Simplest..

Load balance: $m$ balls in $n$ bins.
For simplicity: $n$ balls in $n$ bins.
Round robin: load 1!
Centralized! Not so good.
Uniformly at random? Average load 1.
Max load?
$n$. Uh Oh!
Max load with probability $\geq 1 - \delta$?
$\delta = \frac{1}{c}$ for today. $c$ is 1 or 2.

Power of two..

$n$ balls in $n$ bins.
Choose two bins, pick least loaded.
still distributed, but a bit less than not looking.
How much lower?
$\log n/2$? $\sqrt{\log n}$? $O(\log \log n)$?
$O(\log \log n)$ ! ! ! ! ! ! ! ! !
Exponentially better! Old bound is exponential of new bound.

Analysis.

$n/8$ balls in $n$ bins.
Each ball chooses two bins at random.
picks least loaded.
View as graph.
Bin is vertex.
Each ball is edge.
Analysis Intuition:
Add edge, add one to lower endpoint’s “count.”
Max load is max vertices count.
If max count is $k$.
neighbors with counts $\geq k - 1, k - 2, k - 3, \ldots$ and so on!
No cycles and max-load $k \to \geq 2^{k/2}$ nodes in tree.
No connected component of size $X$ and no cycles
--- max load $O(\log X)$.
Will show:
Max conn. comp is $O(\log n)$ w.h.p.
Average induced degree is small. (E.g.: cycle degree 2)
Extend tree intuition.

Balls in bins.

For each of $n$ balls, choose random bin: $X_i$ balls in bin $i$.
$Pr[X_i] \geq k] \leq \sum_{|S|=k} Pr[\text{balls in } S \text{ chooses bin } i]$
From Union Bound: $Pr[|S|\leq k] \leq \sum_{|A|\leq k} Pr[|A|]$
$Pr[\text{balls in } S \text{ chooses bin } i] \leq \binom{n}{k} \left(\frac{1}{n}\right)^k$
Choose $k$, so that $Pr[X_i] \geq k] \leq \frac{1}{2}$.

$Pr[\text{any } X_i \geq k] \leq n \times \frac{1}{2^n} = \frac{1}{2^n}$ 
$\Rightarrow$ max load $\leq k$ w.p. $\geq 1 - \frac{1}{2^n}$

$k! \geq n/2$ for $k = 2\log n$ (Recall $k! \geq \left(\frac{n}{2}\right)^k$).

Lemma: Max load is $\Theta(\log n)$ with probability $\geq 1 - \frac{1}{n}$.
Much better than $n$.
Actually Max load is $\Theta(\log n / \log \log n)$ w.h.p.
(W.h.p. - means with probability at least $1 - O(1/n^2)$ for today.)

Today

Load balancing.
Balls in Bins.
Power of two choices.
Cuckoo hashing.

Analysis Intuition:
View as graph.
Each ball is edge.
Add edge, add one to lower endpoint’s “count.”
Max load is max vertices count.

For each of $n$ balls, choose random bin: $X_i$ balls in bin $i$.

$$\binom{n}{k} \leq \binom{n}{\frac{n}{2}} \leq \left(\frac{ne}{2}\right)^{n/2}$$

$$\binom{n}{k} = \frac{n(n-1) \cdots (n-k+1)}{k!} \geq \frac{2^{n/2}}{\sqrt{\pi n}}$$

$$n(n-1)/(n-k+1) \leq n^k$$

$$k! \geq \left(\frac{1}{2}\right)^k$$

Max connected component is $O(\log n)$ w.h.p.
Average induced degree is small. (E.g.: cycle degree 2)
Extend tree intuition.
Connected Component.

Claim: Component size in $n$ vertex, $\frac{e}{n}$ edge random graph is $O(\log n)$ w/ prob. $\geq 1 - \frac{1}{n}$.

Proof: Size $k$ component, $C$, contains $\geq k - 1$ edges.

$$\Pr[|C| \geq k] \leq \left( \frac{n}{k} \right) \left( \frac{n/B}{k} \right)^{k-1} \left( \frac{k}{n} \right)^{2k-1}$$  \hspace{1cm} (1)

Possible $C$. Which edges. Prob. both endpoints inside $C$.

$$\Pr[|C| \geq k] \leq \frac{n}{k} \left( e \frac{k}{8k} \right)^k \leq \frac{n}{k} \left( \frac{e}{8k} \right)^k \leq \frac{n}{k} (0.93)^2$$

Choose $k = -(c + 1)\log_{93} n$ make probability $\leq 1/n^2$.

Power of two choices.

Max load: $\log X$ where $X$ is max component size.

$X$ is $O(\log n)$ with high probability.

Max load is $O(\log \log n)$.

Not dense.

Induced degree of node on subset, $S$, is degree of internal edges.

Induced degree of nodes in blue subset is 2, not 5!

Claim: Average induced degree on any subset of nodes is $\leq 3$ with probability $\geq 16$.

Proof: Induced degree $\leq 8$ for each internal edges for subset of size $k$.

$$\Pr[\text{dense } S] \leq \left( \frac{n}{k} \right) \left( \frac{n/B}{k} \right)^{k-1} \left( \frac{k}{n} \right)^{2k-1} \leq \left( \frac{e}{8k} \right)^k \leq \left( \frac{1}{2} \right)^k$$

Starts at $1/n^2$, decreasing till $k \leq n/8$ (at least)

$\rightarrow$ Total $O(1/n^2)$.

Cuckoo hashing.

Hashing with two choices: max load $O(\log \log n)$.

Cuckoo hashing:

Array. Two hash functions $h_1, h_2$.

Insert $x$: place in $h_1(x)$ or $h_2(x)$ if space.

Else bump elt $y$ in $h_i(x)$ u.a.r. for $i \in [1, 2]$.

Bump $x$: place $x$ in $h_1(y)$ where $y \neq i$ if space.

Else bump $y$ in $h_i(y)$.

If go too long, Fail. Relhash entire hash table.

Fails if cycle for insert.

$G_i$: event of cycle of length $t$ at a vertex.

$$\Pr[G_i] \leq \left( \frac{m}{t} \right) \left( \frac{n}{t} \right) \left( \frac{1}{n} \right)^{t-1} \leq \left( \frac{e}{8} \right)^i$$  \hspace{1cm} (3)

Probability that an insert hits a cycle of length $t$ $\leq \left( \frac{e}{8} \right)^i$

Relhash every $G(i)$ inserts (if $\leq n/8$ items in table.)

$O(1)$ time on average.

Removal Process!

Random Graph: Component size is $c\log n$ and max-induced degree is $8$ w.h.p.

Process: Remove degree $\leq 16$ nodes and incident edges. Repeat.

Claim: $O(\log X)$ iterations where $X$ is max component size.

For any connected component:

Average induced degree $8 \rightarrow$ half nodes removed in each iteration.

$\rightarrow$ $\log X$ iterations to remove all nodes.

Claim: Max load is $O(\log \log n)$ w.h.p.

Recall edge corresponds to ball.

Height of ball, $h_{i, j}$ is load of bin when it is placed in bin.

Corresponding edge removed in iteration $n$.

Property: $h_i \leq 16$.

Case $i = 1$ - only 16 balls incident to bin $\rightarrow h_i \leq 16$.

Induction: Previous removed edges/ball induce load $\leq 16/(r - 1)$.

$\rightarrow$ $\log_2$ edges/balls this iteration.

$\rightarrow h_i \leq 16_i$.

Sum up

Balls in bins: $O(\log n/\log \log n)$ load.

Power of two: $O(\log \log n)$.

Cuckoo hashing.
See you on Thursday...