An efficiency and Scalability Model for Heterogeneous Clusters

Luis Pastor Pérez

José Luis Bosque

Universidad Rey Jun Carlos
Departament of Computer
Architecture

Index

1. Introduction.
2. Isoefficiency Model.
3. Heterogeneous Isoefficiency Function.
4. Experimental Results.
5. Conclusions and future work.

1. Introduction

- Efficiency and scalability are two very important features of parallel systems, particularly in clusters.
- Efficiency is an indication of the actual degree of performance achieved compared with the maximun value.
- Scalability gives us an idea about the system behaviour when the number of processors is increased.

1. Introduction

- An important issue to take into account is system heterogeneity.
- There are two main reasons that have stimulated the introduction of heterogenety in clusters:
  - The clusters flexibility and reconfigurability, that allow to change old nodes or to introduce new nodes very easily.
  - The tremendous improvement rate of PC's and workstations (Moore's Law), so newly introduced nodes will have very different computing capabilities.

1. Introduction

- All metrics and methods for efficiency and scalability assume homogeneous systems.
- Expressions for both, use "N" the *node count*
- In heterogeneous systems *node power* is also important.
- There is a strong need for developing tools and techniques for analyzing the behaviour of heterogeneous environments.

1. Introduction

- New contributions:
  - *A new definition of efficiency that can be applied both to homogeneous and heterogeneous systems.*
  - *A technique for analyzing scalability within heterogeneous systems, based on heterogeneous isoefficiency concept.*

2. Isoefficiency

- Isoefficiency take into account a parallel system composed by N identical processors.
- The Isoefficiency model is based in two considerations:
  – The efficiency of a parallel system decreases when the number of processors is increased.
  – For a given processor count, larger problem instances result in higher speedups and efficiencies.
- Simultaneous increase in both factors might keep efficiency constant.

C

• te t this
e p y

$$W = K \cdot T_o(N)$$

3. Heterogeneous Isoefficiency Model

- In heterogeneous systems the computational power depends on the number of processors but also on each node's individual power, which can be very different from one node to another.
- The isoefficiency model is based on the number of processors as a parameter, which is not valid in heterogeneous systems.
- Heterogeneity forces us to look for an alternative formulation for efficiency and isoefficiency.

3. Heterogeneous Isoefficiency Model

- Previous definitions:
  - The average computational power of the node “i” in a heterogeneous system (P_i) is the amount of work finished during a unit of time span in that processor.
  - The total computational power of a heterogeneous system composed of N processors, ($P_T(N)$) is the sum of the computational power of all processors in the system.
- The total computational power depends on the number of the processors and the computational power of each of then.

3. Heterogeneous Isoefficiency Model

- The *efficiency* of a parallel system is defined as the ratio between the best response time achievable for solving a specific problem in that system and the real response time achieved during the algorithm execution.

$$EF = \frac{W}{T_R \cdot P_T(N)}$$

3. Heterogeneous Isoefficiency Model

- Given a heterogeneous parallel system S(N, P_T, W) and a scaled system S'(N', P'_T, W') with $P'_T > P_T$ it can be said that S is a *scalable system* if, whenever the system is enlarged from S to S', it is possible to select a problem size W' such that the efficiencies of S and S' are kept constant.
- Efficiency depends on three factors:
  - The computational workload.
  - The system's total computational power.
  - The system response time.

3. Heterogeneous Isoefficiency Model

- To minimize the system response time all the processors should finish at the same time, which requires a well-balanced workload distribution.

- Depending on the workload distribution and on its associated overheads a combination algorithm-architecture will show a different scalability degree.

3. Heterogeneous Isoefficiency Model

- Given a heterogeneous system S (N, P_T, W), a scaled system S' (N', P'_T, W'), and a workload divisible *ad infinitum*. If the workload has been evenly distributed according to each node's computational power and the overhead times are constant for all of the processors, then both systems have the same efficiency if and only if:

$$W' = W \cdot \frac{P'_T \cdot C'_0(N')}{P_T \cdot C_0(N)}$$

3. Heterogeneous Isoefficiency Model

- For the same conditions if the overhead times can be expressed as a constant term for all of the processors, plus a second term proportional to each node's computational power, then the heterogeneous isoefficiency function is:

$$W' = W \frac{C_0 P_T' + C_1 \sum_{i=1}^{N'} (P'_i)^2}{C_0 P_T + C_1 \sum_{i=1}^{N} (P_i)^2}$$

3. Heterogeneous Isoefficiency Model

- For the same conditions, if the overhead times can be expressed as the sum of a term which is constant for all of the processors, plus a second term proportional to each node's workload, then the heterogeneous isoefficiency function is:

$$W' = W \frac{C_0 P'_T}{C_0 P_T + C_1 \frac{W}{P_T} \sum_{i=1}^{N} P_i^2 - C_1 \frac{W}{P_T} \sum_{i=1}^{N'} P'^2_i}$$

4. Experimental Results

- System architecture:
  - Hardware: PC cluster with 25 nodes, linked with fast Ethernet 100 Mbps.
  - Software: master/slave architecture, to solve a large data set, divided in work-packages.
  - Heterogeneity: extra load generated through different processes running in different nodes.

4. Experimental Results

- The overhead time has three components:
  - One term identical for all nodes: node initialization process.
  - A second term directly proportional to the number of nodes: intercommunication processes between the master and the slaves.
  - A third term direcly proportional to the number of data packages that have to be processed by each of the nodes: transmission of the data packages.

$$T_O = C_O + C_1 \cdot N + C_2 \cdot P_i \cdot \frac{W}{P_T}$$

4. Experimental Results

- It's possible to analytically deduce the heterogeneous isoefficiency function.
- Given a heterogeneous system S (N, P_T, W), a scaled system S' (N', P'_T, W'), and a workload divisible *ad infinitum*. If the workload is evenly distributed according to each node's computational power and the overhead times are the described above then the *Isoefficiency function* is:

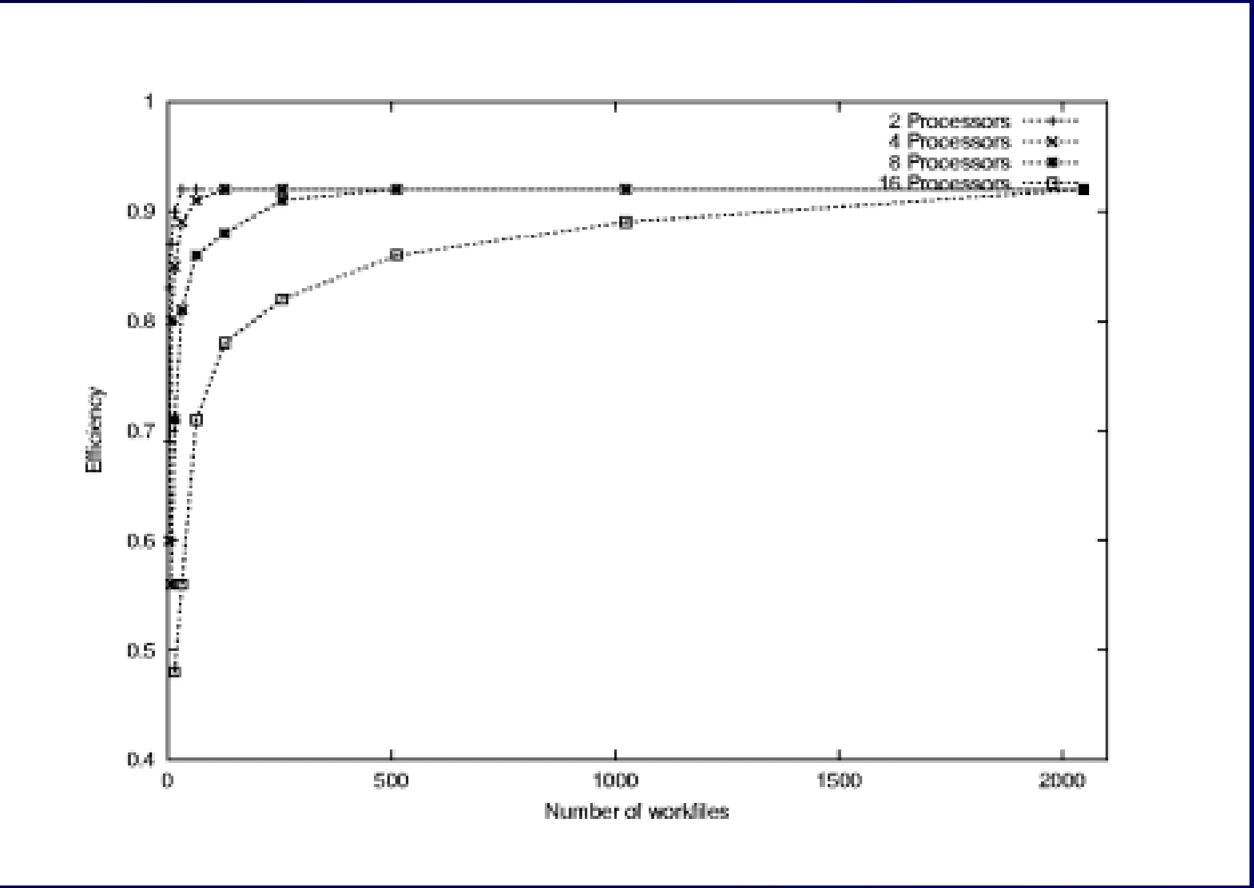
$$W' = W \frac{C_0 P_T' + C_1 N' P_T'}{C_0 P_T + C_1 N P_T + C_2 \frac{W}{P_T} \sum_{i=1}^{N} P_i^2 - C_2 \frac{W}{P_T'} \sum_{i=1}^{N'} (P_T')^2}$$

4. Experimental Results

- It's possible to perform tests to verify experimentally the validity of proposed models.
- The experimental setup has three objectives:
  - To verify experimentally that the heterogeneous efficiency definition can be applied both to homogeneous and heterogeneous systems, and becomes the classical efficiency whenever the cluster becomes homogeneous.
  - To verify that the homogeneous and heterogeneous isoefficiency functions can be applied to model and predict the system's scalability.
  - To show that it is not possible to apply the classical scalability and isoefficiency definitions to heterogeneous clusters.

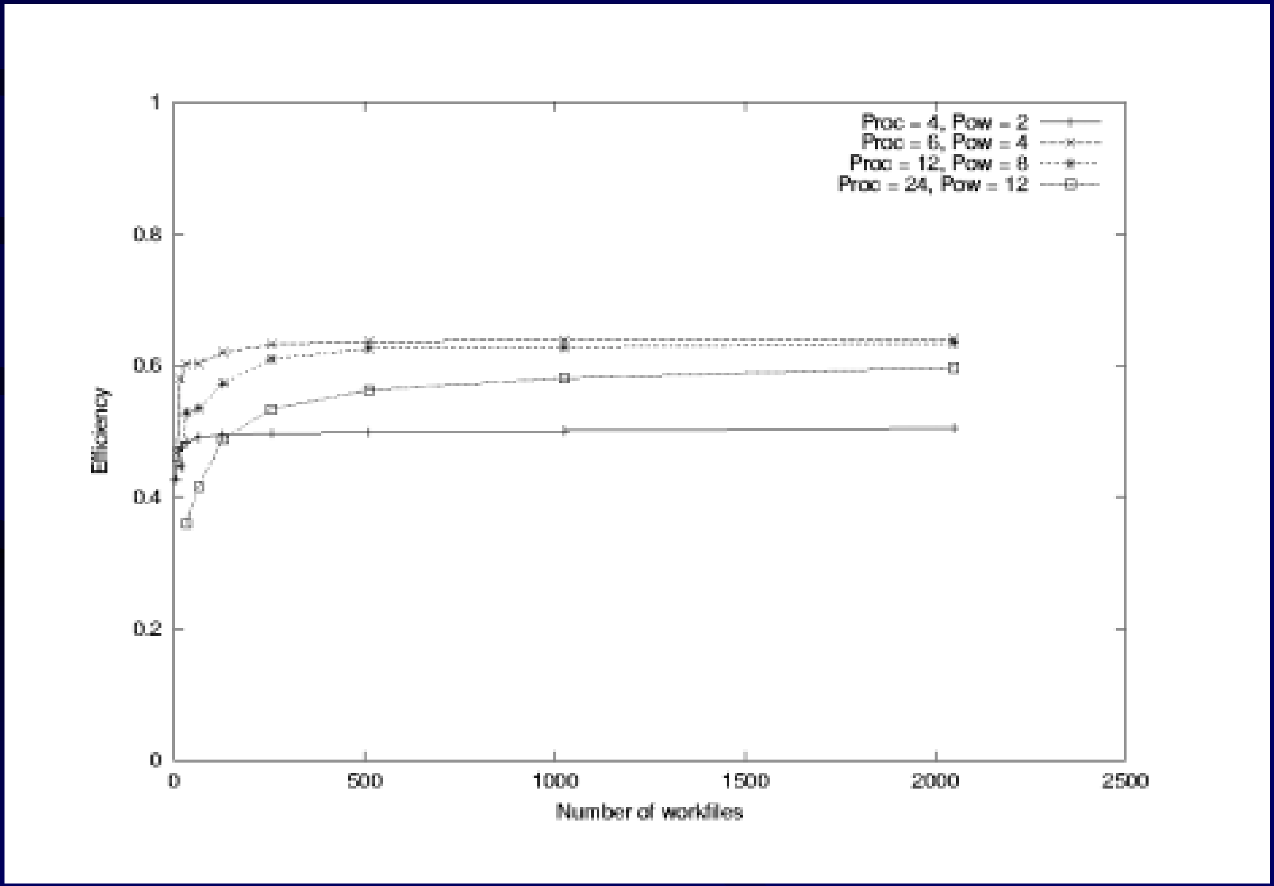
4. Experimental Results

Classical Efficiency in a homogeneous cluster.



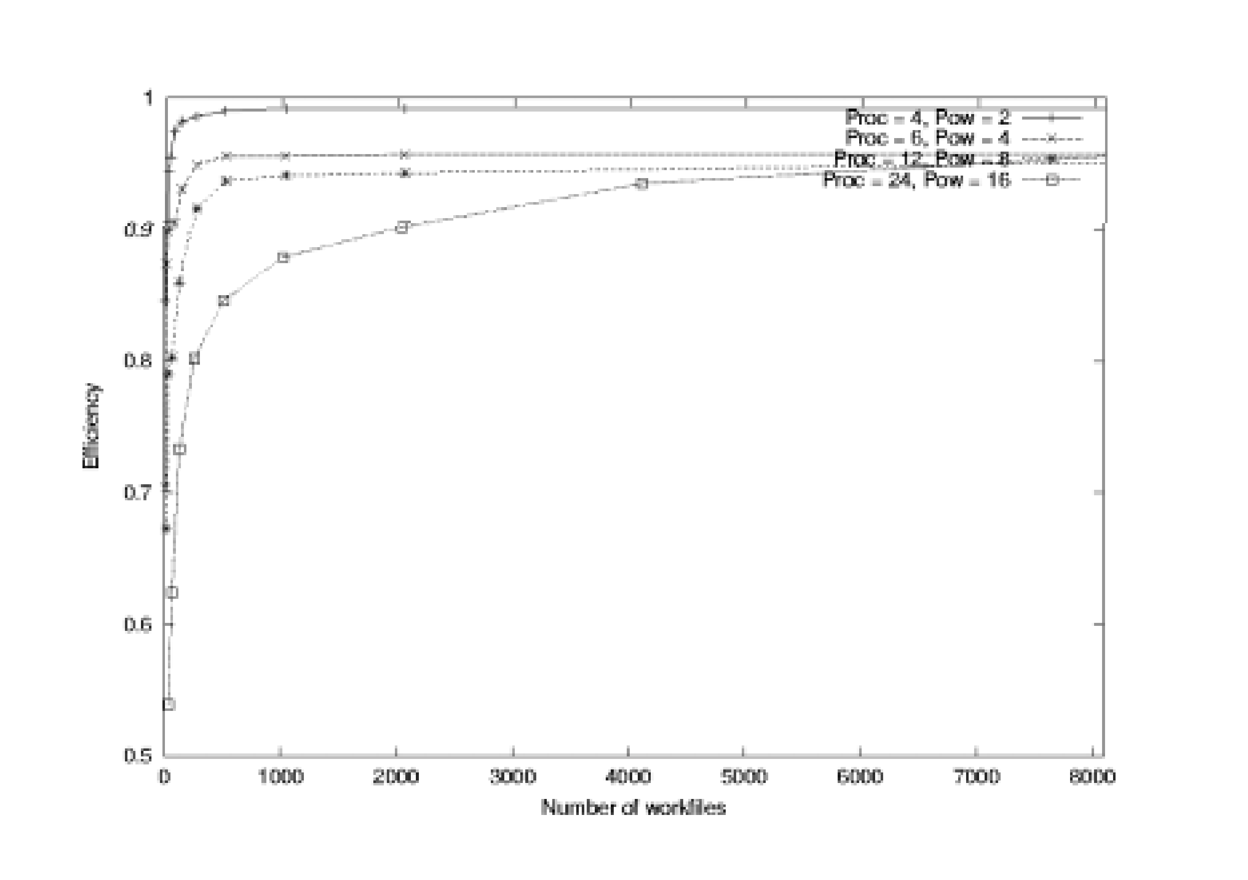
4. Experimental Results

Classical efficiency in a heterogeneous cluster



4. Experimental Results

Proposed efficiency in a heterogeneous cluster



5. Conclusions

- We present a new definition of efficiency which can be applied both to homogeneous and heterogeneous systems.
  - The efficiency definition does not need to refer to processor performances from any reference processor.
  - If we are able to estimate the workload magnitude for larger problem sizes, we do not need to obtain execution times over any single reference processor.
  - The experimental results confirm the proposed methods' validity.

5. Conclusions

- We presents a scalability model for homogeneous and heterogeneous systems.
- It is a priori method so, can be used for predicting algorithm scalability without the actual implementation.
- Future work includes the development of more systematic and precise methods for estimating both overhead and relative node computation power.