# Chapter 3: Parallel Algorithmic Structures I

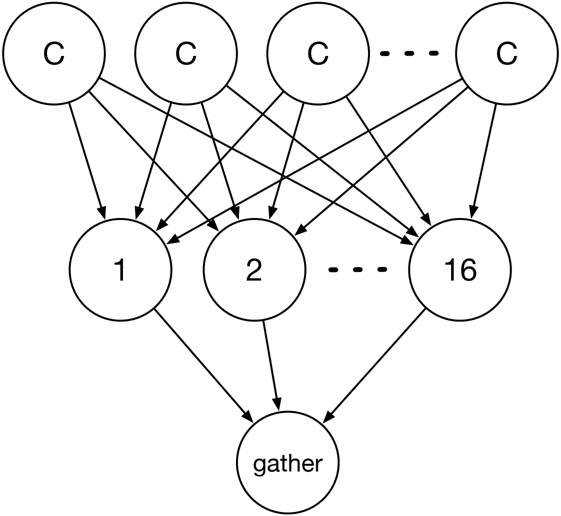
Elements of Parallel Computing

Eric Aubanel

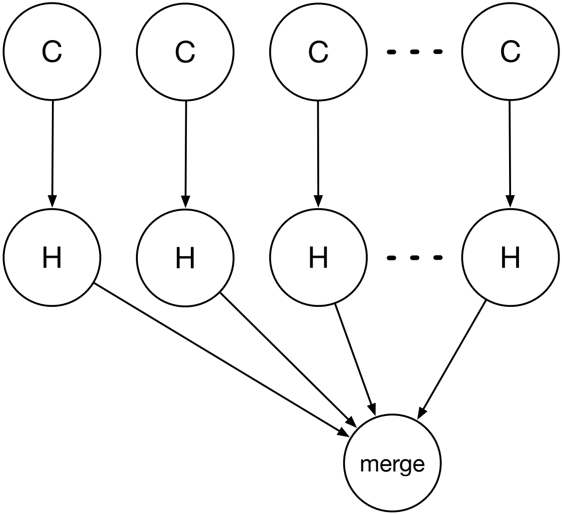
# Histogram Example

- Large collection of short messages from social network
- Want to produce histogram showing distribution of complexity of messages
- Assumptions:
  - complexity estimated as fraction of letters in alphabet that occur in a message,
  - histogram contains 16 bins covering range of complexities from 0 to 1

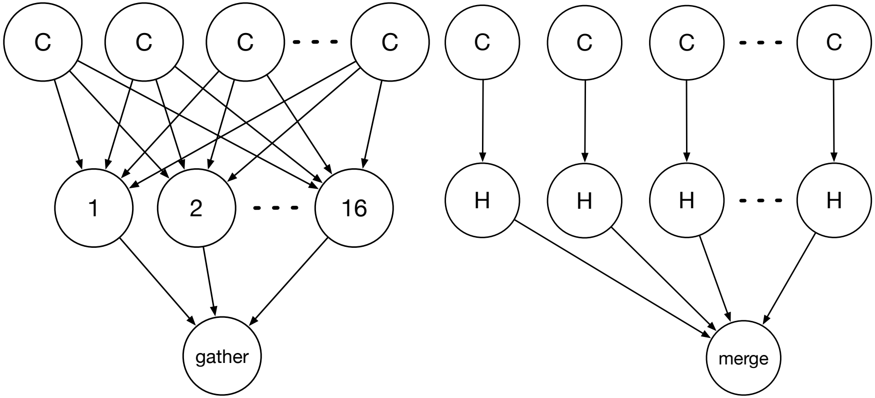
# Histogram Decomposition #1



# Histogram Decomposition #2



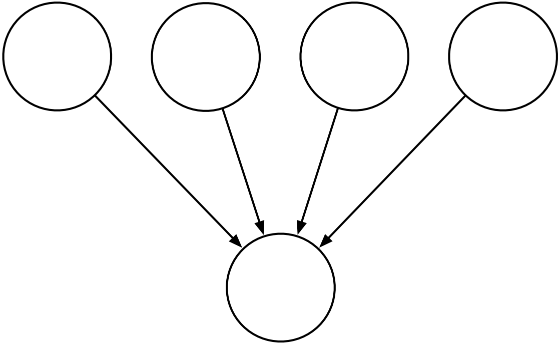
# Histogram Decompositions



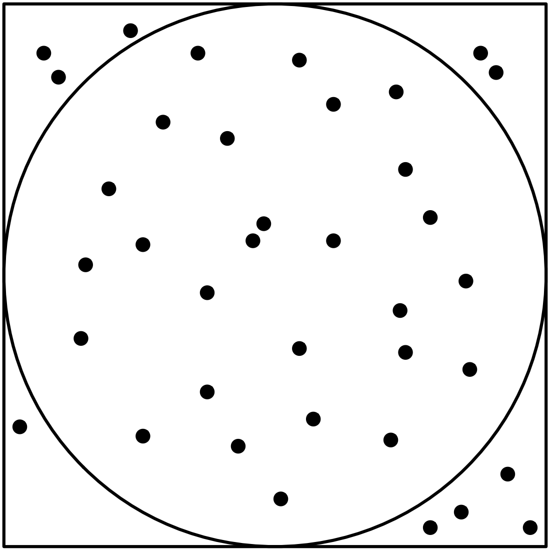
# Guidelines for Parallel Algorithm Design

1. Postpone consideration of the details of the computational platform until after the decomposition phase.
2. Create many independent tasks, whose number increases with the problem size.

# Embarrassingly Parallel

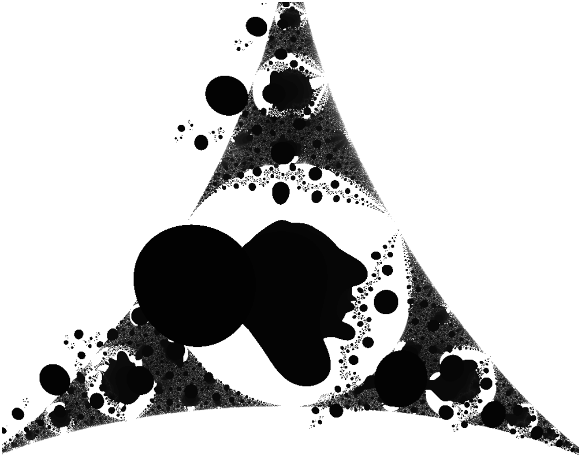


# Monte Carlo estimation of $\pi$



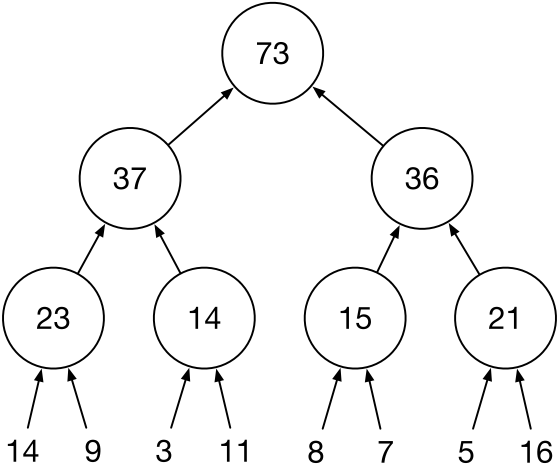
# Generalized Fractal

- Recurrence reliation: $z \leftarrow z^{\alpha} + c$
  - $\alpha$ real and $z$ complex
  - initial value $(0, 0)$ for $\alpha > 0$ and $(1, 1)$ for $\alpha < 0$
  - Coordinates of image are real and imaginary parts of $c$
- Number of iterations to diverge mapped to a colour
  - grayscale: white don't diverge, darker the gray the earlier the divergence



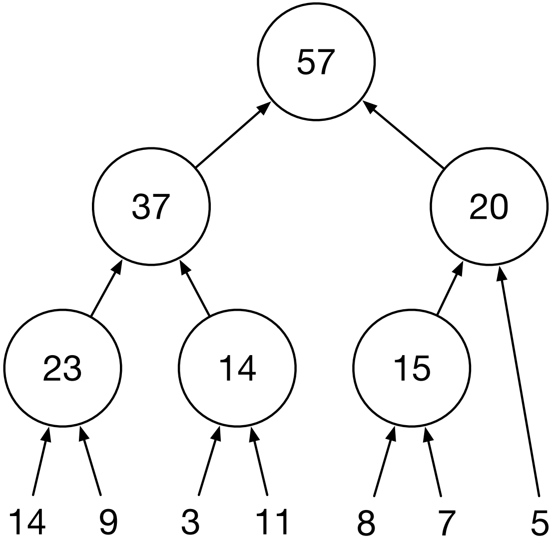
Find the workload imbalance

# Reduction

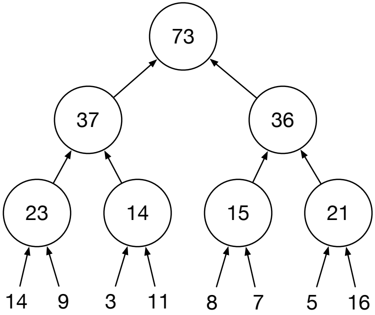


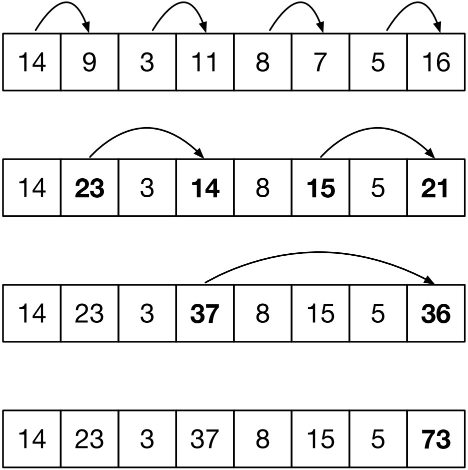

 

# Power of 2 Not Required

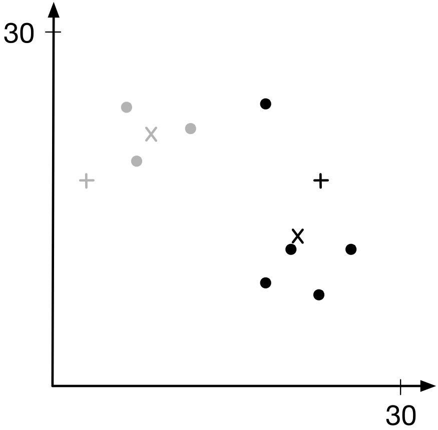


# Two Views of Reduction



# K-means Clustering




$+$: initial guess, $\times$: centroid of points assigned to each cluster

# K-means Clustering Algorithm

**Input**: array of $n$-dimensional vectors, and the number of clusters ($k$)

**Output**: array *closest* containing assignment of each vector to one of the clusters

```
Assign initial guesses of the k cluster centers cluster[k]
Initialize clusterNew[k] and clusterSize[k] arrays to zero
while not converged do
    foreach vector j do
        find cluster center i with smallest Euclidean distance
        to vector j
        closest[j] ← i
        clusterNew[i] ← clusterNew[i] + vector[j]
        clusterSize[i] ← clusterSize[i] + 1
    end
    foreach cluster center i do
        cluster[i] ← clusterNew[i]/clusterSize[i]
        clusterNew[i] ← 0
        clusterSize[i] ← 0
    end
end
```

# K-means Example

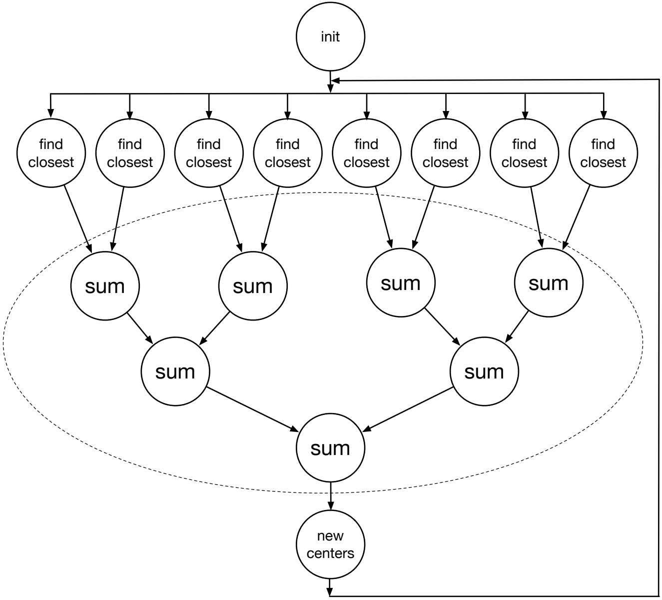
Iteration over vectors:

*clusterNew* array:

$[[6.2, 23.9], 0] + [[7.1, 19.3], 0] + [[11.7, 22.1], 0] + [0, [18.1, 24.2]] + [0, [18.1, 8.9]] + [0, [20.3, 11.8]] + [0, [22.7, 7.9]] + [0, [25.4, 11.8]]$

*clusterSize* array:
$[1, 0]+[1, 0]+[1, 0]+[0, 1]+[0, 1]+[0, 1]+[0, 1]+[0, 1]$

*closest* array: $[0, 0, 0, 1, 1, 1, 1, 1]$


init
find closest
find closest
find closest
find closest
find closest
find closest
find closest
find closest
sum
sum
sum
sum
sum
sum
sum
new centers

# Scan

Scan: binary associative operator to an *ordered* collection of values, produces new collection

- *inclusive*: element $i$ contains the result of the operator applied to the first $i$ elements of the original collection.
- *exclusive*: element $i$ contains the result of the operator applied to the first $i-1$ elements of the original collection.

# Prefix Sum

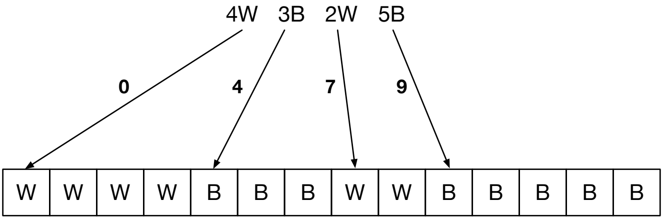
Prefix sum: scan with sum operator

- Inclusive prefix sum of $[2, 16, 9, 7]$: $[2, 18, 27, 34]$.
- Exclusive prefix sum of $[2, 16, 9, 7]$: $[0, 2, 18, 27]$.

# Good Use of Scan: Parallel Write to Array

Run-length encoding: WWWWBBBWWBBBBB represented as 4W3B2W5B

Parallel run-length decoding with 4 tasks: starting point for each write of each task from exclusive prefix sum of $[4, 3, 2, 5]$: $[0, 4, 7, 9]$

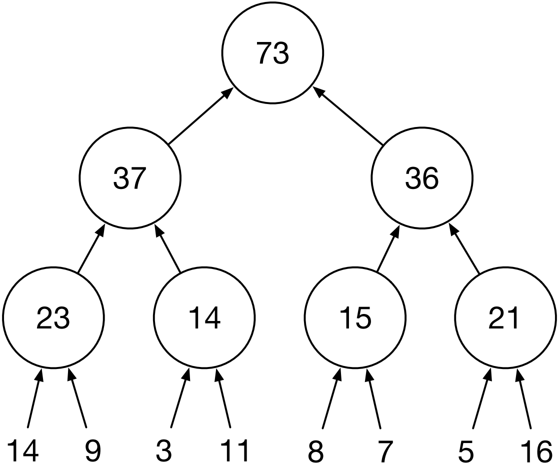


# How to Parallelize Scan?

$sum \leftarrow 0$
**for** $i \leftarrow 0$ *to* $n - 1$ **do**
    $sum \leftarrow sum + a[i]$
    $scan[i] \leftarrow sum$
**end**

# Blelloch Scan (1)

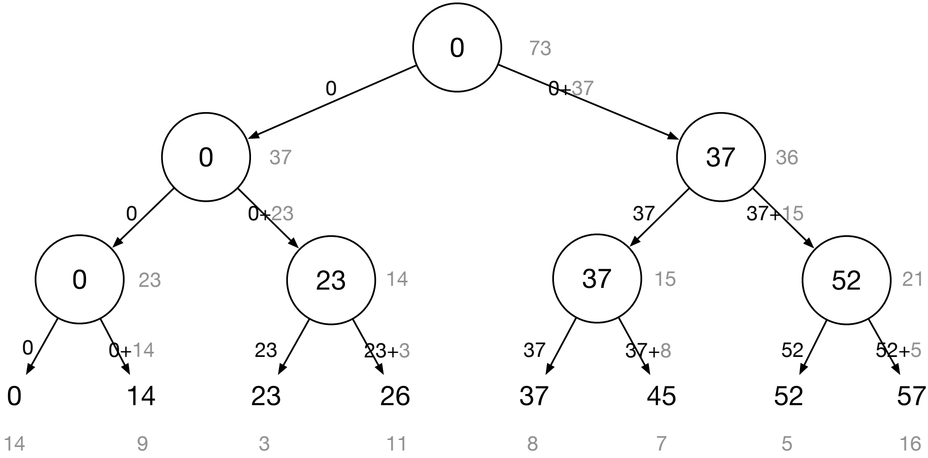
Start with pass up reduction tree:



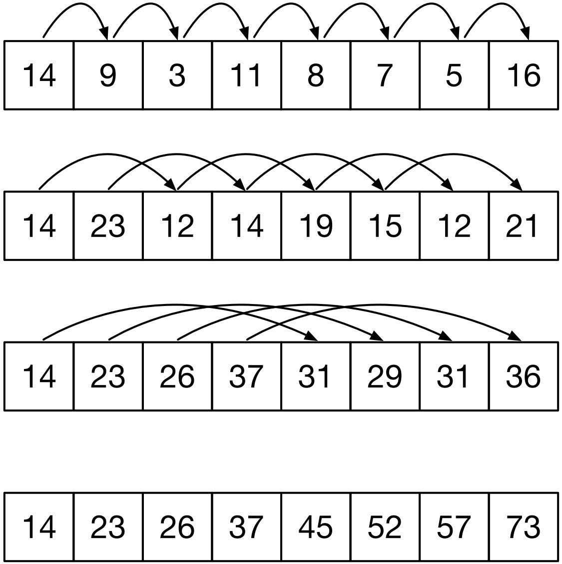
# Blelloch Scan (2)

Back down tree to compute scan:

$$scan[root] \leftarrow 0$$
$$scan[left[v]] \leftarrow scan[v]$$
$$scan[right[v]] \leftarrow scan[v] + reduce[left[v]]$$


0
73
0
0+37
0
37
37
36
0
0+23
37
37+15
0
23
23
14
37
15
52
21
0
0+14
23
23+3
37
37+8
52
52+5
0
14
23
26
37
45
52
57
14
9
3
11
8
7
5
16

# Hillis and Steele Scan



# Task Graph for Hillis and Steele