The 2007 Workshop on Optimization of Parallel Routines and Applications
Universidad de Murcia, Spain

A Performance Model of MPI Collective Communications for Parallel Computing on Computational Clusters

Alexey Lastovetsky, Maureen O’Flynn, Vladimir Rychkov
UCD School of Computer Science and Informatics
Belfield, Dublin 4, Ireland
alexey.lastovetsky@ucd.ie

06 June 2007

Heterogeneous Computing Laboratory
School of Computer Science and Informatics
University College Dublin

Motivation

- **UCD Heterogeneous Computing Laboratory:** Research and development in high performance heterogeneous computing
  - Algorithms: parallel and distributed
  - Programming tools: mpC, HeteroMPI, SmartGridSolve
- **Approach:** Model-based
  - The programming tools build, maintain and use for optimization the performance model of the executing heterogeneous platform
  - => Accuracy and efficiency of the model are critical
- **HeteroMPI:** An extension of MPI for high performance computing on heterogeneous clusters
  - Accurate and efficient performance model of heterogeneous processors
  - Communication model
    - Very basic
    - Cannot be used for optimization of communication operations

Background

- **Goal:** Analytical model for prediction of the execution time of MPI communication operations on heterogeneous clusters based on a switched network (the most common parallel platform)
- **Approach**
  - Start with a performance model of a ***single point-to-point communication***
  - Use the model to construct models for collective communications
  - Results in linear models for collectives
- **Validation**
  - Works for ***simultaneous independent point-to-point*** communications
  - ***One-to-many*** (scatter-like) communications
    - Problem: A step-wise increase of the execution time for large messages
  - ***Many-to-one*** (gather-like) communication
    - Problem: Significant and non-deterministic escalations of the execution time of for medium-sized messages

Performance model for point-to-point communication

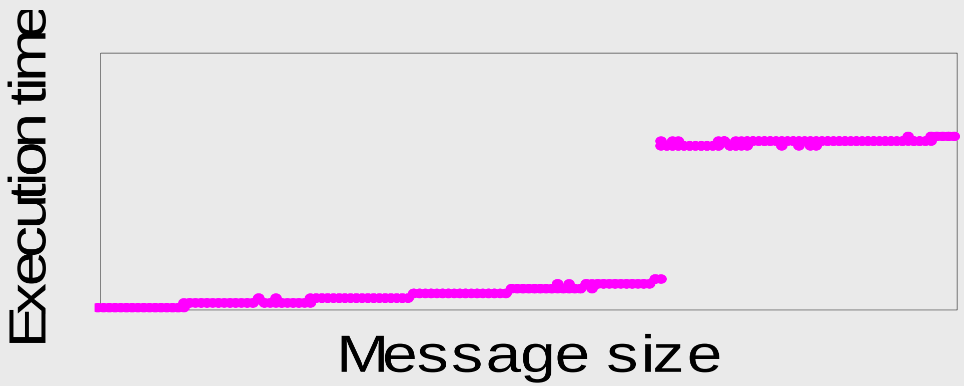
- **Sending a message from processor i to processor j:**

$$T_{ij} = C_i + t_i M + C_j + t_j M + M / \beta_{ij}$$

  - T_{ij} - *execution time*
  - M - *message size*
  - C_i, C_j - *fixed delays*
  - t_i , t_j - *variable delays*
  - β_{ij} - *transmission rate*

Performance model for one-to-many communication

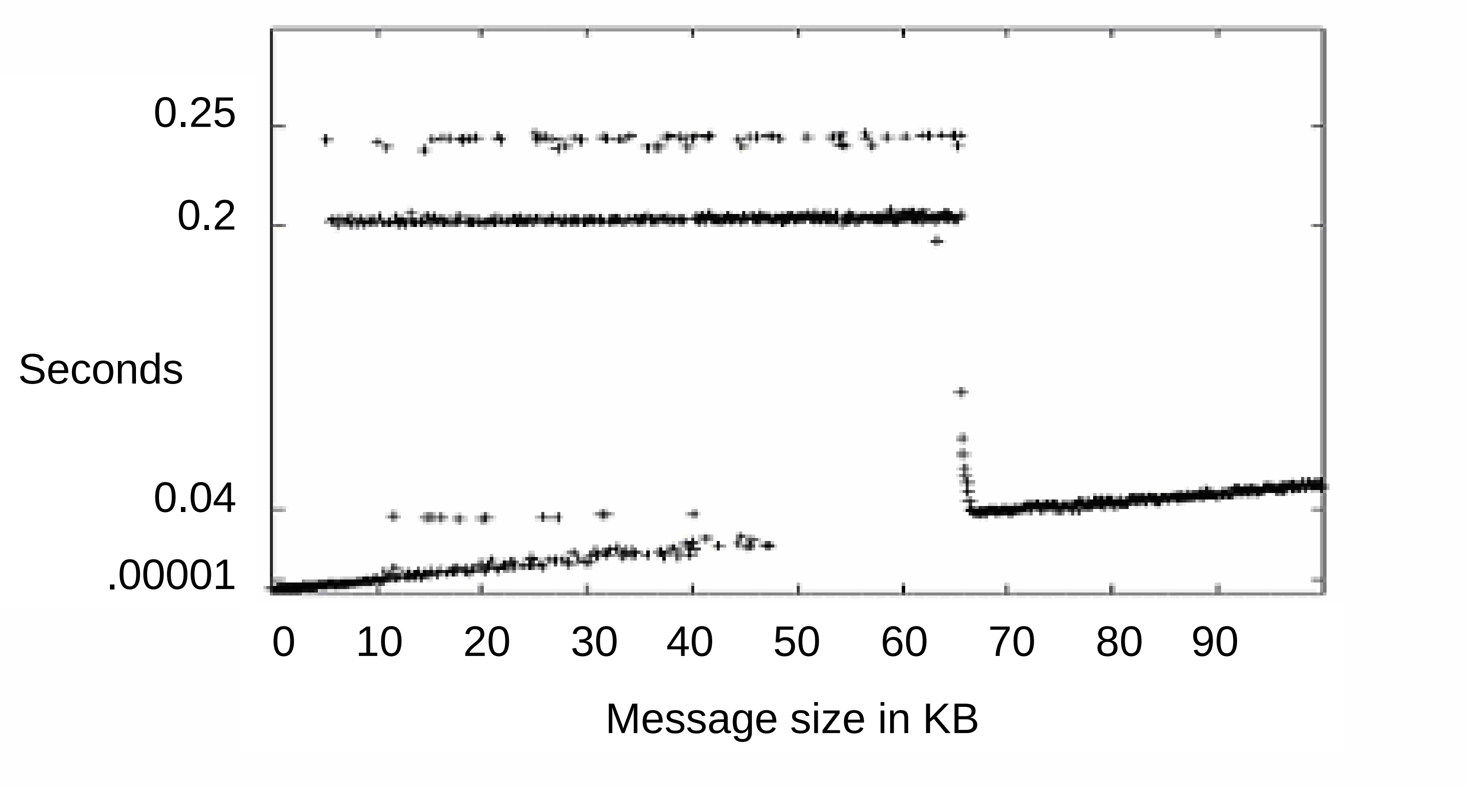
- **One-to-many:**



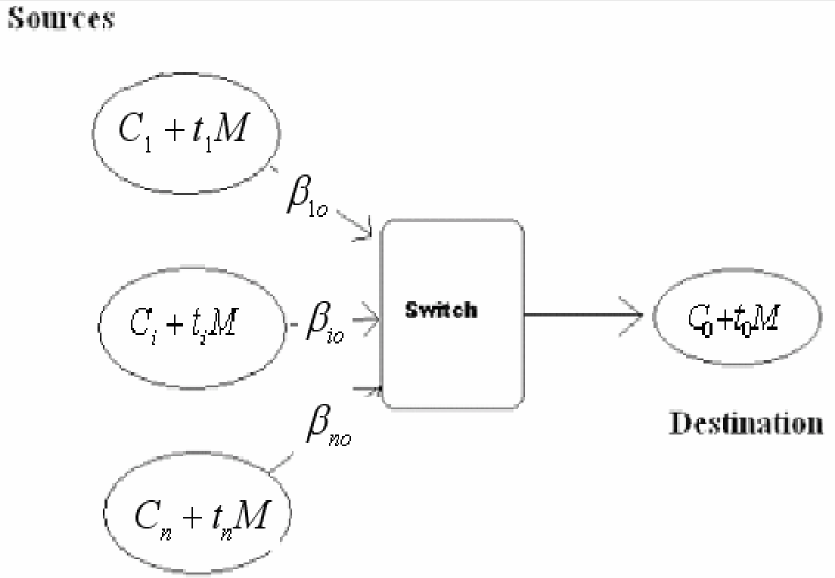
$$C_0 + t_0 \times n \times M + \max_{1 \le i \le n}\left\{C_i + t_i M + M / \beta_{0i}\right\}, M \le S$$

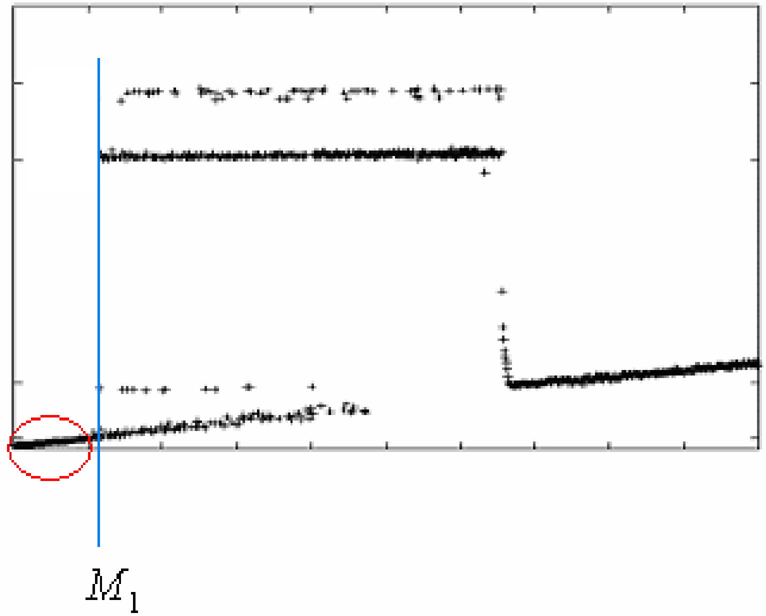
$$C_0 + t_0 \times n \times M + \sum_{i=1}^{n}\left(C_i + t_i M + M / \beta_{0i}\right), M > S$$

Many-to-one collective communications: non-linear and non-deterministic escalations



Many-to-one model for small messages





$$T = n(C_0 + t_o M) + \max_{1<i\leq n}\{C_i + t_i M + M / \beta_{io}\} + \kappa_1 M$$

Parameters of many-to-one model for medium-sized messages

$M_1 = M_1(n)$

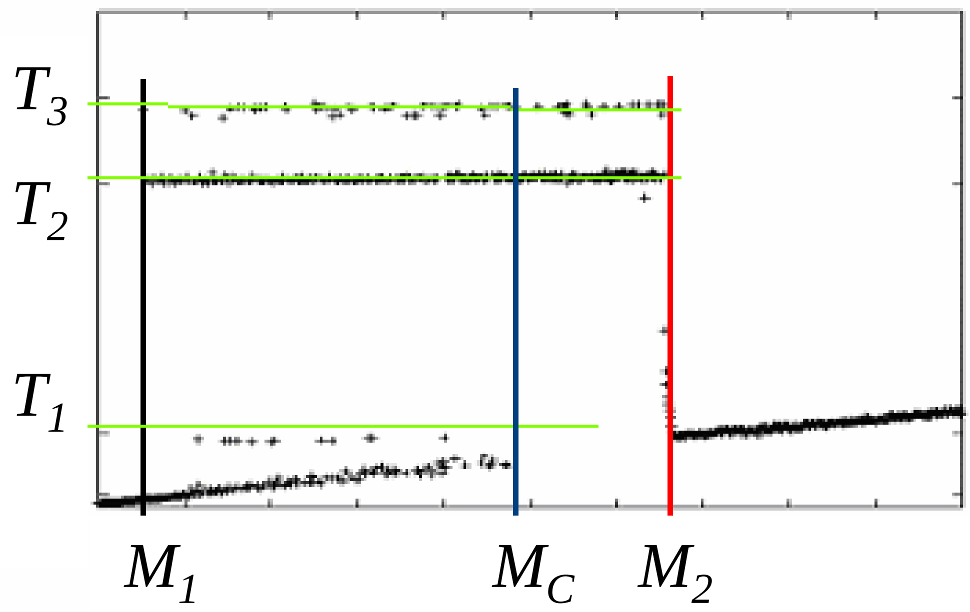
- escalations begin

$M_2 = const$

- escalations stop

$M_C = M_C(n)$

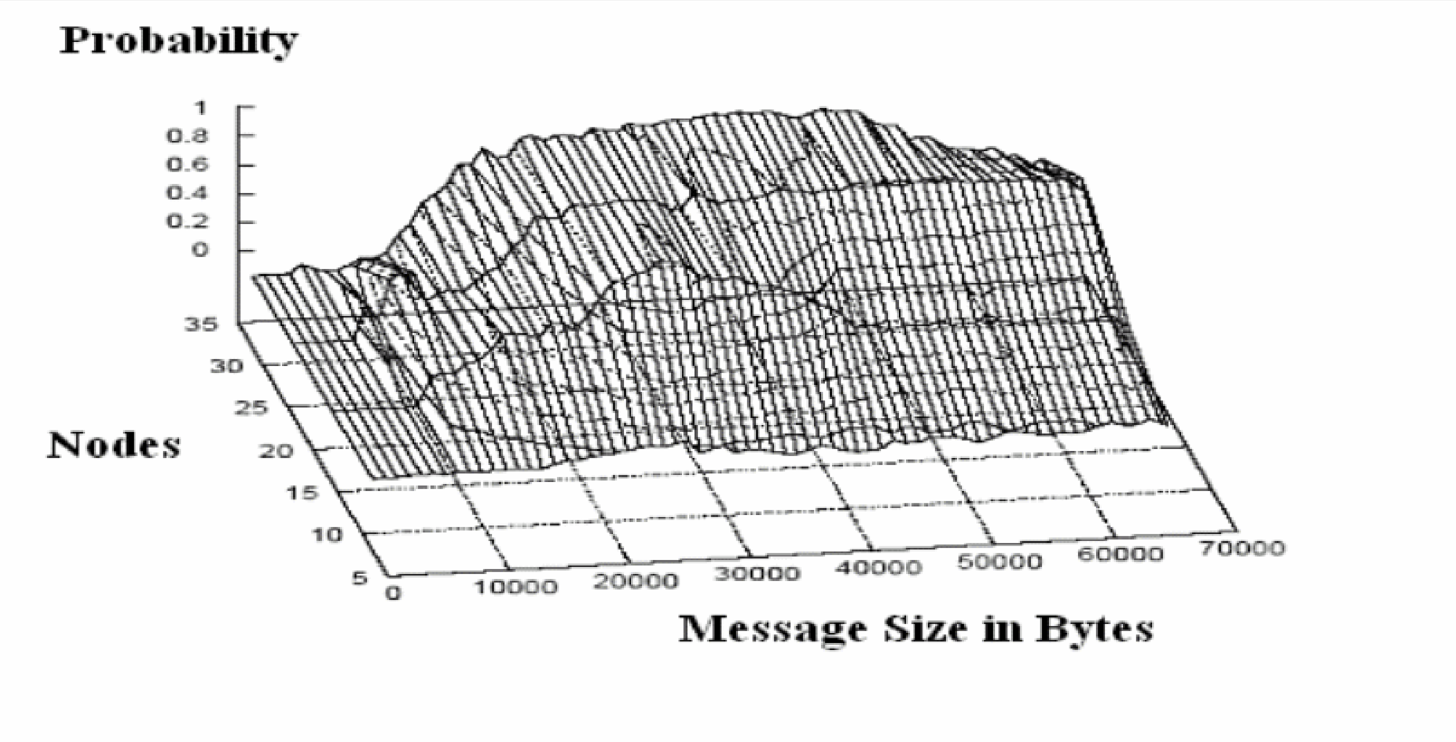
- escalations occur with 100% certainty



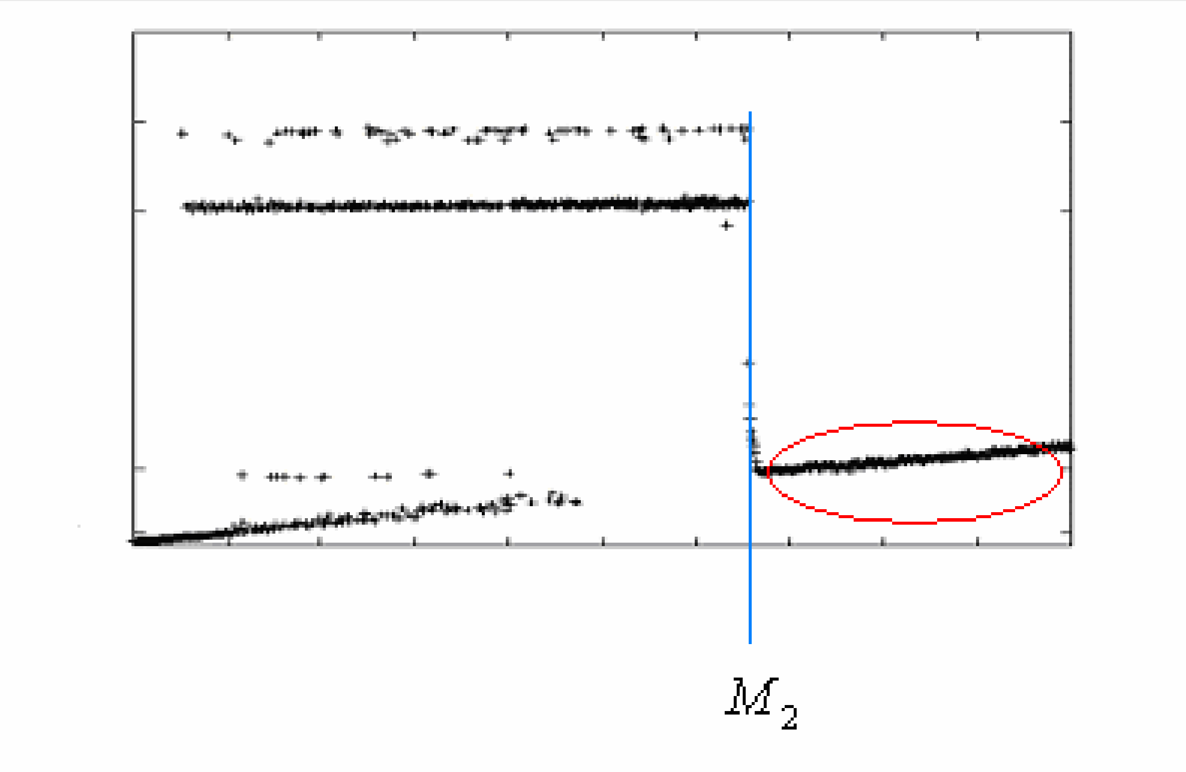
$P_i = P_i(n, M)$ - probability of escalation to T_i (i=1,2,3)

Probability of escalation

- A small number of discrete constant ***levels*** of escalation (10s and even 100s fold slowdown)
- ***Probabilities of escalation*** to each level



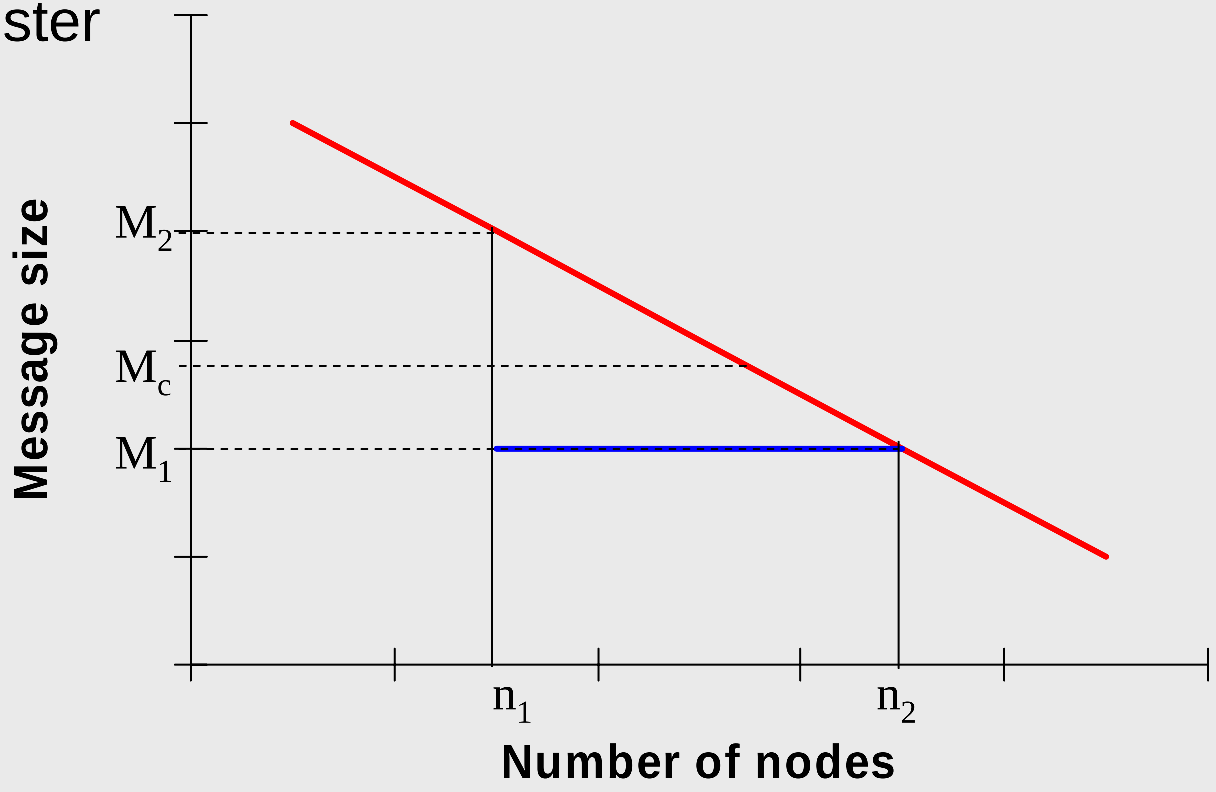
Many-to-one model for large messages



$$T = C_0 + t_0 M + \sum_{i=1}^{n} (C_i + t_i M + M / \beta_{0i}) + \kappa_2 + \kappa_3 M$$

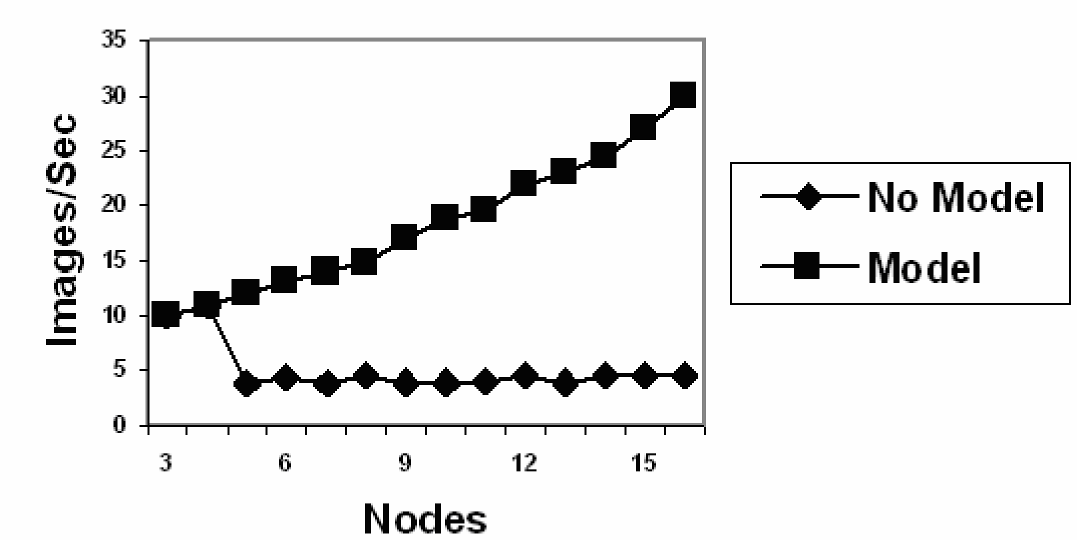
Application: Multi-spectral satellite imaging

- A typical real-time satellite imaging application (512x512 bytes)
- A sequence of raw data images divided into *partitions* for parallel processing by a cluster



Application: Multi-spectral satellite imaging (ctd)

- Calculate the number of sub-partitions m of a partition of the medium size M so that: $\frac{M}{m} \le M_1, \frac{M}{m-1} > M_1$
- Replace a single **`MPI_Gather`** with a sequence of m **`MPI_Gather`** for smaller messages



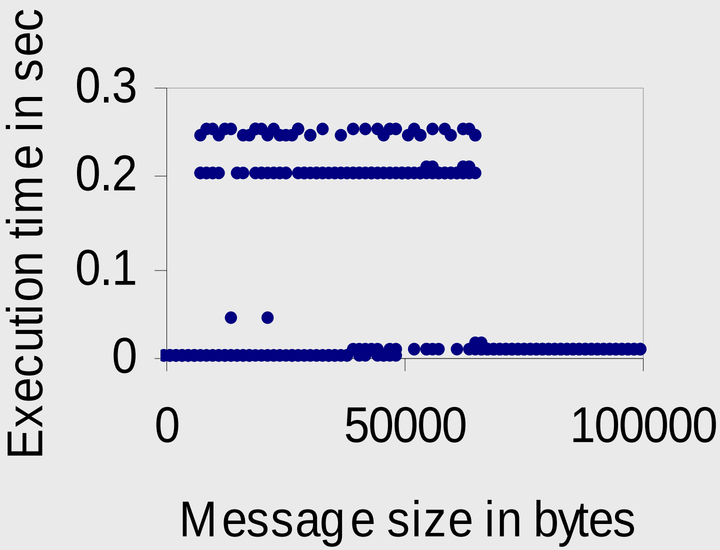
Application: Optimization of collective communications

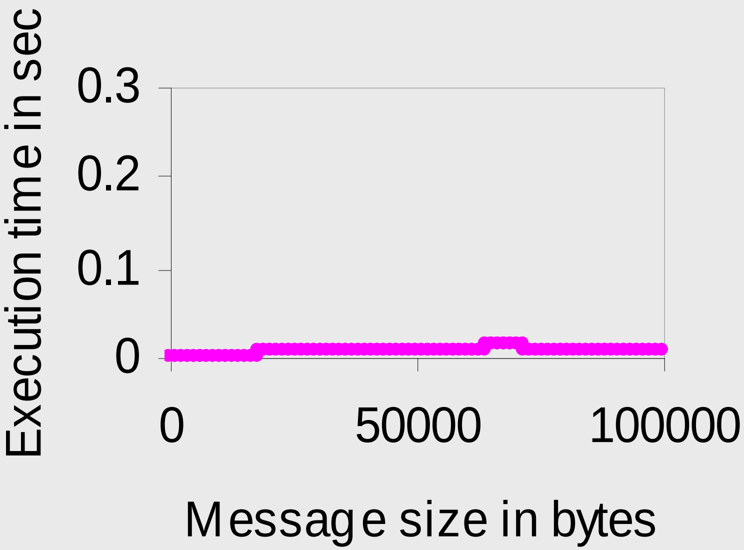
- **Idea**
  - Use the models for high level optimization of MPI collective communications
  - Implemented in HeteroMPI
    - Parameters of the models are found upon installation of HeteroMPI
- **`HMPI_Gather`**
  - Avoids escalations in the execution time for **`MPI_Gather`**
  - Revoke **`MPI_Gather`** for small and large messages
  - Implement by a sequence of calls to **`MPI_Gather`** (separated by barriers), each gathering small sub-messages ($<M_1$), for medium messages ($M_1 \leq M \leq M_2$)

Optimization of collective communications (ctd)

- **`HMPI_Scatter`**
  - Avoids the leap in the execution time for **`MPI_Scatter`**
  - Revoke **`MPI_Scatter`** for small and medium messages
  - Implement by an equivalent sequence of calls to **`MPI_Scatter`**, each scattering sub-messages of the size less than S

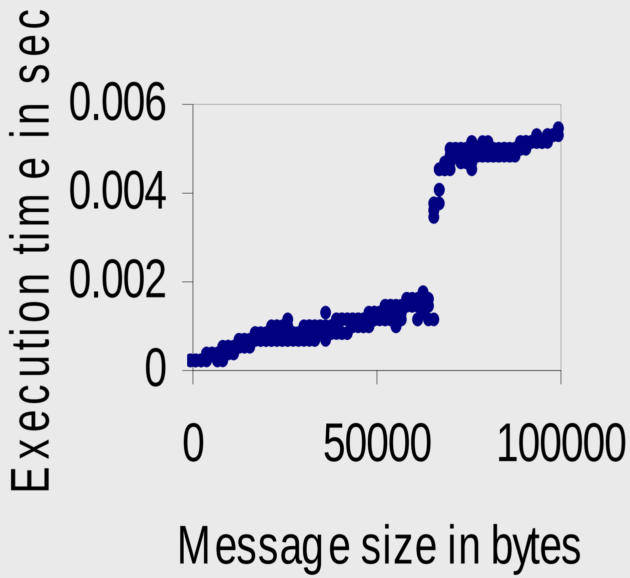
Optimization of collective communications (ctd)

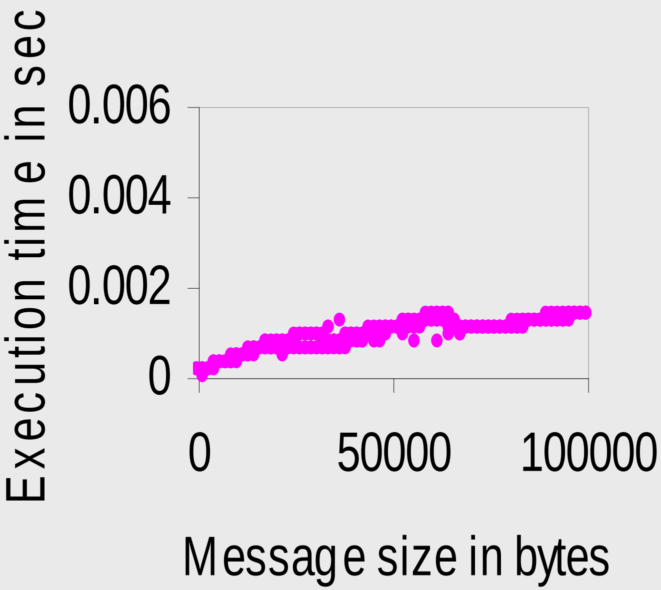




- Performance of native **`MPI_Gather`** and **`HMPI_Gather`**
  - LAM MPI 7.1.3 on a 16-node heterogeneous GigabitEthernet-based cluster

Optimization of collective communications (ctd)





- Performance of native **`MPI_Scatter`** and **`HMPI_Scatter`**
  - LAM MPI 7.1.3 on a 16-node heterogeneous GigabitEthernet-based cluster

Conclusion

- **Results**
  - Previously undocumented non-linear and non-deterministic behaviour of gather–like MPI communications for medium messages is reported and analysed
  - Many-to-one model is built on the empirical data and point-to-point model
  - Application of the model to optimization of MPI collective communications => to better performance of MPI-based applications on heterogeneous clusters

The work was supported by Science Foundation Ireland.