A Proof that Multiprocessor Scheduling Over Two Processors is NP-Complete

Shaun Meyer

Feb 2012

Abstract

This paper attempts to explain how Multiprocessor Scheduling over two processors is NP-Complete by restricting such that it is equivelent to a known NP-Complete problem, Partition.

0.1 Partition Definition

INSTANCE: A finite set A and a "size" $s(a) \in Z^+$ for each $a \in A$.
QUESTION: Is there a subset $A' \subseteq A$ such that

$$\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a)?$$

Informally, this problem may be viewed as a set A of bowling balls with varying weights. The question is then "Can one divide A into two sets: A' and $A - A'$ such that each set weighs the same?

0.2 Multiprocessor Scheduling Definition

INSTANCE: A finite set A of "tasks," a "length" $l(a) \in Z^+$ for each $a \in A$, a number $m \in Z^+$ of "processors," and a "deadline" $D \in Z^+$.
QUESTION: Is there a partition $A = A_1 \cup A_2 \cup \cdots \cup A_m$ of A into m disjoint sets such that

$$\max \left\{ \sum_{a \in A_i} l(a) : 1 \leq i \leq m \right\} \leq D?$$

Proof(as stated in the text): Restrict to PARTITION by allowing only instances in which $m = 2$ and $D = \frac{1}{2} \sum_{a \in A_i} l(a)$.

Informally this problem may be viewed as a set of tasks of time $l(a)$. The question is asking if one may take a number of tasks and split them into a given number of processors, can these all finish before the Deadline, D, has passed?

An instance of Multiprocessor Scheduling

$A = \{a_1, a_2, a_3\}$	A set of processes.
$A_i = \emptyset$	A_i represents a queue for processor i.
$l(a_1) = 8$	The length of task 1.
$l(a_2) = 2$	The length of task 2.
$l(a_3) = 6$	The length of task 3.
$m = 2$	The number of processors tasks may be assigned to.
$D = 8$	The deadline. We may not process longer than this time.

One satisfactory arrangement is $A_1 = \{a_1\}, A_2 = \{a_2, a_3\}$. To test, we will look at the qualification equation.

$$max\{\sum_{a \in A_i} l(a) : 1 \leq i \leq m\} \leq D$$

The qualification equation should be read as "Is the sum of all processes in each A_i less than or equal to D?" Since A_1 sums to 8 and A_2 sums to 8, $A_1 \leq D$ and $A_2 \leq D$. Thus the problem is satisfied.

0.3 Proof of NP-Completeness

If it is not evident from our instance of Multiprocessor Scheduling, I will mention here that when $m = 2$ and D is determined by $\frac{1}{2}\sum_{a \in A_i} l(a)$ the problem becomes equivalent to Partition.

What remains to be seen is how $m = 1$ and $m > 2$ may be seen as partition.

m=1

The special case, $m = 1$, is not in NP. It is easy to see that a set of tasks may be summed and compared to the deadline in polynomial time.

m>2

However, through this restriction, we have not proved that Multiprocessor Scheduling is NP-Complete for $m > 2$.