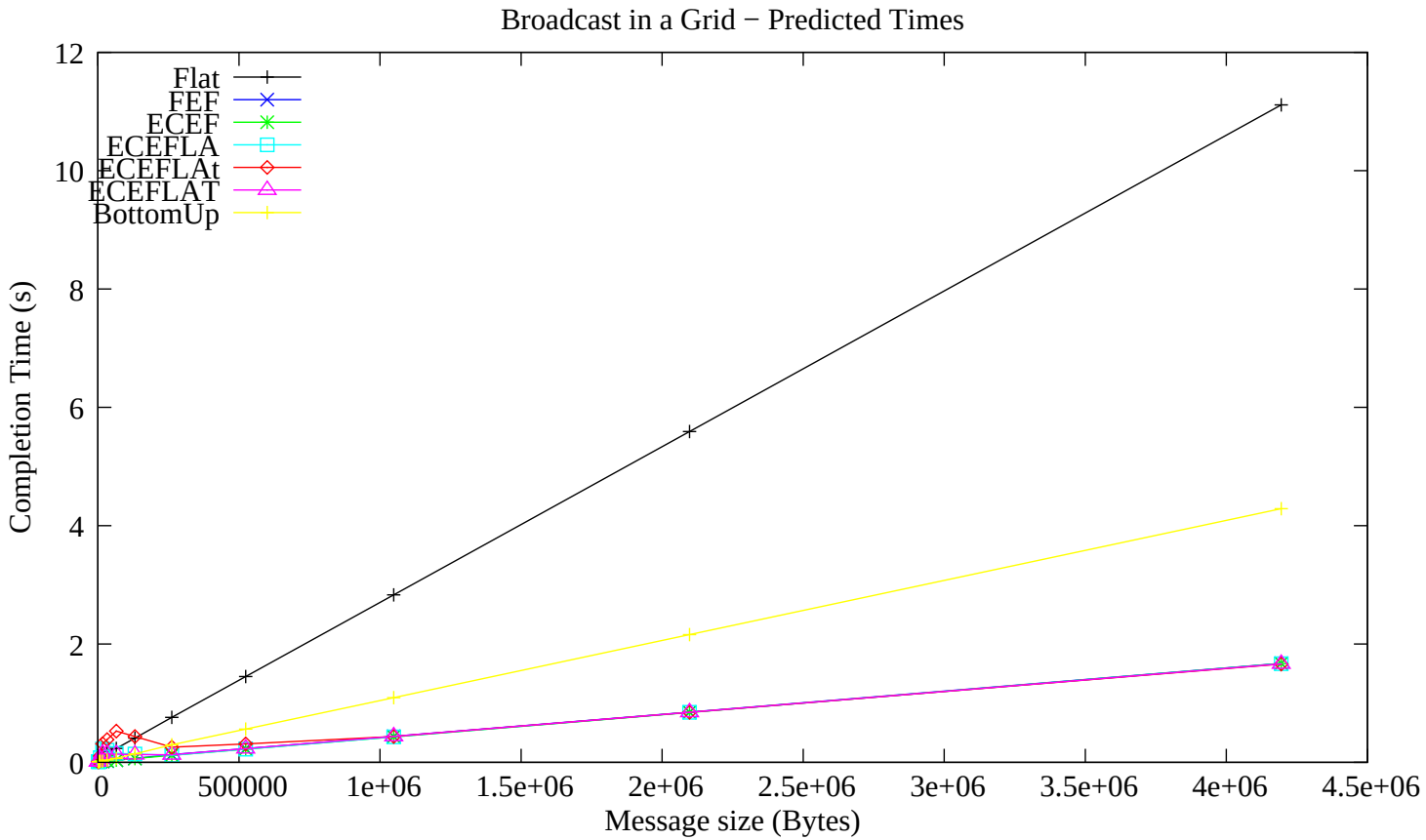
Hit Rate

- A different metric to evaluate the heuristics

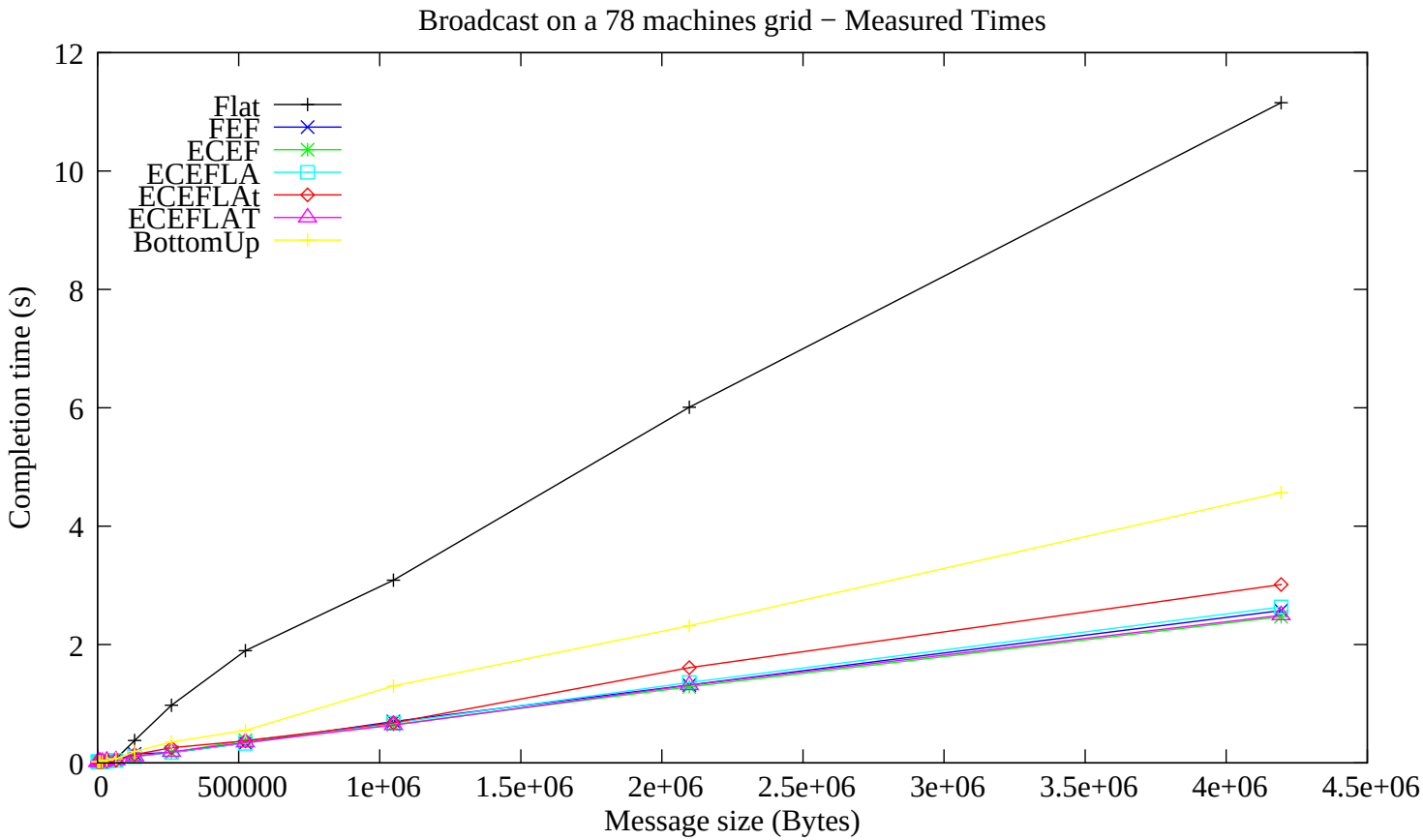


Experimental validation

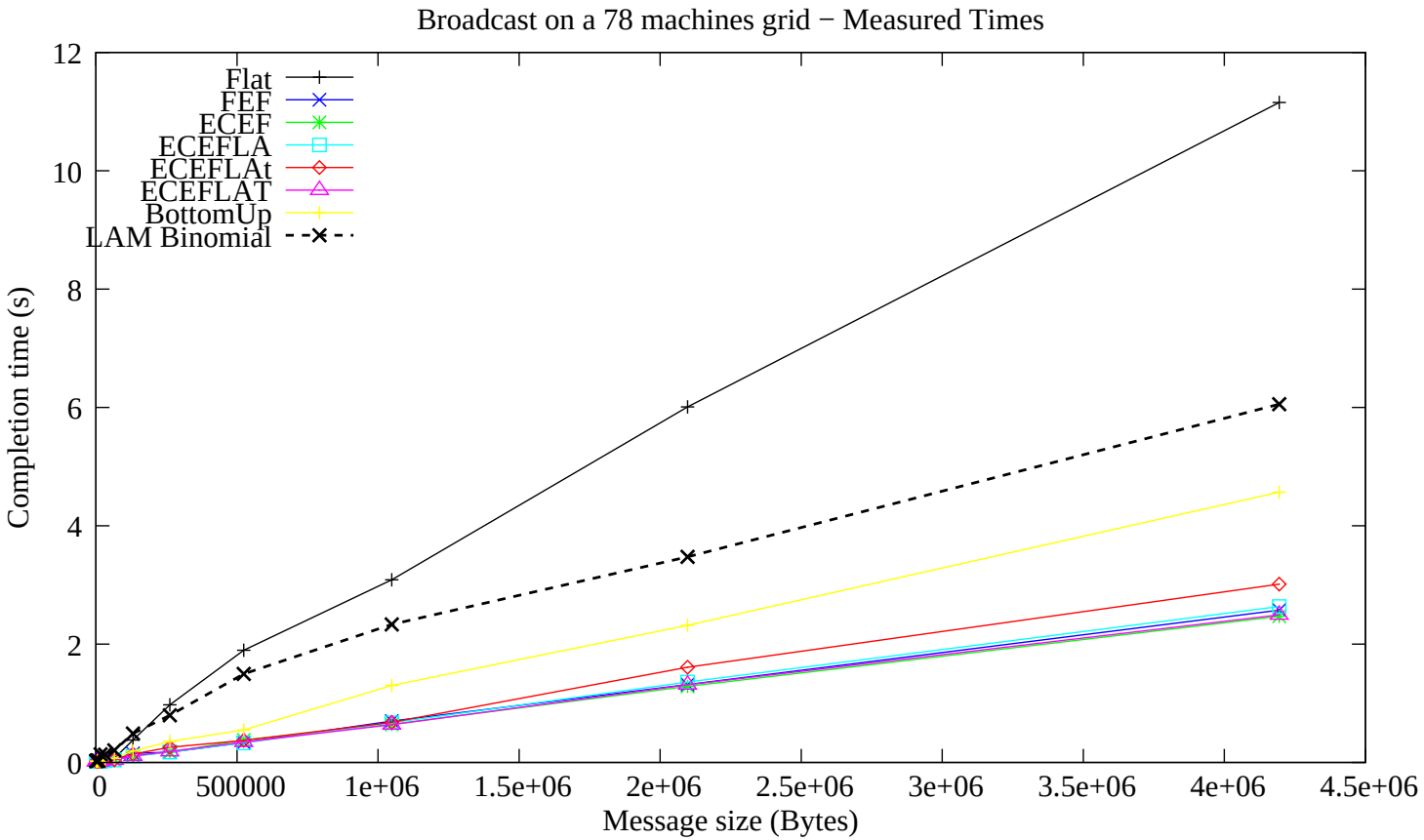
88 machines, 6 homogeneous clusters (3×IDPOT,2×GdX, Toulouse)



Experimental Validation



Experimental Validation



Conclusions

- Scheduling communications on a grid environment
  - Hierarchical communication reduces the optimisation complexity
  - Multi-layered communication with hybrid algorithms

Conclusions

- Scheduling communications on a grid environment
  - Hierarchical communication reduces the optimisation complexity
  - Multi-layered communication with hybrid algorithms
    - efficient ”well known” intra-cluster strategies

Conclusions

- Scheduling communications on a grid environment
  - Hierarchical communication reduces the optimisation complexity
  - Multi-layered communication with hybrid algorithms
    - efficient ”well known” intra-cluster strategies
    - dynamically scheduled inter-cluster communications

Conclusions

- Scheduling communications on a grid environment
  - Hierarchical communication reduces the optimisation complexity
  - Multi-layered communication with hybrid algorithms
    - efficient ”well known” intra-cluster strategies
    - dynamically scheduled inter-cluster communications
- Importance of Topology Discovery
  - Helps to better describe the real network
  - Prevents mistakes induced by manual configuration
  - Simplify further optimisation tasks

Future Works

- Extend our experiments
  - More experiments on a grid environment
  - Compare with other heuristics and optimisation techniques
  - Evaluate the impact on the performance of real applications

Future Works

- Extend our experiments
  - More experiments on a grid environment
  - Compare with other heuristics and optimisation techniques
  - Evaluate the impact on the performance of real applications
- Study deeply other communication patterns
  - One-to-many personalised, many-to-one, many-to-many, ...

Future Works

- Extend our experiments
  - More experiments on a grid environment
  - Compare with other heuristics and optimisation techniques
  - Evaluate the impact on the performance of real applications
- Study deeply other communication patterns
  - One-to-many personalised, many-to-one, many-to-many, ...
- Apply the optimisation techniques with other operations
  - Reduce, Gather, Barrier, etc.

Questions ?

Thank you!