inodes explained

Tutorial 3 - CSCI212

by: Roman Tarnavski

Data Blocks

A file system contains 'blocks' which hold data called inodes. A inode describes a single file in the file system. Typical information described includes, ownership, modification times, size and permissions. We call this information meta-data.

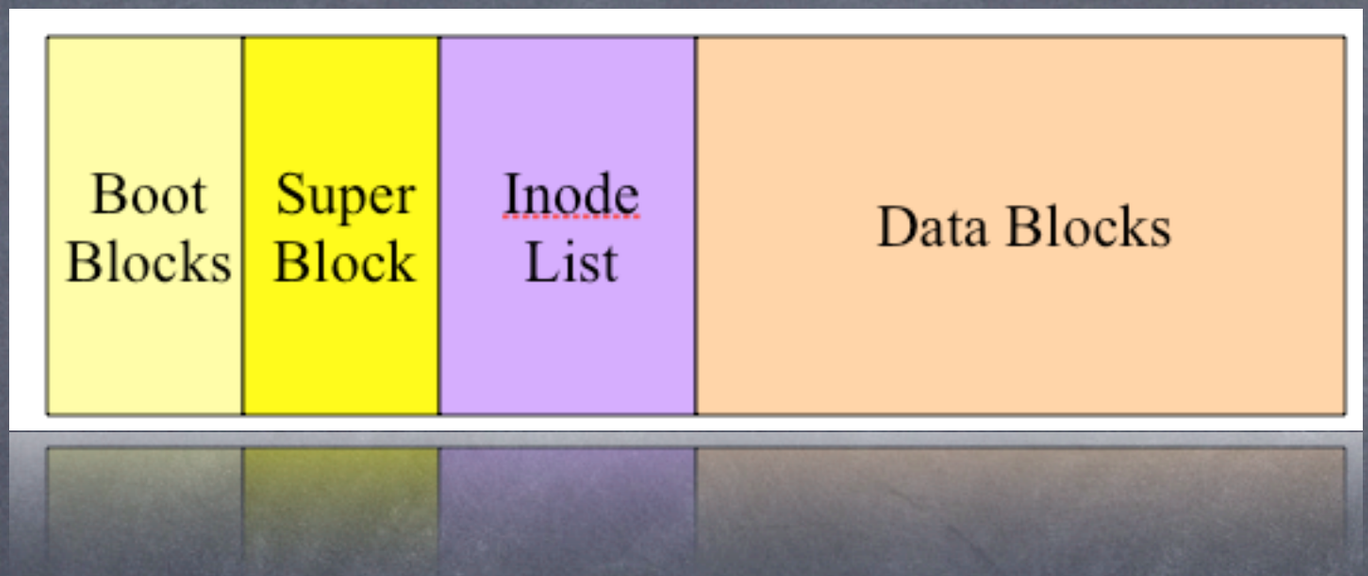
Source: CSCI212 Lecture Notes
by: Daniel Saffioti

Data Blocks

- **Data Blocks**

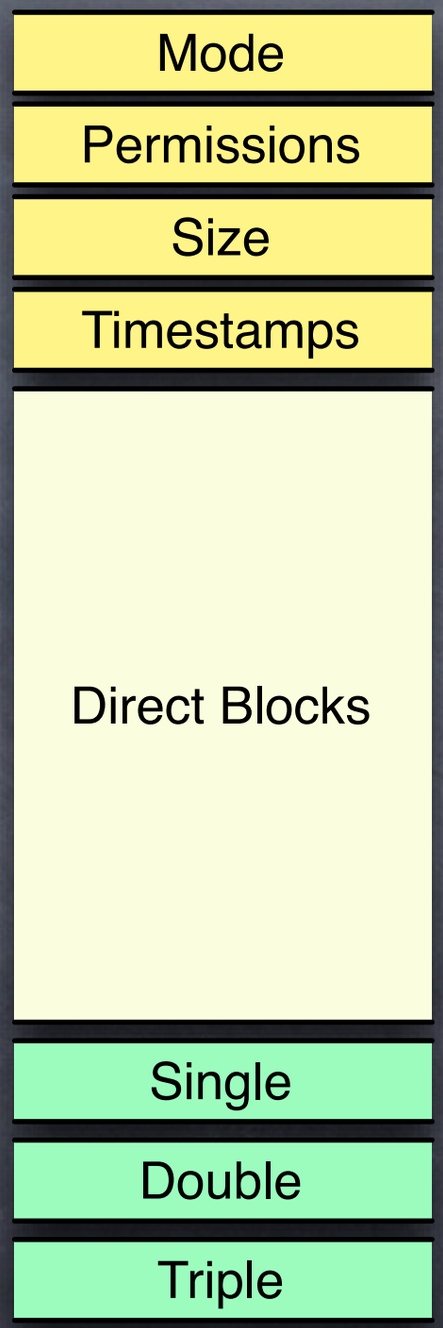
The remainder of the file system (larger proportion) contains data blocks. These store the contents of files.

Source: CSCI212 Lecture Notes
by: Daniel Saffioti

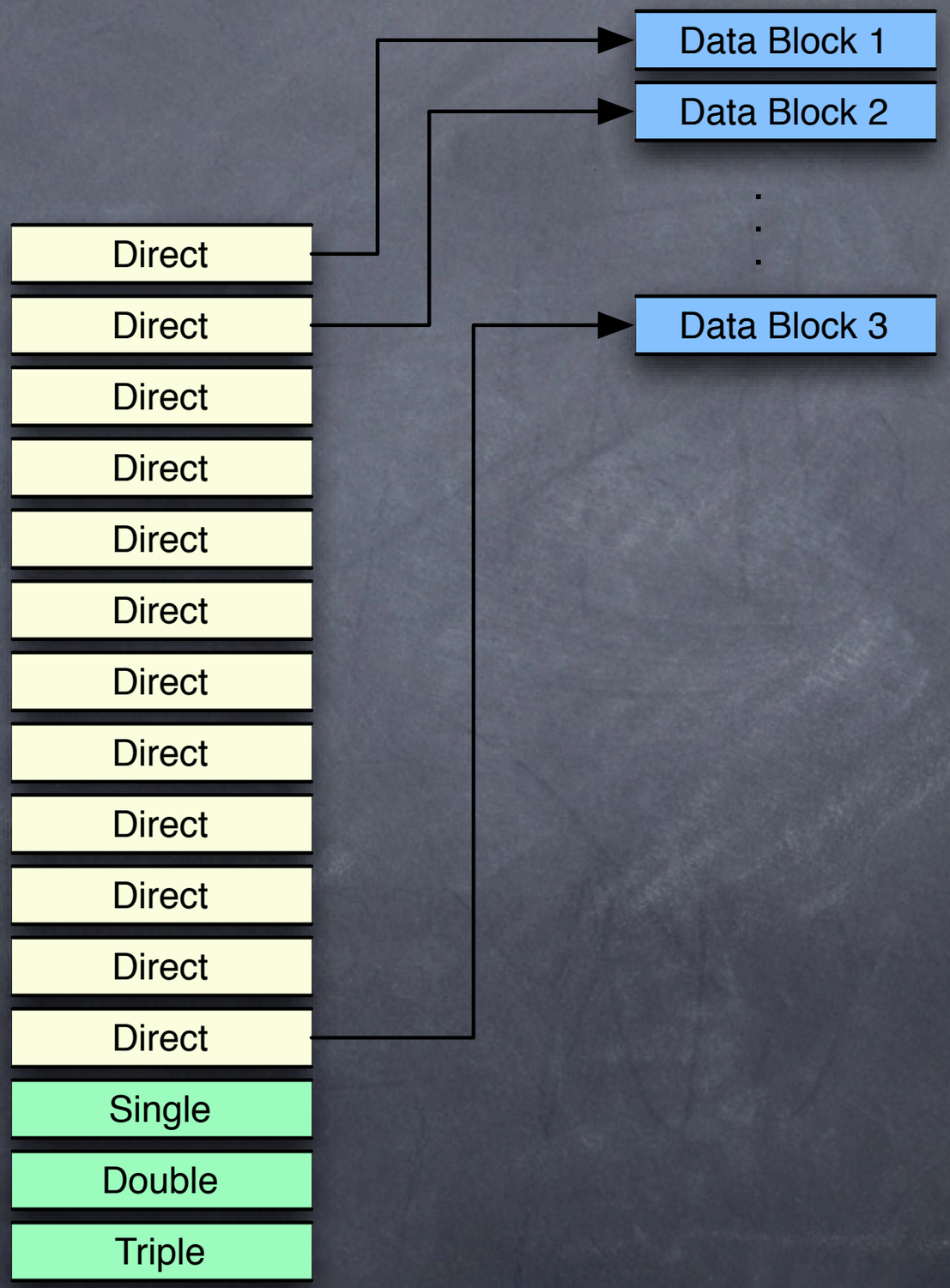


Source: CSCI212 Lecture Notes
by: Daniel Saffioti

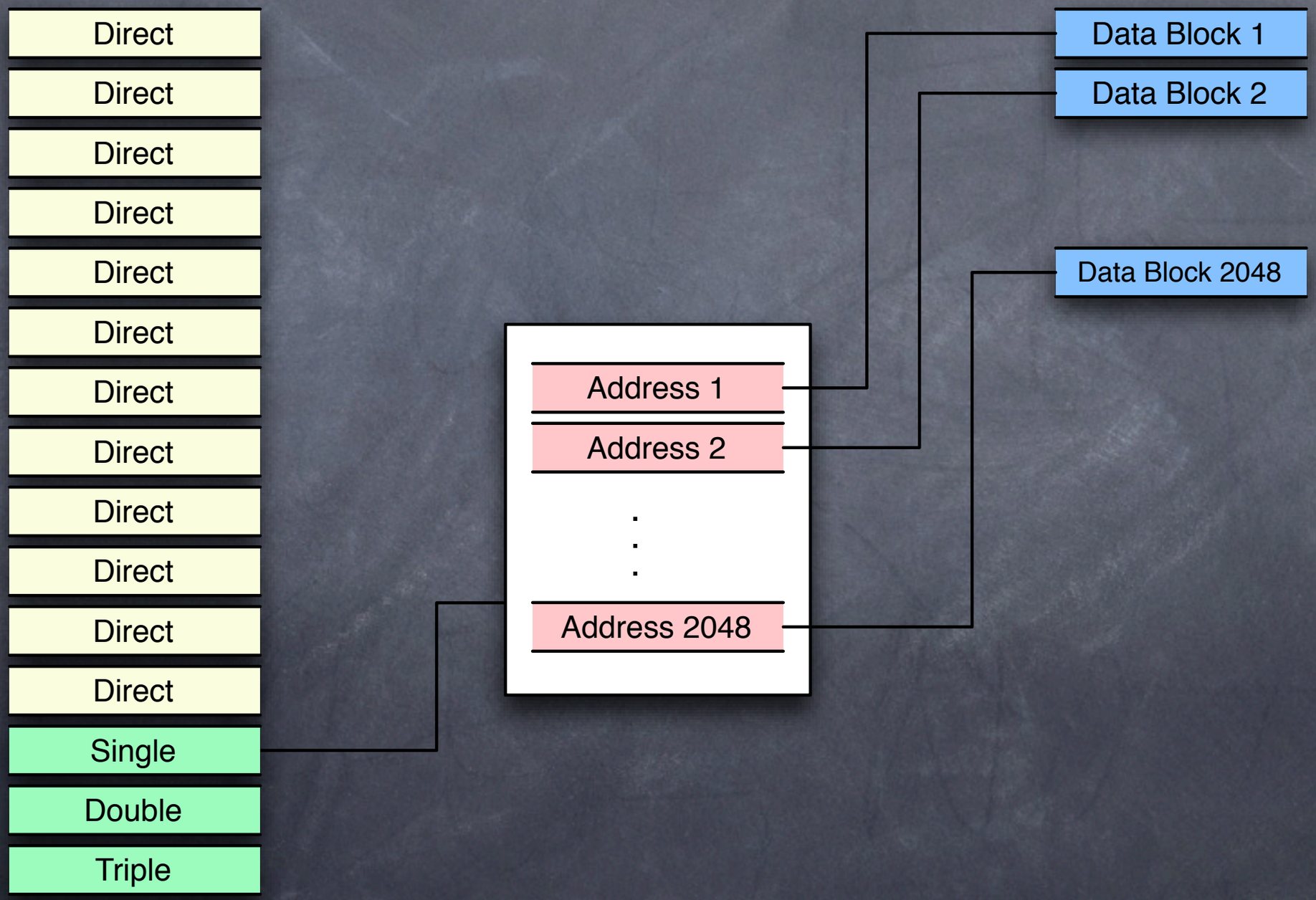
Your typical inode



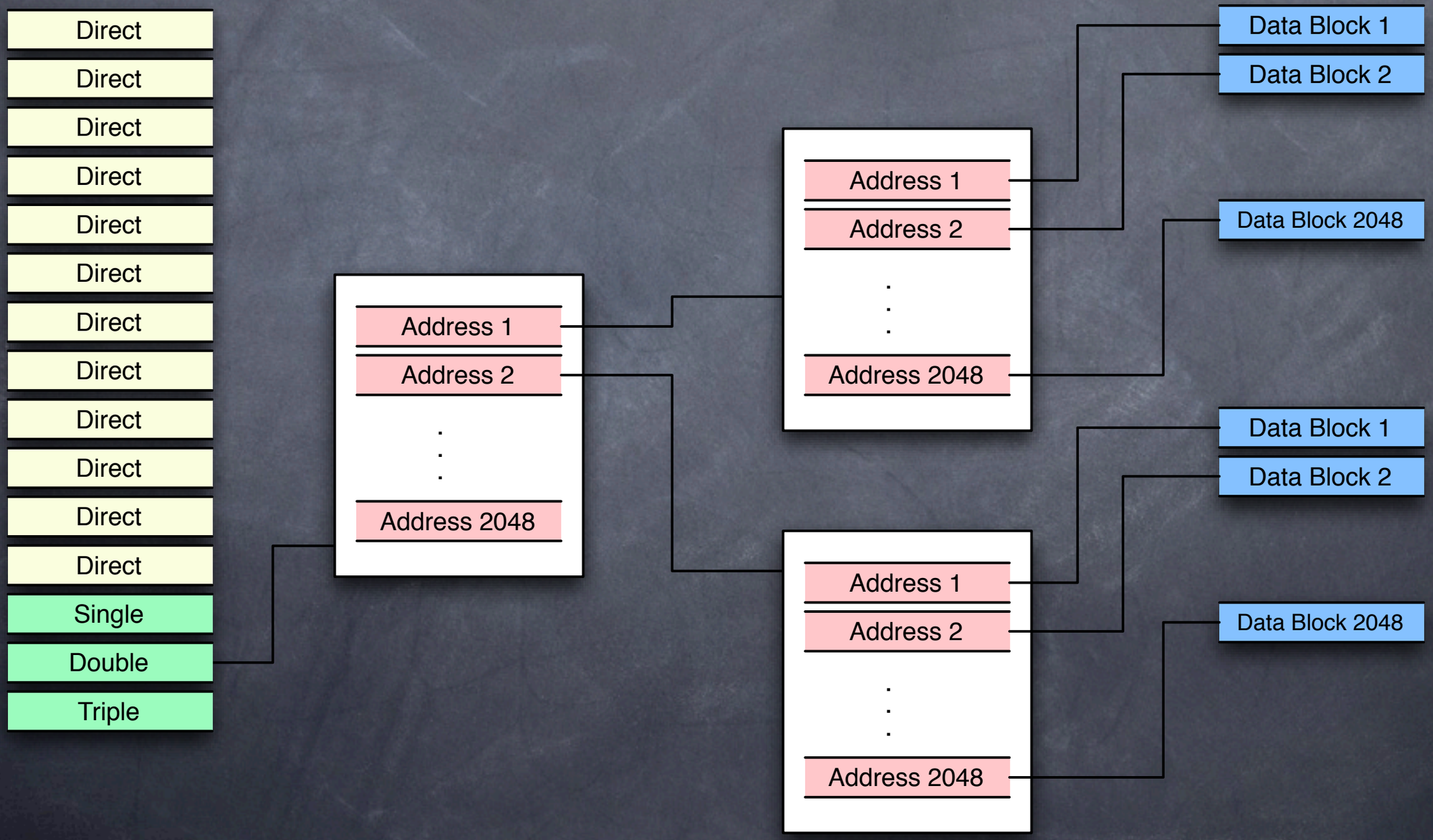
Direct


Data Block 1
Data Block 2
Data Block 3
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Single
Double
Triple

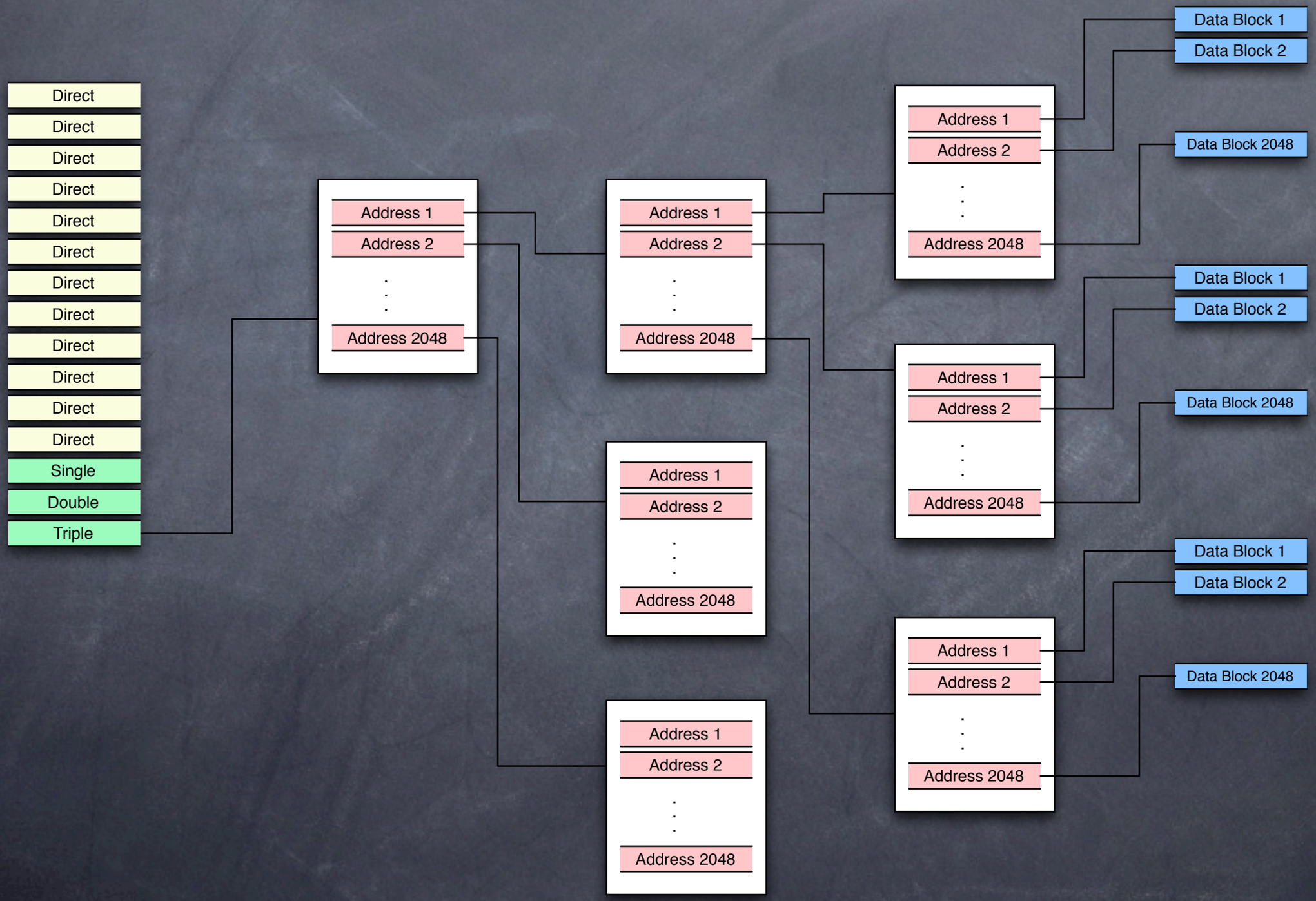
Single


Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Single
Double
Triple
Address 1
Address 2
.
.
.
Address 2048
Data Block 1
Data Block 2
Data Block 2048

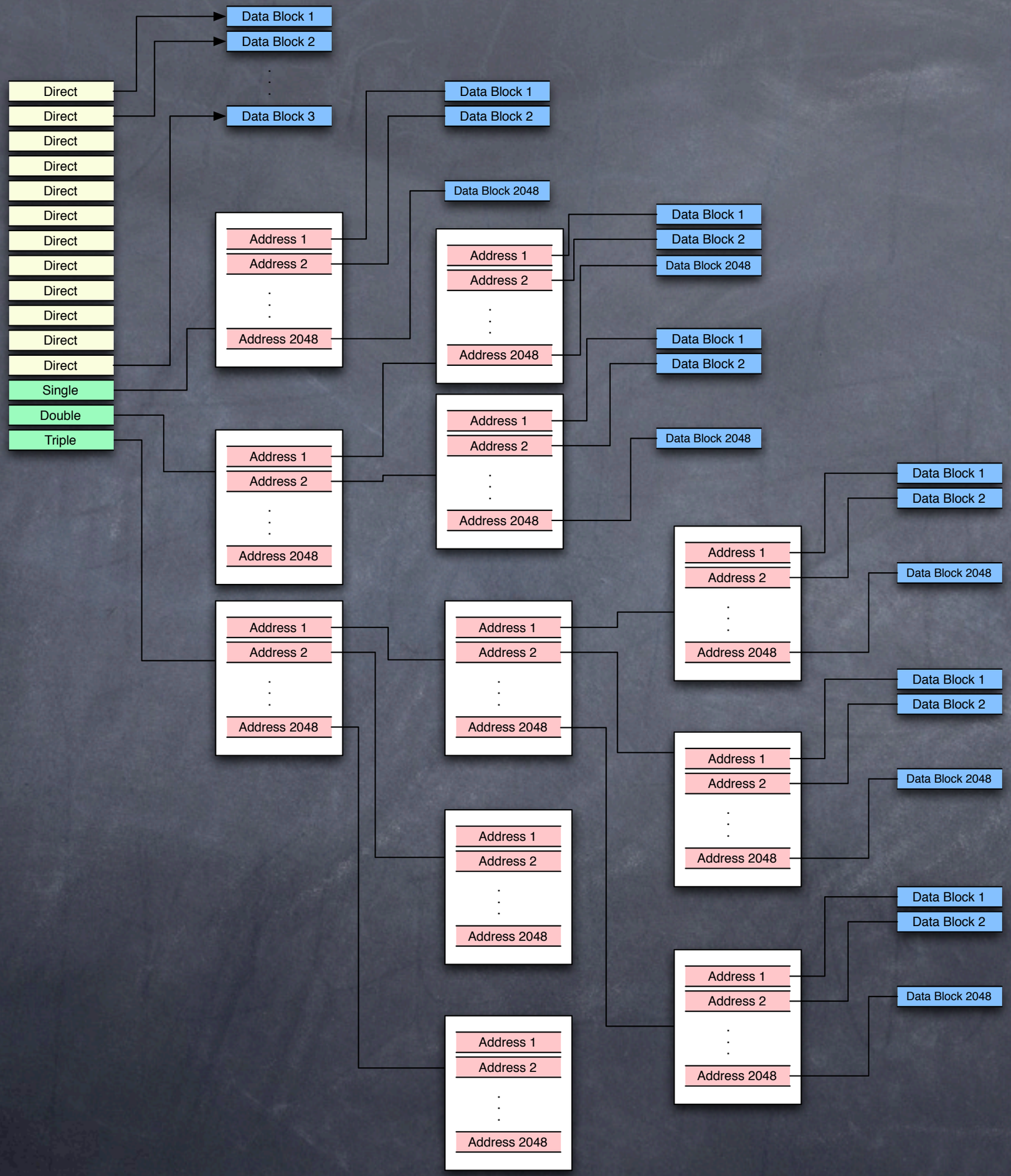
Double


Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Single
Double
Triple
Address 1
Address 2
Address 2048
Address 1
Address 2
Address 2048
Data Block 1
Data Block 2
Data Block 2048
Address 1
Address 2
Address 2048
Data Block 1
Data Block 2
Data Block 2048

Triple


Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Single
Double
Triple
Address 1
Address 2
Address 2048
Address 1
Address 2
Address 2048
Address 1
Address 2
Address 2048
Address 1
Address 2
Address 2048
Address 1
Address 2
Address 2048
Data Block 1
Data Block 2
Data Block 2048
Data Block 1
Data Block 2
Data Block 2048
Data Block 1
Data Block 2
Data Block 2048

All Together?


Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Single
Double
Triple
Data Block 1
Data Block 2
Data Block 3
Address 1
Address 2
Address 2048
Data Block 2048

Maximum Size of Direct Blocks

Maximum Size of Direct Blocks

- Where did the 2048 Addresses come from?

Maximum Size of Direct Blocks

- Where did the 2048 Addresses come from?
- Assuming 32-bit architectures

Maximum Size of Direct Blocks

- Where did the 2048 Addresses come from?
- Assuming 32-bit architectures
  - 32-bit = 4 bytes (32 / 8)

Maximum Size of Direct Blocks

- Where did the 2048 Addresses come from?
- Assuming 32-bit architectures
  - 32-bit = 4 bytes (32 / 8)
  - Thus 8192 [block size] / 4 [bytes per addressable block]

Maximum Size of Direct Blocks

- Where did the 2048 Addresses come from?
- Assuming 32-bit architectures
  - 32-bit = 4 bytes (32 / 8)
  - Thus 8192 [block size] / 4 [bytes per addressable block]
  - = 2048 Addressable Blocks

Questions

- What is the maximum possible file size that can be represented via direct data blocks?

Maximum Size of Direct Blocks

block size

blocks

- = 8192 x 12 = 98,304 bytes = 96Kb

Maximum FileSize using First-Indirect Linking

Maximum FileSize using First-Indirect Linking

- Direct: 12 blocks @ 8192 bytes (12 x 8192)

Maximum FileSize using First-Indirect Linking

- Direct: 12 blocks @ 8192 bytes (12 x 8192)
- =96Kb

Maximum FileSize using First-Indirect Linking

- Direct: 12 blocks @ 8192 bytes (12 x 8192)
  - =96Kb
- Indirect: 2048 @ 8192 (2048 x 8192) = 16,384 Kb = 16,480 [Inclusive of Direct] = 16.09Mb

Maximum using Double Indirect

- 16,875,520 x 2048 addressable blocks
- = 34,561,064,960 bytes
- = 32,960 Mbytes

Maximum Filesize

[direct + single + double + triple]

- 2048 x 32.1875Gb

  = 64.375 Tb

Overhead

Overhead

- Wish to store: 8,437,760 bytes

Overhead

- Wish to store: 8,437,760 bytes
- 8437760 / 8192 [bytes per block]

= 1030 blocks x 4 bytes [per address]

BUT

Overhead

- Wish to store: 8,437,760 bytes
- 8437760 / 8192 [bytes per block]

  = 1030 blocks x 4 bytes [per address]

  BUT
- 12 Blocks are stored directly

Overhead

- Wish to store: 8,437,760 bytes
- 8437760 / 8192 [bytes per block]

  = 1030 blocks x 4 bytes [per address]

  BUT

- 12 Blocks are stored directly
- Hence we only have 1018 blocks

Overhead

- Wish to store: 8,437,760 bytes
- 8437760 / 8192 [bytes per block]

  = 1030 blocks x 4 bytes [per address]

  BUT
- 12 Blocks are stored directly
- Hence we only have 1018 blocks
- 1018 x 4 bytes = 4072 bytes = 3.98Kbytes