Introduction to Parallel Computing
Errata

John C. Kirk

27 November, 2004

Overview

Book: Introduction to Parallel Computing, Second Edition, first printing (hardback)
ISBN: 0-201-64865-2

Official book website:
`http://www-users.cs.umn.edu/~karypis/parbook/`
(no errata page at present)

Status: I emailed George Karypis with an initial version of this errata list on 2003-06-03, and he acknowledged my email on the same day. I've made a few corrections to the list since then.

Errata

Page 6:

In the last paragraph, it should say "This section spans Chapters 2 through 5 of this book" (rather than "2 through 4").

Page 46:

Table 2.2 looks a bit iffy. How can the arc connectivity be $>$ the bisection width? Surely any bisection is splitting the nodes into two groups? Also, the cost looks wrong for Omega Network and Dynamic Tree. Surely it's $p(\log p + 1)$ for an Omega network (see figure 2.12 on page 38), and for a Dynamic Tree it's 2 * the number of switching nodes $= 2 \times (p - 1)$

Alternately, if it's only "external links" that count, then it's $2p$ for the Omega Network, and p for the Dynamic Tree, which still doesn't match what's listed.

Or if the cost columns actually means "number of switching nodes" rather than "number of links" (which would be consistent with the description on page 45), then that's an error in the table heading, and the Omega Network is still wrong - it should be $(p/2)\log p$.

Page 66:

It looks like the congestion in figure 2.29 (d) is 5 rather than 6, since that's the maximum number of lines connecting any two nodes (1 and 5).

Pages 67-69:

The mapping on page 68 is wrong - it should be G(i,r)||G(j,s)
Otherwise, in the example on p68, when r=1 and s=2, (0,0) would map to G(0,0)||G(0,1) but G(0,0) is not defined.

Also, although this section refers to wraparound meshes, it only actually works for meshes without wraparound. If the 2x4 mesh in figure 2.31 (b) had wraparound, then node (0,0) would be linked to (0,3), and (1,0) would be linked to (1,3). Similarly, all of the vertical links (0,j) to (1,j) would be doubled up. This would then give us congestion of 2, not 1.

Page 72:

Figure 2.33 (a) doesn't seem to match figure 2.31 (a) (on page 69). Based on that, I think it should be a standard 2D mesh, with links between nodes that only differ in 1 bit.

Page 104:

In figure 3.15, Stage I and Stage II, "D1,2,1" is referred to as "D1,2,2", and "D2,2,1" is referred to as "D2,2,2".

Page 111:

It seems like the static and dynamic decompositions (3rd paragraph) are actually the same thing. The descriptions are phrased slightly differently, but the idea seems to be the same for both.

Page 117:

The abbreviation "cc-UMA" is used without being defined. I guessed that it meant "cache coherent UMA", and verified this by doing a Google search, but I think it would be a good idea to make this explicit in the text.

Page 120:

Line 4 should say "for j := k+1 to n do"
As it is, it doesn't match the description on page 121, and it means that in line 5 you would always set A[k,k] to 1, which would result in a unit diagonal in U as well as L.

Page 121:

This isn't exactly an error, but figure 3.27 isn't very clear. I didn't understand it until I read the description in problem 3.5, which is 23 pages later. Personally, I think that the explanation fits better in the main text of the chapter, rather than in the problem, but at the very least it would be helpful to add a line on page 121 saying "See problem 3.5 (p144) for further details".

Page 122:

In figure 3.28, the references to item (k,j) are incorrect. This corresponds to line 5 of the algorithm, at which point it is item (j,k) which is being updated. In other words, the algorithm goes down through the remaining items in column k, not along through the remaining items in row k.

Page 123:

The final paragraph says: "if we increase α to its upper limit of n/p, then each block is [...] a single element of the matrix in a two-dimensional block-cyclic distribution." This is not the case. Looking at the example in figure 3.30 (b), when n=16, and p=4, then $n/p = 16/4 = 4$, and $\sqrt{p} = 2$. So, if we partition this array "into square blocks of size $\alpha\sqrt{p} \times \alpha\sqrt{p}$", then when $\alpha = n/\sqrt{p}$, the block size is $(4 \times 2) \times (4 \times 2) = 8 \times 8$. This is clearly not a single element of the matrix. More generally, if we assume that $n \geq \sqrt{p}$, then the length of a side of a block will have a minimum value of $\sqrt{p}$ and a maximum value of $n/\sqrt{p}$. So, the only way to get a block size of one element is to have $p = 1$, or $p = n^2$.

Page 157:

Algorithm 4.2 is missing a line - there should be an "endif;" line between 14 and 15, corresponding to line 14 in algorithm 4.1.

Page 158:

Algorithm 4.3 is missing a line - there should be an "endif;" line between 15 and 16, corresponding to line 14 in algorithm 4.1.

(In both of these cases, I'd accept the idea that this line is optional when the "if" statement only contains one compound statement. However, I think it's a good idea to be consistent with the style, rather than sometimes including it and sometimes not, particularly when the algorithms are this similar.)

Page 160:

At the end of the second paragraph, it says "As the program shows, all-to-all broadcast on a mesh applies the linear array procedure twice, once along the rows and once along the columns." This is actually a description of algorithm 4.6 rather than algorithm 4.4, so it belongs at the end of section 4.2.2, on page 161.

Page 163:

In the description for figure 4.10, the last sentence should say: "By the end of the second phase, all nodes get (0,1,2,3,4,5,6,7,8) (that is, a message from each node)." At present, the sequence misses out "8".

Page 175:

In the "Cost Analysis" section, the result of the substitution should be:

$$(t_s + t_w m\sqrt{p}/2)(\sqrt{p} - 1)$$

Similarly, equation 4.8 should then be:

$$T = (2t_s + t_w m\sqrt{p})(\sqrt{p} - 1)$$

I.e. the coefficient of t_w should be $\sqrt{p}$ rather than p in both cases.

Page 186:

The first paragraph should refer to "all-to-all reduction and all-to-all broadcast", to get the desired cost. As described there, it would take twice as long (see page 185).

Page 196:

In figure 5.1, "Interprocessor Communication" should be "Interprocess Communication".

Page 215:

Example 5.14 should refer to equation 5.7 rather than equation 5.9.

Page 220:

In example 5.19, I think that it should be referring to equation 5.22 (or equation 5.7, which is identical), rather than equation 5.2. In this case, it should say:

$$T_p^{cost\_opt} = \log n + 2\log(n/(\log n)) = 3\log n - 2\log\log n$$

Page 226:

In equation 5.36, the top half of the right hand side of the equation should end with " - 1/p", to be consistent with equation 5.35.

As printed, the following simplification is incorrect, but it does work if you make this correction to equation 5.36.

Page 342:

In the final paragraph, the overall parallel run time is $\Theta(\log n)$, not $\Theta(n)$.

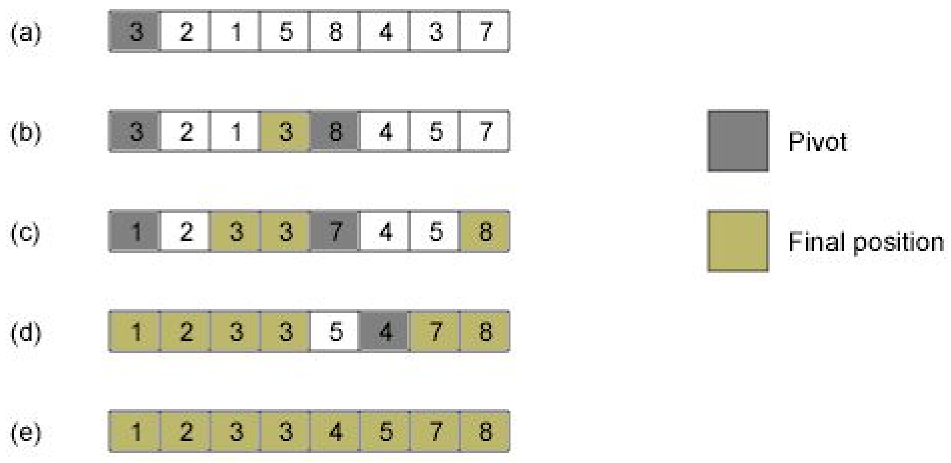
Pages 400-401:

Algorithm 9.5 is wrong - line 14 should be "QUICKSORT (A, q, s-1)". You don't need to include A[s] in that, since you know it is the largest item in that partition. If you leave it as is, then you can wind up in an infinite loop. This is illustrated by figure 9.15, which also contains a mistake, using the printed version of the algorithm.

After step (a), the sequence should be: 3,2,1,3,8,4,5,7. The issue here is that the algorithm tests whether $A[i] \leq x$, rather than < x, so the 3 in position 7 should be moved into the first partition. So, at the end of this stage, q=1, s=4, r=8 (assuming a one-based array).

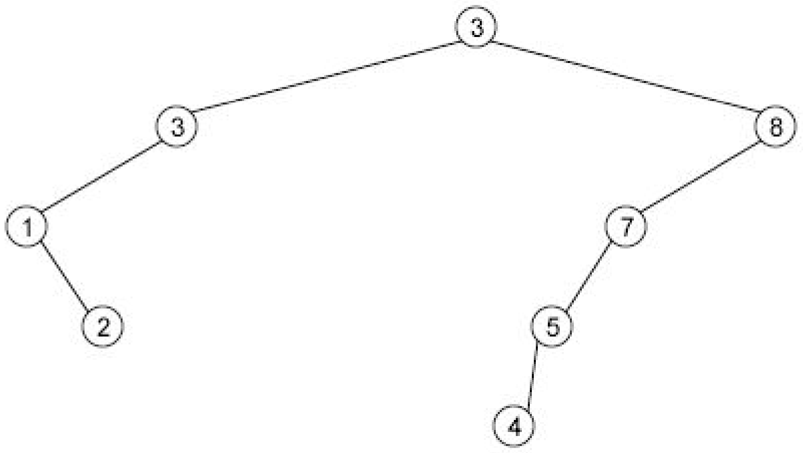
If you then call the function recursively, as printed in algorithm 9.5, you will attempt to sort the list (3,2,1,3), using 3 as the pivot. All of the values here are $\leq$ 3, so the only effect will be to swap the two 3s around in line 13 of the algorithm. At this point, q=1, s=4, r=4. So, it would then call QUICKSORT(A,1,4) again in line 14, which is an identical list, and it will never terminate.

So, after changing algorithm 9.5, figure 9.15 will also need to be updated. It should now say:



Page 403:

The error in figure 9.15 means that figure 9.16 is also wrong. It should be:



In algorithm 9.6, on line 11, it should say "$i \neq parent_i$" rather than "$i < parent_i$". Otherwise, looking at the example in figure 9.15, when $i = 7$, and $parent_i = 1$, then $A[i] = A[parent_i] = 3$. This means that i should go into the left subtree, rather than the right subtree, even though $i > parent_i$. This is supported by the text on page 402 (2nd paragraph of section 9.4.2), which says "elements smaller than or equal to the pivot go to the left subtree".

And on line 16 of algorithm 9.6, it should be "end if", rather than "end for".

Page 405:

Figure 9.17 is consistent with algorithm 9.6 as printed, but if you make the change above then process 6 should be the right child of process 2.

Page 406:

In the first paragraph, it should say "smaller than or equal to the pivot" rather than "smaller than the pivot".

In the third paragraph, it should say "$\lfloor |S|\, p/n + 0.5 \rfloor$" rather than "$\lceil |S|\, p/n + 0.5 \rceil$". Otherwise, when $|S| = n$ (i.e. if the pivot was the largest element in the array), the number of processes would be $\lceil p + 0.5 \rceil = p+1$, which is impossible. This is also illustrated in figure 9.18, which correctly assigns two processors to sort the elements ≤ 7 in the second step. In this case, $|S| = 7$, $n = 20$, and $p = 5$. So, $|S|p/n + 0.5 = 2.25$. $\lfloor 2.25 \rceil = 2$, whereas $\lceil 2.25 \rceil = 3$.

Page 407:

In figure 9.18, the third step is wrong - if the pivot is 11, then 12 should be in L rather than S. The simplest correction is to make the pivot 12 instead.

Page 408:

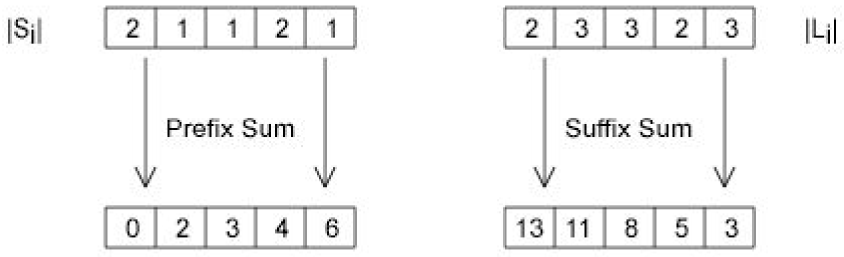
In the third paragraph, the equations for Q_i and R_i are wrong. They should say:

$$Q_i = \sum_{k=0}^{i-1} |S_i|$$

and

$$R_i = \sum_{k=i}^{p-1} |L_i|$$

This also means that in figure 9.19, the prefix sum for $|L_i|$ is wrong. In fact, it should be a "suffix sum", but we can use a slightly modified version of the prefix sum algorithm. Similarly, Q and R are supposed to be of size p (i.e. 5), so the final element shown for Q is redundant. The relevant part of the figure should look like this:



Page 412:

In the second paragraph of section 9.5, "[a,b)" should be "[a,b]".