Bucket sort

BY GIRI BABU SINNAPOLU
Bucket sort

• 1) what is a Bucket sort?

• **Bucket sort**, or **bin sort**, is a sorting algorithm that works by partitioning an array into a number of buckets.

• Bucket sort works as follows:

• Set up an array of initially empty "buckets".

• **Scatter**: Go over the original array, putting each object in its bucket.

• Sort each non-empty bucket.

• **Gather**: Visit the buckets in order and put all elements back into the original array.
Bucket Sort

Sequential sorting time complexity: $O(n\log(n/m))$.
Works well if the original numbers uniformly distributed across a known interval, say $0$ to $a - 1$. 
Parallel Version of Bucket Sort
Simple approach

Assign one processor for each bucket.

Diagram:
- Unsorted numbers
- p processors
- Buckets
- Sort contents of buckets
- Merge lists
- Sorted numbers
Sequential time

\[ t_s = n + m((n/m)\log(n/m)) = n + n \log(n/m) = O(n \log(n/m)) \]

Parallel Algorithm

Bucket sort can be parallelized by assigning one processor for each bucket - reduces second term in the preceding equation to \((n/p)\log(n/p)\) for \(p\) processors (where \(p = m\)).
Parallel Bucket and Sample Sort

An example of the execution of sample sort on an array with 24 elements on three processes.
Parallel Bucket and Sample Sort

- $p$ processes message-passing computer and $O(p)$ bisection bandwidth.

- **Parallel Sample Sort**
  - Internal sort => $\Theta((n/p)\log (n/p))$
  - Selection of $p-1$ samples => $\Theta(p)$
  - Send $p-1$ elements to $P_0$ (gather) => $\Theta(p^2)$
  - Internal sort of $p(p-1)$ elements at $P_0$ => $\Theta(p^2 \log p)$
  - Select $p-1$ splitters at $P_0$ => $\Theta(p)$
– Broadcast p-1 splitters => Θ(p log p)

– Each process partitions its blocks into p sub-blocks, one for each bucket => p-1 binary searches => Θ(p log (n/p))

– Each process sends sub-blocks to the appropriate processes (if distribution is close to uniform then size of sub-blocks is approx n/p² and assume Hypercube) (ts + twmp/2)log p => Θ((n/p)log p)

• Tp = Θ((n/p)log (n/p)) + Θ(p 2 log p) + Θ(p log (n/p)) + Θ((n/p)log p)
Algorithm

BUCKET-SORT(A)
1. let B[0..n - 1] be a new array
2. $n = A\. length$
3. for $i = 0 \text{ to } n - 1$
   4. make B[i] an empty list
5. for $i = 1 \text{ to } n$
   6. insert A[i] into list B[[nA[i]]]
7. for $i = 0 \text{ to } n - 1$
   8. sort list B[i] with insertion sort
9. concatenate the lists B[0], B[1], \ldots, B[n - 1] together in order
Hence $T_p$,

$$T_p = \Theta\left(\frac{n}{p} \log \frac{n}{p}\right) + \Theta\left(p^2 \log p\right) + \Theta\left(p \log \frac{n}{p}\right) + \Theta\left(n/p\right) + O(p \log p).$$

the isoefficiency function is $\Theta(p^3 \log p)$
waveform
Thank you

Q & A