Distributed Optimization of Fiber Optic Network Layout using MATLAB

R. Pfarrhofer, M. Kelz, P. Bachhiesl, H. Stögner, and A. Uhl
uhl@cosy.sbg.ac.at

Outline

- Introduction
- MATLAB Custer Computing: MDICE
- Optimization of Fiber Optic Network Layout
  - Sequential Approach
  - Distribution Strategy
  - Experimental Results
- Conclusions

Introduction

- About 95% of the total costs for the implementation of fiber optic networks may be expected for the area-wide realization of the last mile (access networks) – planning tools currently are employed for the core net domain only.

- NETQUEST-OPT is a MATLAB-based planning tool for the computation of cost optimized and real world laying for fiber optic access networks, it is based on cluster strategies, exact and approximative graph theoretical algorithms, combinatorial optimization, and ring closure heuristics.

- NETQUEST-OPT currently requires computation times of several hours on a single workstation even for medium sized access domains.

- For an efficient planning process interactivity is a desired property, e.g. for rapid prototyping of different clustering strategies $\longrightarrow$ high performance computing systems are required.

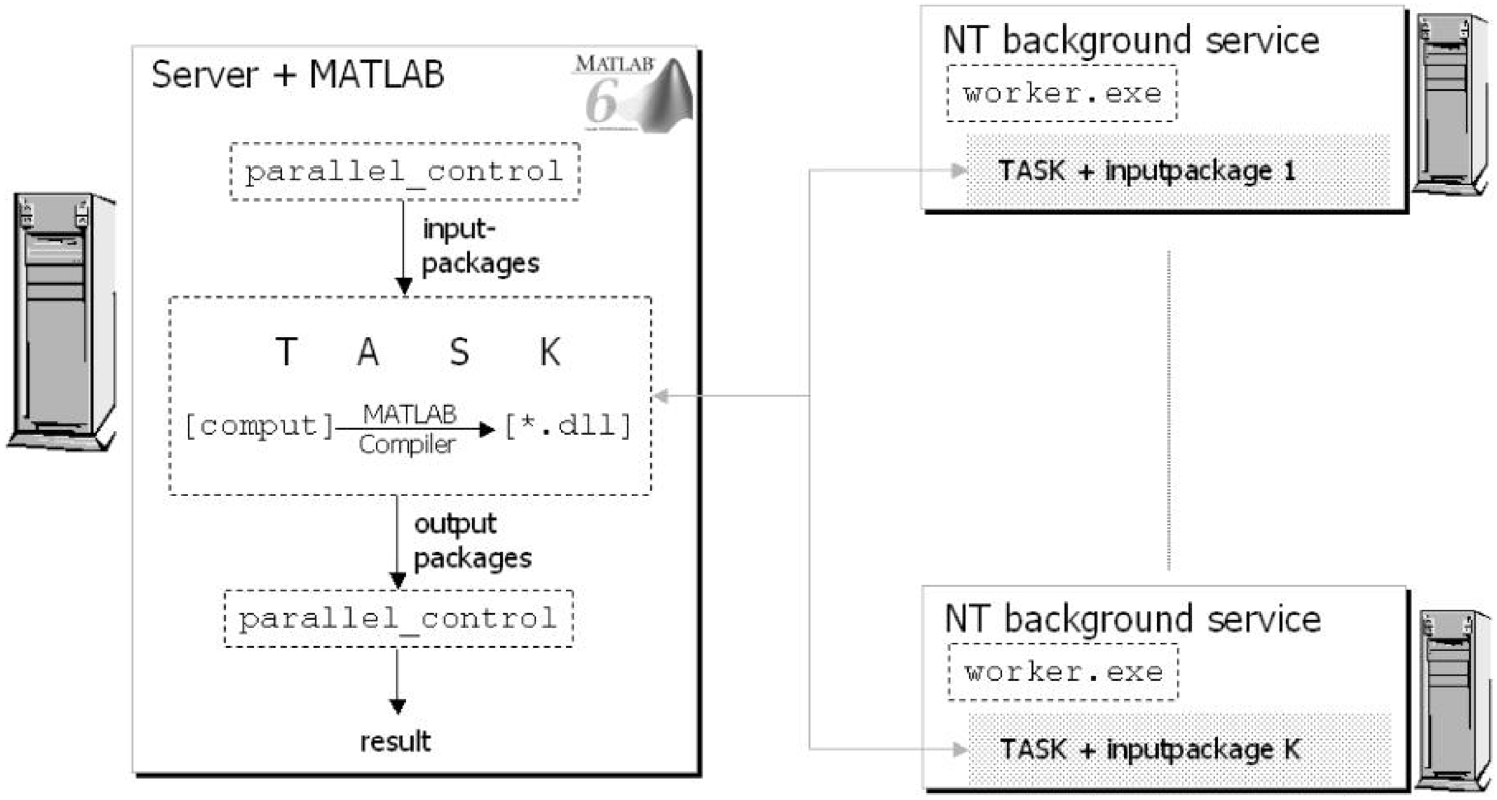
Cluster Computing and MATLAB

- Emerge of cluster computing → a cheap and highly available and yet powerful architecture for image processing applications
- MATLAB provides users with easy access to an extensive library of high quality numerical routines which can be used in a dynamical way and may be easily extended and integrated into existing applications.
- MATLAB in HPC
  - Developing a high performance interpreter (MPI/PVM based communication routines) or coarse grained parallelization by splitting up work among multiple MATLAB sessions.
  - Calling high performance numerical libraries (e.g. SCALAPACK)
  - Compiling MATLAB to another language (e.g. C, HPF)

MDICE

- MDICE: **M**ATLAB-based **DI**stributed **C**omputing **E**nvironment
- Goal: in contrast to previous approaches the goal is to get along with a single MATLAB client (instead of a MATLAB client at each participating compute node)
- The main idea was to change the client program in a way that it can be compiled to a standalone application.
- The compiled client program library and the datasets for the job are sent to the compute node, where a background client service is running with low priority.
- For this reason the involved client machines may be used as desktop machines by other users during the computation.

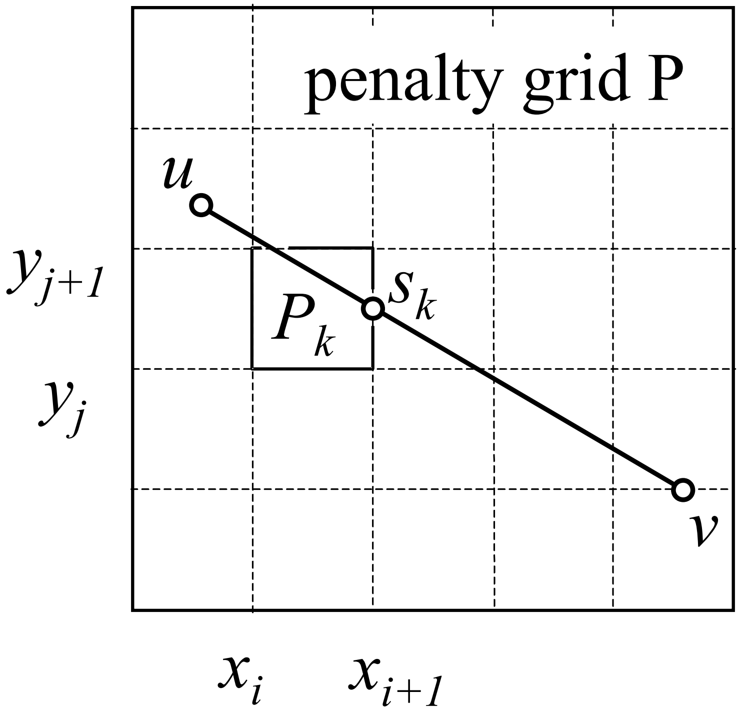
Client-Server Concept of MDICE



Download

http://www.fh-kaernten.at/nocms/studiengaenge/ma-tel/mdice/

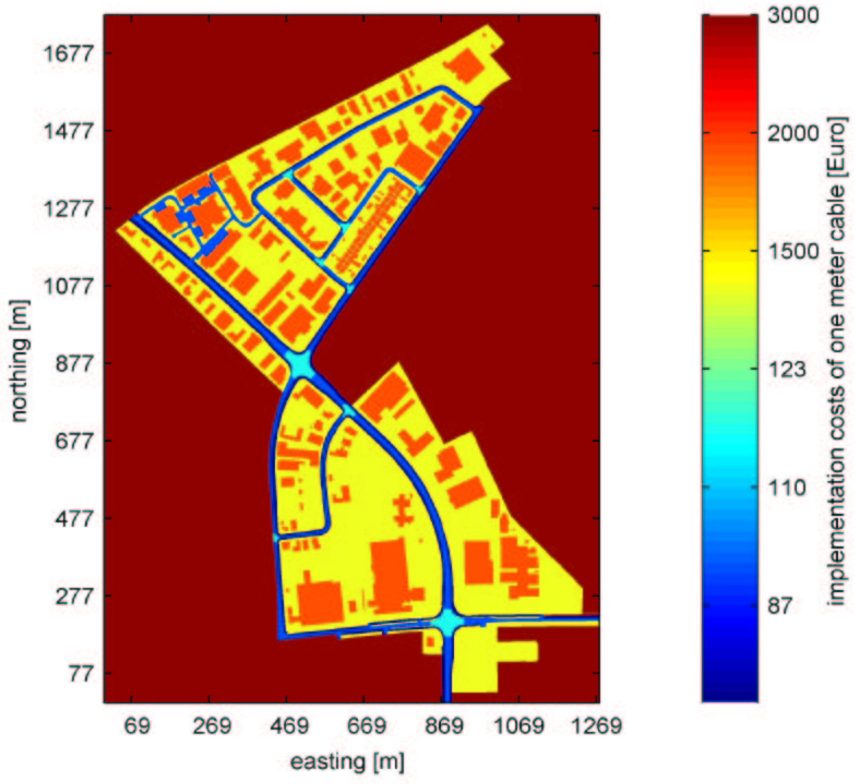
Optimization of Fiber Optic Network Layout: Sequential I



- We map the real world geometries to a set of nodes V of a graph $G = (V, E)$.

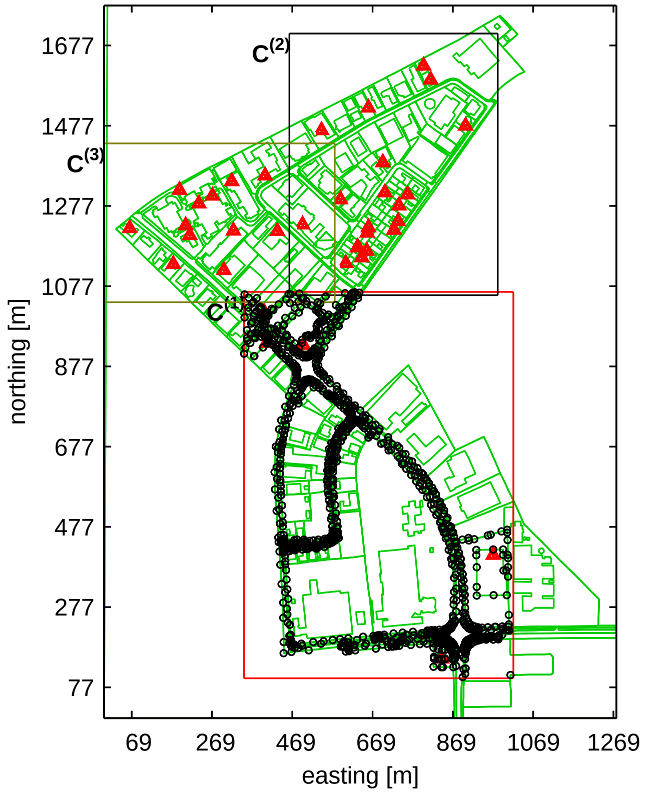
- **Penalty Grid**: A spatially balanced score card combines all relevant land classes with typical implementation costs. The access net domain is regarded as a regular grid $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$, i and j are indices of finite index sets. Each entry of the penalty matrix P describes the specific implementation costs for the grid pixel $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$.

Optimization of Fiber Optic Network Layout: Sequential II

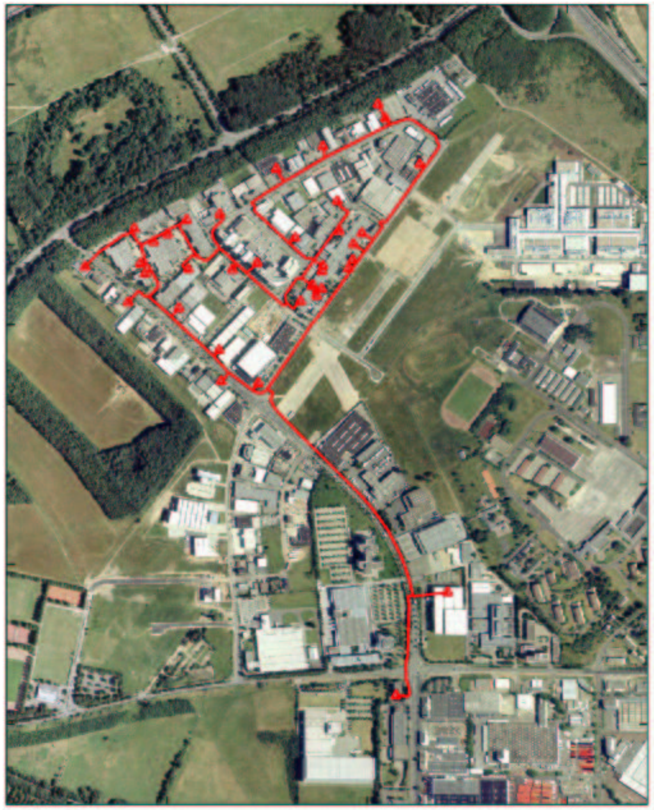


- For an edge $[u,v]$, $u \in V$, $v \in V$, $u \neq v$, we define the entry $W_{u,v}$ of the symmetric cost matrix $W \in Mat\left(|V| \times |V|\right)$ according to $W_{u,v} := \sum_{k=0}^{K-1} \|s_{k+1} - s_k\|_2 \, P_{k+1}$.
- The computation of W is one of the most time demanding procedures in the entire laying optimization process and requires $(|V|-1)\,|V|\,/2$ calls of computing $W_{u,v}$.
- In real world geometries, the network planning problem leads to dimensions up to $|V| \cong 20000$ nodes, therefore we first determine local clusters, compute a local solution in each cluster and find cluster shortest spanning trees.

Optimization of Fiber Optic Network Layout: Sequential III



37 access nodes (triangles) and three clusters



Arial view of target area (with overlayed computed solution)

Optimization of Network Layout: Distribution Strategy

- Inter-cluster approach: parallelization by segmentation of the spatial domain into independent clusters is efficient but leads to suboptimal global umbrella networks only, especially in case of a larger number of clusters which would be required for scalable parallel or distributed execution.

- Intra-cluster approach: does not exploit the parallelism introduced by the clustering process at all, therefore an optimal global solution may be obtained if clustering can be avoided. In this context, the computations associated with the evaluation of the cost matrix W (which are covered by our experiments) may be distributed in a simple fashion by domain decomposition without the requirement of inter-process communication.

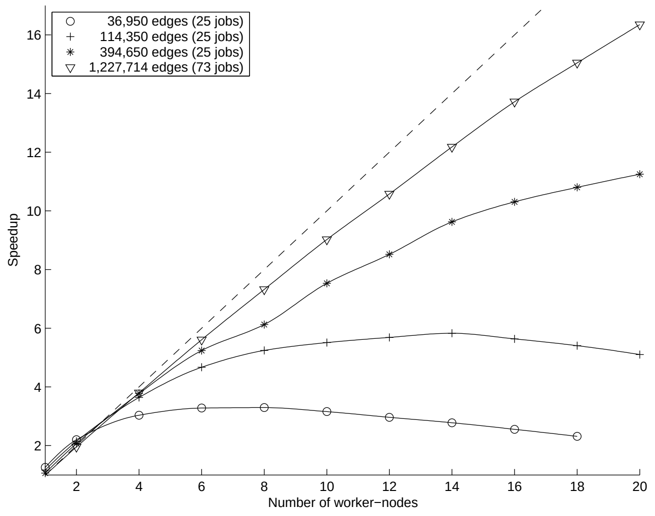
This simple partitioning approach leads to entirely deterministic behaviour since the computational effort of evaluating parts of the matrix W is not data dependent, however, load balancing functionalities are provided in order to be able to cope with the possible interference of other applications and heterogeneous environments.

Experimental Settings

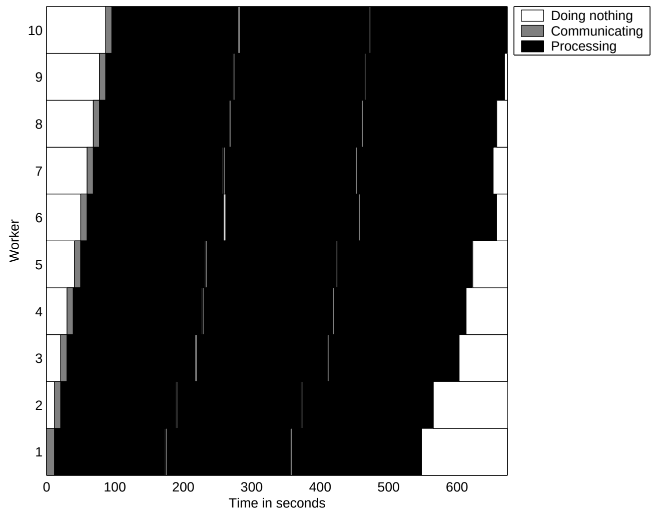
- We consider cost matrices W from real-world problems associated with a different number of edges (i.e. 36950, 114350, 394650, and 1227714) and vary the number of jobs distributed among the workers after fixing the edges to 394650.
- Computational task is split into a certain number of equally sized jobs N to be distributed by the server among the M clients in a dynamic fashion ("asynchronous single task pool method")
- Server machine (1.99 GHz Intel Pentium 4, 504 MB RAM) and client machines (996 MHz Intel Pentium 3, 128 MB RAM and 730 MHz Intel Pentium 3, 128 MB RAM), all types under Windows XP Prof., 100 MBit/s Ethernet network.
- Sequential reference times have been achieved on a 996 MHz client machine with a compiled (not interpreted) version of the application to allow a fair comparison
- We use MATLAB 6.5.0 with the MATLAB compiler 3.0 and the LCC C compiler 2.4.
- Homogenous environment (fast clients only) vs. Heterogenous environment (six 996 MHz and four 730 MHz clients)

Experimental Results I: Homogenous Environment

- It is clearly exhibited that speedup saturates for the smaller problems at 4 or 8 clients, respectively. On the other hand, reasonable scalability is shown for the larger problems up to 20 clients.
- The number of clients is set to 10, and 30 jobs are used. The staggered start of computation at different clients which is due to the expensive initial communication phase.



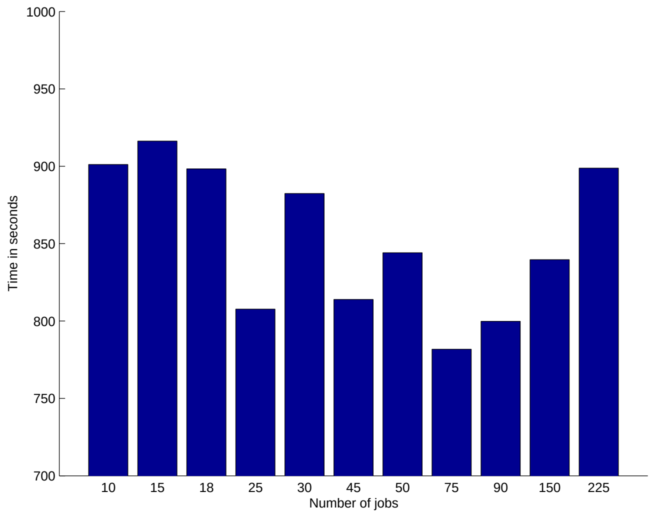
Speedup for varying problem-size



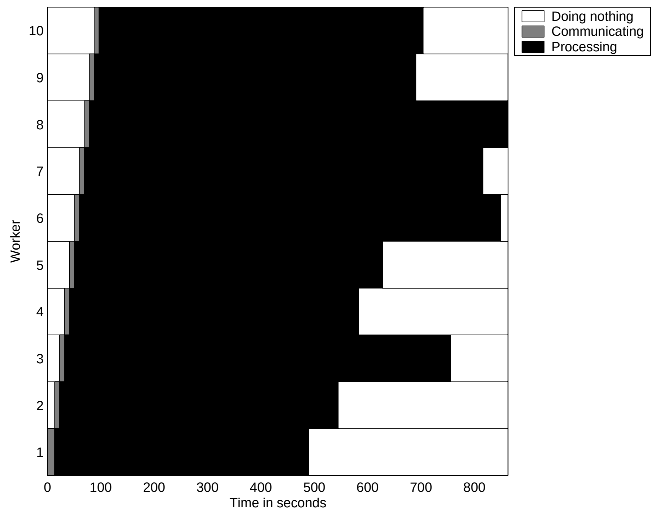
Visualization of execution

Experimental Results II: Heterogenous Environment

- We vary the number of jobs distributed among the clients. Clearly, a high number of jobs leads to an excellent load distribution, but on the other hand the communication effort is increased thereby reducing the overall efficiency. The optimal number of jobs is identified to be 75.
- With 10 jobs on 10 clients, the performance is worse as compared to the analogous homogenous case and the slower clients (numbers 3,6,7,8) are immediately identified.



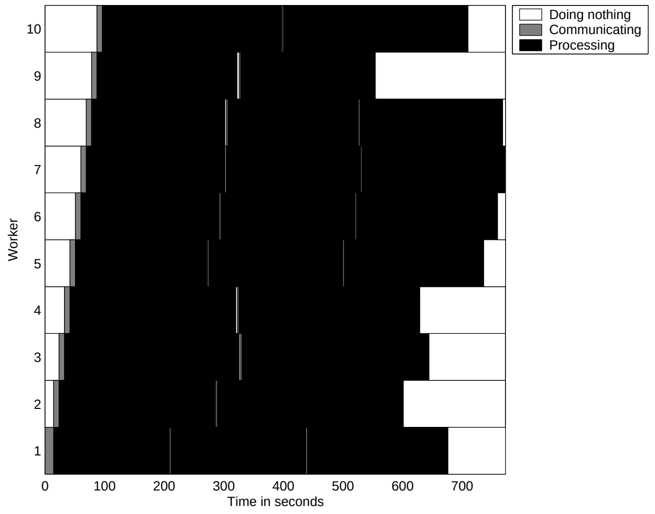
Execution time
(varying number of jobs)



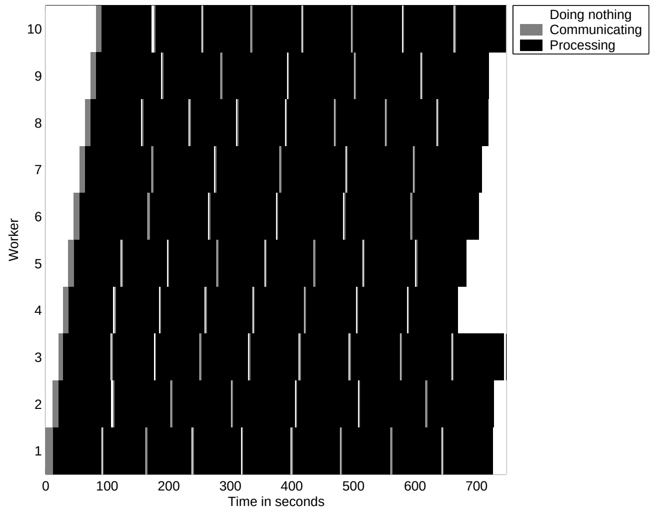
Visualization of execution
(10 jobs)

Experimental Results III: Execution Visualization

- Execution behaviour for 25 and 75 jobs: visual appearance of the two cases is quite different, but we result in almost identical execution times.
- Whereas for $N = 25$ we have little communication effort but highly unbalanced load, the opposite is true for $N = 75$.



25 Jobs



75 Jobs

Conclusions

1. The custom MATLAB toolbox MDICE may take advantage of the large number of Windows NT-architecture based machines available in companies and universities at low licensing costs.

2. The distributed approach suggested in this work allows for an interactive network planning process provided enough PCs are available.

3. Additionally, our approach may lead to a better quality of the computed network solution as compared to a sequential clustering approach in case the clustering can be avoided in the distributed execution.