Standard Skeleton/Parallel Component Library

Project Structure (Draft of July 4, 2001)

Here are some thoughts on how we might structure the project. To satisfy the rules we must have a collection of linked workpackages and clearly specified deliverables. First I give some overview. Then I sketch a possible set of workpackages. Finally there is a time chart which shows how these fit together. This document complements Herbert's document which sketched motivation and technical background.

Outcomes

What are the overall objectives of the project? I think we have three:

1. to produce a standard specification for a skeleton/parallel component library, including
   (a) an overview of our conceptual model;
   (b) an abstract (in the sense of being language independent) specification of our components and utility functions;
   (c) concrete instantiations of the abstract spec into a range of host languages (eg C/MPI, C/BSP, Haskell, or others) (ie APIs for these), ideally with at least prototype implementations
2. to produce a sample collection of small and medium scale applications coded in the concrete APIs, (which could become a benchmark if events so decide);
3. to produce a list of *desiderata* for tools which could support a programming methodology based on our standard, and some "executable specification" style cost modelling implementation.

Other remarks: we should discuss links to "grid" and "pattern" work.

Process

The project is structured into a number of workpackages, each of which has

- one workpackage leader
- a small core team (say 3 or 4 people, including the leader)
- one deliverable

Each workpackage will have two face-to-face meetings of its team, and will solicit regular feedback from the the wider project. There will be a web repository of all documents as the develop. Additionally, there will be two meetings of the entire team, one at the start of the process and one mid way through.

Timescale

The entire project will last 24 months, as indicated on the time plan.

Participants

All the interested academics. Can we get any industrial interest?

Support

The main item we request is support for travel/meetings. Here is one guessed quantification. People with more experience please say if this sounds ridiculous.

There are eight workpackages, so there will be 16 meetings involving four people (the core teams) plus two full project meetings (involving maybe 10 people?). So, we have to fund around 84 units of "one person attending a meeting". Setting an average unit cost of 900 Euros (ie travel, two nights accomondation, small overheads), that would be 75600 Euros. Add 8 individual conference trips (ie to present our stuff) at maybe 1500 Euros each (to also include conference fees) , ie another 12000 Euros. That's 876000 Euros for travel/meetings. Over two years.

Additionally, (and this is the bit which makes our respective institutions interested) I believe that we can claim money for managerial support (ie to employ someone to "manage" the project, or more realistically, a fraction of the time of an existing person), together with costs for computing equipment that we already have but will use (such as a web server for all our docs), also consumables etc, and a general 20% of direct costs as "overheads". No doubt I can get guidance on all of that side of it.

Workpackages

Volunteers to lead preparation of WP specs (including order of individual preference):
WP1: Kuchen(3), Cole(2), Pelagatti(2)
WP2: Kuchen(2), Cole(1), Pelagatti(3)
WP3: Hains, Pelagatti(3)
WP4: Kuchen(1), Gorlatch (pref Java), Pelagatti(1), Cole (3)
WP5: Tiskin (C/BSP/Java)
WP6: Serot (Caml)
WP7: Kuchen(4)
WP8: Cole (4)

WP 1: Pseudo Application Set

Before we start to specify our framework we should give some serious thought to the range of programs we expect to be able to express. To do so we will collect a range of application sketches, ranging from the simple to the realistic, covering a wide range of parallel program structures. This will be deliberately informal. The aim is to try to cover the kind of things parallel programmers will naturally want to express, not to squeeze them into any chosen format.
Deliverable: a document outlining these programs and the key aspects of their structure, with at most a page or two devoted to each[1].

WP 2: Conceptual Model & Abstract Spec

This is the core work package, in which we pin down exactly what we are talking about.
Deliverable: a document describing the overall conceptual framework, and more formally, describing the specific set of functions and utilities to be provided by the library, described in a language independent way.

WP 3: Tools

Here we imagine what a program development environment and methodology based around our library would look like. What tools would there be for performance prediction, monitoring and management?
Deliverable: a document describing the above and some implementation, eg executable specifications for cost modelling

WP 4, 5, 6: Concrete Instantiations

Here we show how the abstract concepts from WP2 can be instantiated for real languages. We can discuss exactly what this means. I have provided some examples for the sake of argument. To me, a "real language" on this context means a combination of an existing

[1]For example, in the case of pipelines, some examples require the stages to have internal state, others don't. Similarly for some examples the "result" is the sequence of objects emitted by the last stage, for others (eg Sieve of Eratosthenes) the result is left behind in the stages.

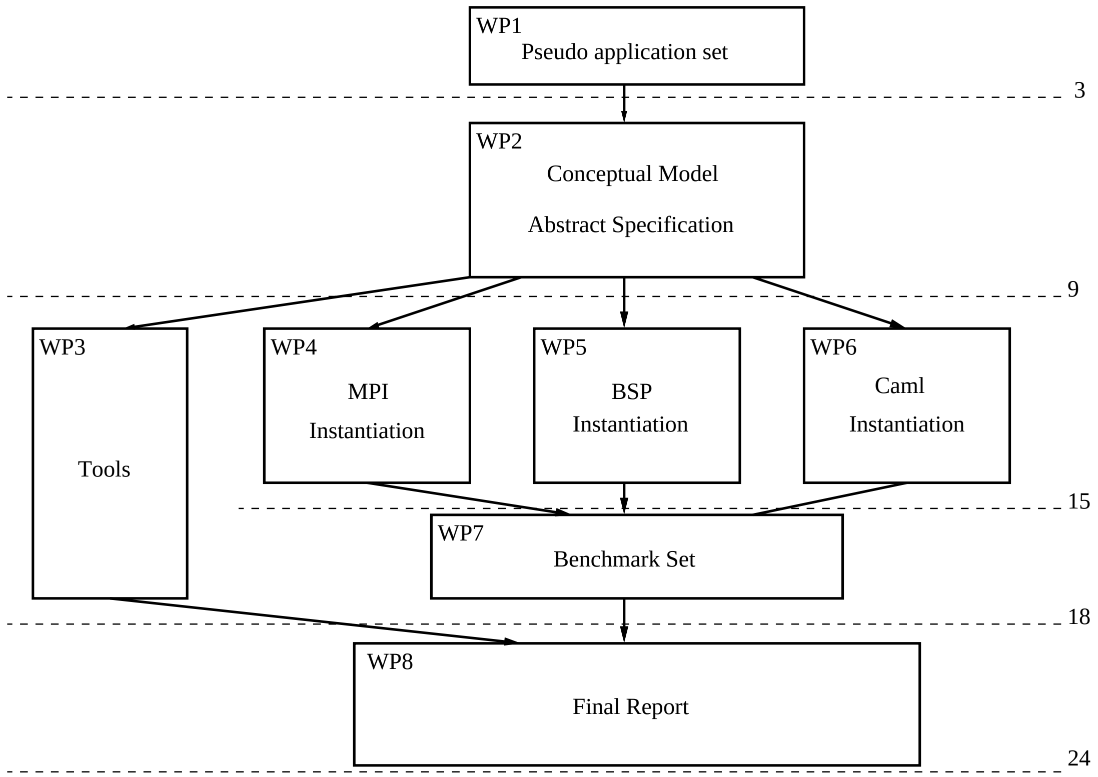
sequential language and a parallelisation mechanism (for example C/MPI or Fortran/BSP). There are too many permutations to cover, so we will have to choose carefully. **Deliverables:** Specifications of the bindings from our library to various real platforms, and hopefully prototype implementations

WP 7

Here we demonstrate the semantic applicability of our library to real problems, some drawn from WP1, others new.
Deliverable: A set of programs for real problems in expressed in the real instantiations of the library, each with a short tutorial discussion document and (?) a set of timing results across a range of architectures.

WP 8

Pull everything together. **Deliverable:** A final report, pulling together the documents which describe our overall outcomes as listed above.


WP1
Pseudo application set
3
WP2
Conceptual Model
Abstract Specification
9
WP3
Tools
WP4
MPI
Instantiation
WP5
BSP
Instantiation
WP6
Caml
Instantiation
15
WP7
Benchmark Set
18
WP8
Final Report
24