Response to referee's memo

The referee makes a number of comments; we have taken them on board in this revised submission. We thank the reviewer for his comments, and trust the concerns will have been settled.

The comments possibly reflect a failing on our part to appreciate the diversity of readership of CPE and so we have added paragraphs placing our work in context; we no longer assume all readers to be familiar with the PRAM. The comments were:

> *U. S. readers will find the paper difficult to understand as it uses a rather parochial European notation. It is fine to use the BSP library, but MPI undoubtedly supports the BSP style (which style is well known without a fancy name and is in common use by PVM and MPI programmers) and the conceptual discussion should reflect this. MPI is of course the dominant message passing system in international state of the art practice.*

This is a response from a more practical side of concurrency than the formal theory base from which we work. We appreciate the point, but see a small number of theorems and algorithms as essential to justify our approach; we have tried to strike a balance. Our side is not solely European - BSP is due to Valiant of Harvard! Similar presentation is usual from Computer Science departments of many US schools.

That BSP exists within MPI and PVM is important: Restricting MPI and PVM to a simpler subset of functions uncovers a simple efficient model. The reviewer's criticism is a reflection of an omission to justify BSP - this has now been corrected by mention of the models and the close relationship with BSP is made clear.

> *.. it would be useful to mention other attempts to implement a shared memory programming model (The US name for PRAM programming). Treadmarks from Rice is the best known system in the US.*

This comment is an unexpected misconception. The PRAM is no more European than apple pie! The PRAM is due to Fortune and Wyllie of Cornell (1978). *The* book of parallel algorithms is by JáJá of Maryland (1992) - in this the PRAM is the only model considered. Similarly the US academics Cormen, Leiserson and Rivest use the PRAM for parallel algorithms in their essential book "Introduction to Algorithms" (1991). The models are very different. The PRAM is highly synchronised compared to the use of locks in shared memory programming. A PRAM also has an unlimited number of physical processors, something we simulate.

The PRAM model has been made clearer in the introduction and we have further emphasized the difference in the conclusion of the revised version. Direct comparison with non-PRAM systems would not be fruitful and would introduce confusion; indeed we would not want to be compared to such systems as we have different goals, one of which is tremendous scalability to systems such as the Cray T3D with 256 real processors. Suitable comparable systems had already been considered: SB-PRAM, FORK and 11 (published in CPE).

With best wishes,

David Lecomber