Already Crawling at One Month

- Unsearched URL List function:
  - ListIndex = (Mp3Count - HostCount) * Unsearched.size() / (Mp3Count * HostCount) + 1

MP3 Count	Host Count	unsearched.size()	Index
33	3	100	31
66	30	100	2
333	10	1000	97
667	100	1000	9
3333	100	10000	97
100	10	100000	9001
10000	10	100000	9991

She Does More Than Spit Up

- Two tables in SQL database for holding songs:
  - Song – "Artist" found in Artist table
  - SongByContext – "Artist" not in table, but...
- Compare "artist" guesses against table with Aaram Hatchaturyan + 26,510 "artists" harvested from Yahoo!
- Plenty of JavaScript to validate forms
- When searching by keyword, it is highlighted in results
- Uses HttpSession for keeping user "logged in"
- The only site with "Each Link Lovingly Found by Chuck Norris"

Query Processing Heuristic

- Items matching the query exactly and in Song
- Exact matches from SongByContext table
- LIKE 'search %' in Song
- Repeat 3 for SongByContext
- LIKE '%search%' from Song
- Repeat 5 for SongByContext

```
(SELECT Name, Title, URL, Refer-URL, 1 AS Rank
 FROM Song WHERE ...)
      UNION
(SELECT Name, Title, ... 2 AS Rank
 FROM SongByContext WHERE...) ...
      ORDER BY Rank
```

Daddy's So Proud

- After a search of 1.5 hours:
  - Total Number of MP3's 2230
  - Number of MP3's in Table Song 303
  - Number of MP3's in Table SongByContext ... 1927
  - Ratio # in Song / # in SongByContext 1/6
  - Songs in SongByContext with a good Artist guess ..... 386 = 20%

Final Thoughts

- What we learned
  - Lots o' Java
  - Creating a Crawler is easy
  - Creating a great Crawler is much more difficult
  - Parsing all MP3 links correctly is nearly impossible in a chaotic medium.
- In the future
  - Ping the song links to make sure they're there (no ICMP support in Java)
  - Links to a music site if a user is interested in getting more info on an artist or song
  - Create better parsing algorithms
  - Take care of yourself... and each other.