Algorithms

Supercomputing Challenge Kickoff

Oct 10th, 2015

Bob Robey

LA-UR-15-27821

What is an Algorithm?

- A process to produce a desired outcome
- Analogy – like a recipe is to cooking
- Pre-dates computing
- Responsible for more performance gain than hardware improvements
- Something a human mind is better at discovering than a computer
- First computer algorithm – Ada Lovelace
- Typically your 2nd semester programming class

Analysis, Asymptotic notation

- O – Big-Oh, examples O(N), O(N^2), etc.
  - Drop the leading constants
  - Describes the growth in operation count as input size (N) increases
  - Upper limit, less or equal to
  - Verbally – order N or Big-Oh N
- Ω – omega, lower limit;
- Θ – theta, equals
- o – little-o, less than

Search Problems

- Given a smooth function, f(n), how do you find the minima?
- How do you find a dictionary entry in a sorted list?
- You have four workers, how can you use them in the above problems.
  - Work efficient vs Step efficient

Eliminate Duplicates

- You have a list of database entries, but some are duplicates. What algorithm would efficiently eliminate duplicates?
- Developing operations from algorithm building blocks
  - sort
  - uniq
- What is the order of this algorithm?

Sorts

- Quicksort – recursive divide and conquer. Pick a pivot and divide the list into those greater and less than it. Repeat.
- Merge sort – take two sorted arrays and merge into third by comparing leading pointers. Recursively presort the two subarrays.
- Radix sort – sort on the first digit into "buckets" and then recursively sort the buckets on following digits

Trees

Based on bi-section concept. Note that these are comparison based and nominally O(N log N)

- Binary trees -- Think sorting dictionary entries by alphabetical order
- kDtrees – Alternate bisection by x and then y dimensions up to k dimensions.
- Octree or quadtree – breaks up domain into 4/8 quadrants and repeats as needed.

Hashing

- Key-value pairs
- O(N) – linear. No comparisons.
- Basically stores values in a location based on a "hash" of the input key. The hash term originally was slang for the "chopping" of the key to make it into a hash value
- Examples
  - Names – Take the first and second letter and add
  - Computer address – take the upper 2 bits

Hashing Examples

- Let's sort the class by last name
  - Come up to the front of the room in front of the "bin" with the letter of your last name
  - Scan across the front of the room to sort the class
- Let's hash the school names of all students in this class
  - Come up to the board in the "bin" with the letter of your full name
  - Write your name (key) and school (value)
- Can you think of a way to use a hash to eliminate duplicates?

Reproducible sums

- Algorithms are not all about speed
- Let's look at addition for finite precision arithmetic
  - Not associative. What does this mean?
- You have the following numbers:
  - 8.12, 4.19, 0.03, 2.17
  - Your calculator only holds three digits

Finite Precision Arithmetic

Forward Order	Reverse Order
8.12+4.19 = 12.31	2.14+0.03=2.17
12.3+0.03 = 12.33	2.17+4.19=6.36
12.3+2.14 = 14.44	6.36+8.12=14.48
14.4	14.5

For summing large datasets, this error can grow large.
How would you do this operation to fix this problem?

Possible solutions:

- Sort data into ascending order – doesn't scale
- Carry storage for remainder terms
- Pair-wise addition

Sample code at http://www.github.com/losalamos/GlobalSums

Advanced Task
Inclusive Prefix Sum

- Assigning lockers in alphabetical order -- in the class sorting algorithm, each student find the sum of all students with name entries that come before you plus 1 for you. This is needed to know where your assigned locker number will be.
- How would you do this in parallel?

More Examples of Algorithms

- Supermarket checkout lines
- Shortest path
- School registration
- Class scheduling
- Traveling salesman
- Network routing
- Partitioning
- Greedy algorithms
- Hill climbing
- Simulated Annealing