A Test Bed for Modelling the Liberation of Silica from Feldspar Using Parallel Computing Techniques

Shirley A. Williams, Graham E. Fagg and Philip C. H. Mitchell
Departments of Computer Science and Chemistry, Whiteknights,
The University of Reading, Reading, UK, RG6 2AY.
e-mail: shirley.williams@reading.ac.uk

Introduction

Experimental data on the dissolution of silica from feldspar at various pH values is available (e.g. Wollast [1]), we wished to model the dissolution rate as a summary succession discrete event simulation so that all of the dissolution events over a time interval were modelled. Wollast recorded data over a period of 25 days; modelling each event in this time period is therefore a computationally intensive procedure.

A model has been developed of the liberation of silica from a potassic feldspar in solutions buffered at various pH values. The programs developed use PVM constructs to take advantage of the powerful computing facilities offered by clusters of workstations. The parallelism allows many simulations to be run simultaneously, at an almost linear speed up. Thus it is possible to experiment with the variables which influence the dissolution and functions used to describe the process; and rapidly compare a computed set of results with experimental work.

The high power provided by clusters of computers means we are able to include a large quantity of events in simulations that approach the number in real physical processes and so we are able to avoid the need to set up analytic expressions.

A Sequential Model

The liberation of silica is expressed experimentally as the increase in concentration of silica in the aqueous phase. We chose to model the liberation of silicon species from the feldspar. Wollast's work indicates that the liberation of silica is dependent on the pH value of the solution and the concentration of silica already in the solution. At a molecular level we can define a time interval t in which we expect an atom of silicon to be liberated. The time interval will be a function of the pH effect, number of particles already in solution and the sample size.

Our initial approach took the effect of pH at pH 4 to be 1 and the sample size used by Wollast to also be 1. This meant that the units we produced our results in would be arbitrary, but they could then be mapped on to the experimental results available. Thus the first liberation of silica was assumed to happen in the time range 0 to 1 (the units of measurement at this point were not significant). A random number was generated in this range using a sophisticated random number generator, earlier work [2] had shown that results for large scale simulations were questionable when simple random number generators were used. The time of the next liberation would then be in the range:

$$0 \text{ to } 1 + \delta t$$

Where δt is a function of the number (n) of particles previously released, the sample size (s) and a constant (k) dictated by the pH value of the solution. The function is defined to be:

$$n^3 * k / s$$

For a pH value of 4 the constant is taken to be 1. Thus the average time for particles to be released increases in proportion to the cube power of the number of particles already released.

Thus the basic algorithm is:

```
1.Initialise 0 particles released at time 0

2.Calculate time from now to release of next particle, as:

        random number * factor related to pH *
        cubed particles already released

3.Repeat 2. until sufficient particles released.
```

The simulation was run a number of times, each starting with a different seed for the random number generator, and the average result found. The match produced was reasonable and so the approach was repeated for other pH values, altering the value of k to produce a close match on the graph of experimental data.

A Parallel Model

The speed of the computation could be readily increased by running each iteration of the simulation of a particular pH value on a separate processor. This is achieved using PVM [3] with a master process calculating initial seeds, passing these values to identical slaves that perform the same basic calculation. The master then collects and averages the results. The performance increase with this approach is linear [4]. With five simulation runs using PVM taking one fifth of the corresponding serial time.

It is also possible to run simulations for all pH values in parallel. Our approach to this was to introduce a level between the master and the slaves, that we call the servant. There is one servant started for each pH value. Each servant starts a number of slaves which carry out the simulations for that pH value.

Master Process

```
1.Initialise
2.Start one servant for each pH value
3.Receive results from servants
4.Display.
```

Servants

```
1.Receive pH value
2.Start the slaves
3.Calculate seeds
4.Pass seeds to slaves
5.Receive results from slaves
6.Pass results back to master.
```

Slaves

```
1.Receive seed
2.Initialise random number generator
3.Calculate time of release of particles, according to
  function.
4.Pass results back to servant.
```

A Parallel Workbench

Although the results produced by the initial simulations provided a reasonable match (see figure 1), it was felt that not all factors were taken into account in the early stages of the process and so the match was not as satisfactory as it could have been. Wollast suggests a residual layer has an effect on the rate of dissolution, most certainly the ambient temperature also effects the rate of dissolution. We wished to be able to vary the variables and the function used to calculate dissolution rate, quickly run the simulations, for each pH value, and then compare the calculated results with the experimental results.

We took the same approach as described above, using a single file to contain variables and the function, thus a simple edit to the file and recompilation enabled variants to be tried. By spreading the work across multiple machines the results for each pH value was available simultaneously and the precision compared to the experimental results easily calculated.

Conclusions

The use of clusters of workstations have enabled us to develop a test bed for the simulation of the actions of particles, we can easily adjust the variables and functions used, then be able to rapidly compare our results with experimental data. This approach can be applied to other chemical processes and extended to more complex systems, such as the liberation and depositing of silica in sea water.

References

[1] R. Wollast, *Kinetics of the alteration of K-feldspar in buffered solutions at low temperature*, Geochimica et Cosmochimica Acta, 31 (1967), 635-648.

[2] S.A. Williams, P.C.H. Mitchell and G.E. Fagg, *A Cluster Computing Implementation of a Monte Carlo Simulation of a Particle Growth Mechanism*, Parallel Algorithms and Applications 3 (1994) (to appear).

[3] V. Sunderam, *Pvm: A Framework for Parallel Distributed Computing*, Concurrency Practice and Experience, 2 (1990), 315-339.

[4] G.E. Fagg and S. A. Williams, *The Use of PVM on a Campus Wide Network*, First European PVM Users Group Meeting, Rome, October, 1994.

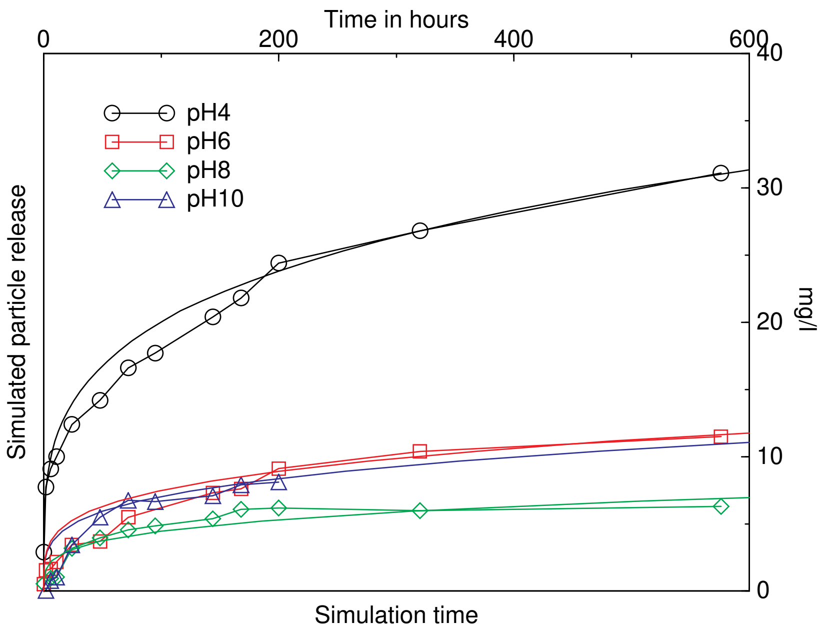

Figure 1: Experimental and Simulated Data

Experimental data marked with symbols, simulated data has solid lines.