Clustering with openMosix

M.Michels & W.Borremans

February 2005

Presentation contents

What is openMosix?

- Our experiment
  - Infrastructure
  - Test methods
  - Test results
- Issues running openMosix
- Security precautions in openMosix
- When to use openMosix?
- Future work of this project
- (Time for questions and discussion)

What is openMosix?

- Linux kernel extension for single-image clustering
- Turns multiple Linux hosts into one large virtual SMP
- Adaptive load balancing techniques
- Implementable on every Linux flavor by applying a kernel patch
- No specific libraries needed to run applications (MPI & PVM)

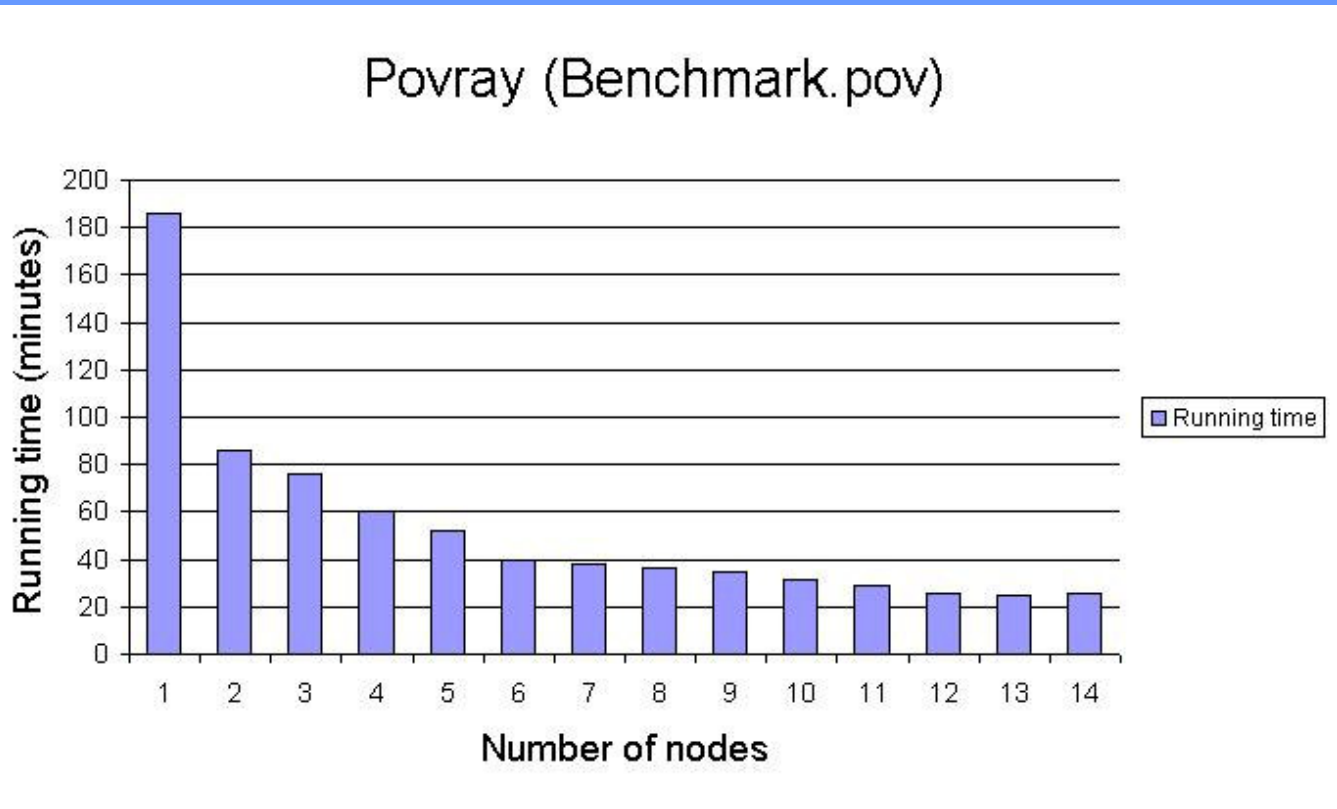
Our experiment

- Infrastructure
  - 14x Intel PIII 1Ghz, 256MB,3COM 3C905x (including server node)
    - Each node had a swap partition of 512MB
  - 100Mbit Ethernet switched network
- Focus on
  - Performance
  - Reliability
  - Network load

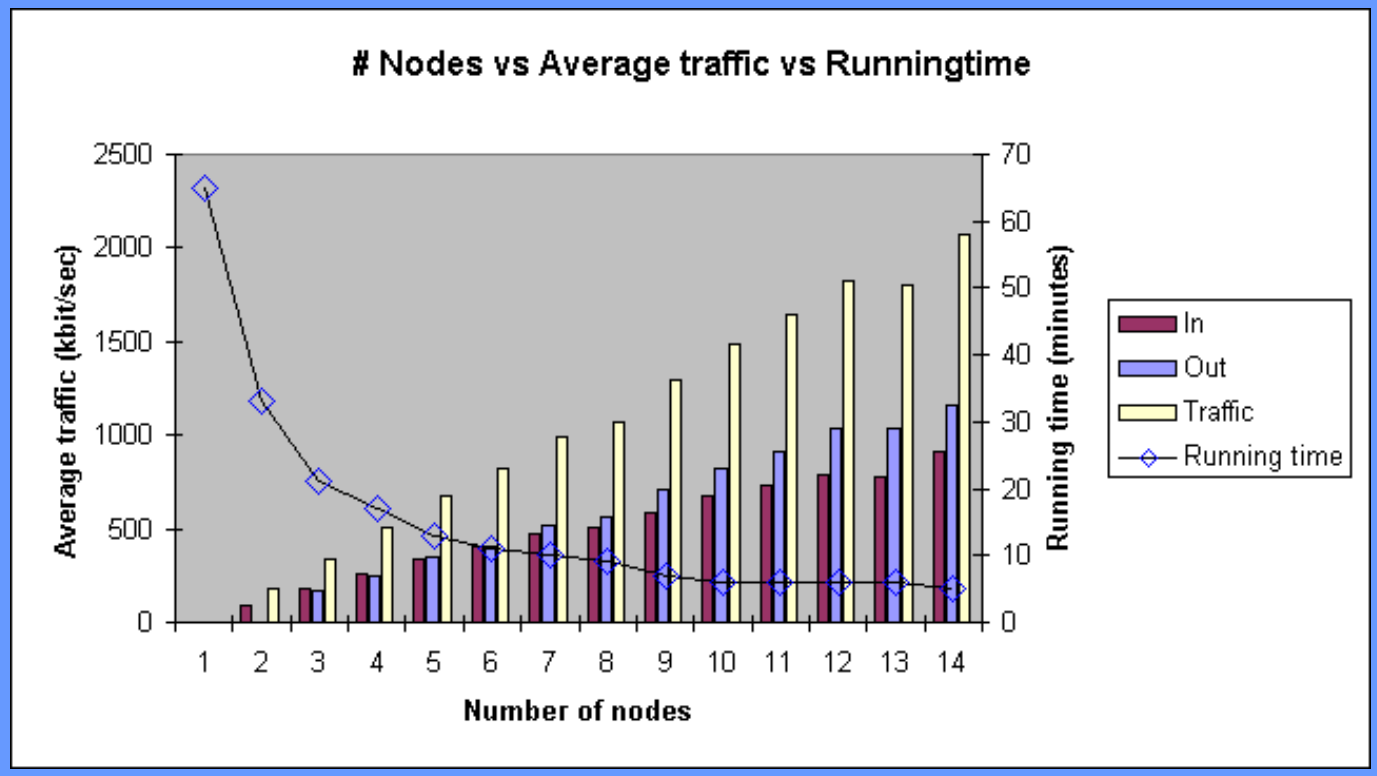
Tested programs

- Povray (image calculation)
  - Generate images from instructions file
    - Divide instructions in sub-jobs and distribute them over the cluster
- Encryption
  - Generate 4000 RSA public/private keys
    - Each node generates a specific number of keys
- Compiling
  - Make a kernel using all the nodes (Make –j28)
    - Distribute parts of the make process over the cluster

Test results - Povray



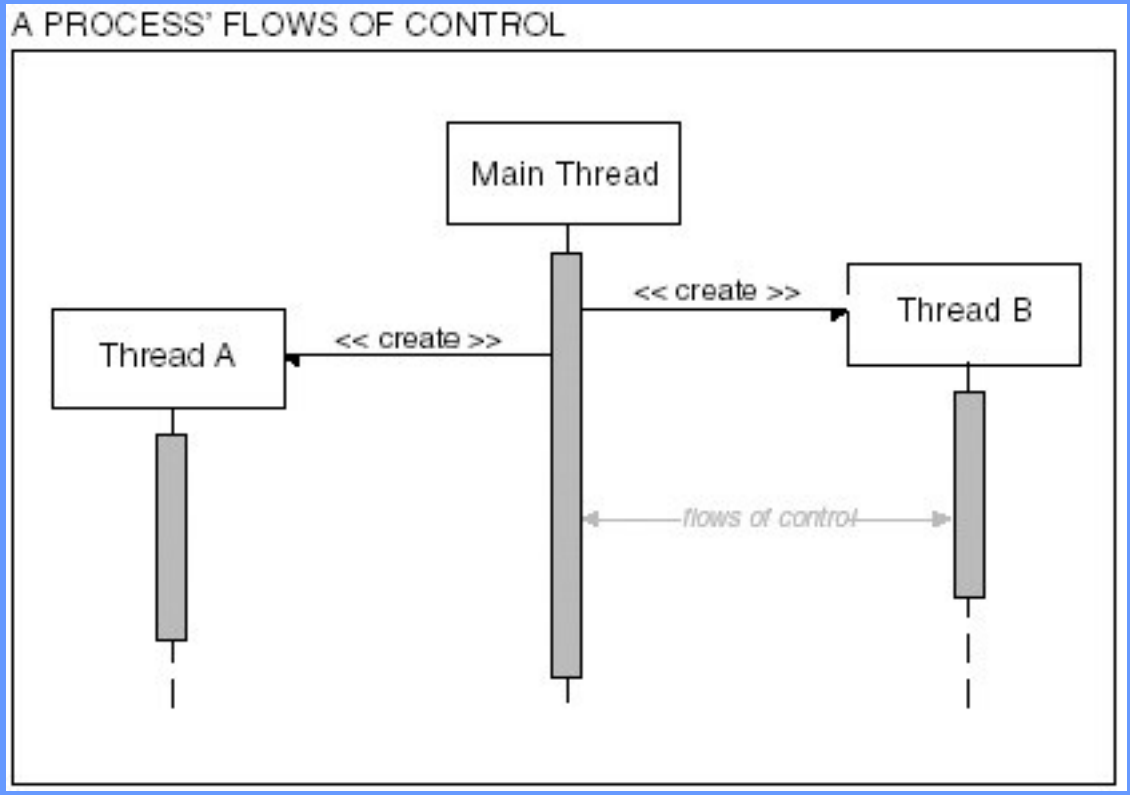
Test results - Encryption



Test results - compiling

- Unreliable on openMosix
- During compiling make ‘looses’ process → dependencies fail
- Too much requests in one interrupt

Threads vs Processes



Issues running openMosix (1)

- openMosix **cannot** distribute threads over the cluster
- Distribution of processes lasts relatively long
- After six nodes performance increase drops

Issues running openMosix (2)

- Processes ‘hop’ over the cluster, no performance increase
- Jobs that fail cannot be reassigned to the cluster
- If a node fails, server node could crash

Security precautions in openMosix

- None
  - No authentication of nodes
  - Everyone can inject a job in the cluster
  - Unencrypted transportation of data
- Only implementable in non-public environments

When to use openMosix?

- Applications which create processes
- Processes without special libraries
- Use of applications which store their results during running time
- Implementable on old / cheap hardware giving you a ‘supercomputer’

Future work of this project

- Rewriting openMosix to add support for threaded applications
- Develop a method to find the optimal number of processes per node on a cluster

Questions / discussion

- Thanks for your attention
- Project proposal, report and presentation at:
  - http://www.os3.nl/~wborremans/rp1.html