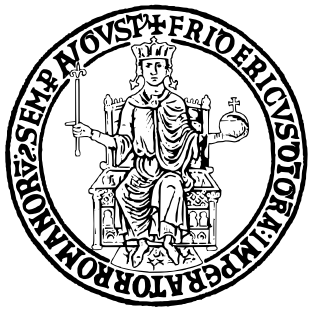
Università degli Studi di Napoli Federico II

Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione



Parallel and Distributed Computing submissions

Giuliano Aiello

2024

Contents

Chapter 1

Prolusion

1.1 Goal

This document offers a comprehensive overview of a project developed in `C`, consisting of multiple modules delivered in incremental phases.

It is not intended as a user guide, but rather aims to describe the project's exploration of parallel computing techniques, leveraging High Performance Computing in certain instances.

1.2 Environment

The project was entirely developed on a macOS system with the help of Xcode IDE.
Naturally, this will mainly impact the build process.

1.3 Project directory layout

The structure of the project's root directory is outlined below.

```
parallel-distributed-computing/
├── common/
├── hpc/
│   ├── gemm/
│   ├── matmatblock/
│   ├── matmatdist/
│   └── matmatthread/
├── laplace/
├── maxsum/
├── ringsum/
└── parallel-distributed-computing.entitlements
```

`common` package serves as a library of utility functions designed to support and be reused by various modules across the project.

The remaining directories represent the individual project modules, which constitute the deliverables of the project. Within each module, the directory structure follows a standard format:

```
<module>/
├── build/
│   ├── deploy-cluster.pbs
│   └── Makefile[.gcc]
├── src/
│   ├── <module>/
│   └── main.c
├── config.sh
└── run.sh
```

Most parts of the `main.c` files are provided by the project supervisor.

1.4 Build

The project was primarily compiled using the Clang compiler.
The build process was carried out either through the `Makefile` (some of which support compilers other than Clang) or via Xcode.

Regardless of the build process, every module of the project was compiled with:

- `-O3` optimization flag to maximize performance.

1.4.1 Libraries

The following are the dynamically linked libraries integrated into the project.

- `math.h`
- `mpi.h`
- `omp.h`
- `stdio.h`
- `stdbool.h`
- `stdlib.h`
- `sys/time.h`
- `unistd.h`

1.4.2 Xcode

When it came to build with the Xcode, the development process adhered to the workflow and conventions defined by the chosen IDE, leveraging its built-in tools and features to organize and manage the project. Particularly, this includes:

- Xcode targets
- Xcode schemes
- Xcode `.entitlements` file

1.5 Run

1.5.1 Network Requirement

Running an MPI module with no internet connection, makes the following error occur:

```
[Giulianos-MacBook-Pro.local:05355] ptl_tool: problems getting address for
index 0 (kernel index -1)
--------------------------------------------------------------------------
The PMIx server's listener thread failed to start. We cannot continue.
--------------------------------------------------------------------------
Program ended with exit code: 213
```

Acronyms

HPC High Performance Computing 1

IDE Integrated Development Environment 1, 3

MPI Message Passing Interface 4