

# Low Hanging Fruits

Dzień dobry

# Performance tuning

„Performance tuning is the improvement of system performance.“

# Performance measurement

Instruments

# Hardware capabilities

multithreading, arm64

# Optimization

- **Design** proper architecture
- **Code** using framework capabilities
- **Compilation** LLVM
- **Runtime** ?

# Fixing architecture

cool story, but it takes time

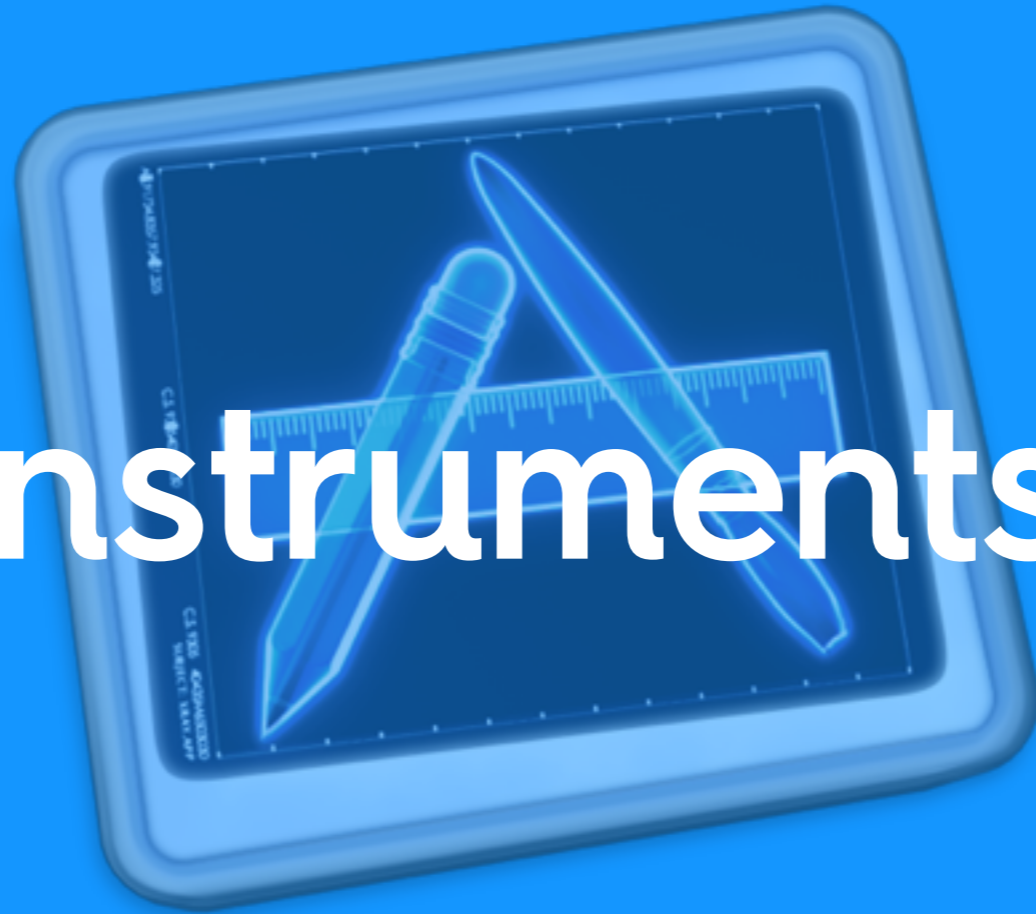
# Optimizing code

simpler and still effective



# Low Hanging Fruits

# Instruments.



# Bottlenecks

- Memory
- CPU
- Resources

Memory

The screenshot shows the Xcode Instruments interface. The 'Leaks' instrument is selected in the Instruments panel. The main window displays a 'Leak Cycles' table with one entry: a simple cycle involving four nodes: Anna, Danny, Jessica, and Jonny. To the right, a graph visualizes this cycle as a vertical chain of nodes with a curved arrow at the top pointing back to the start.

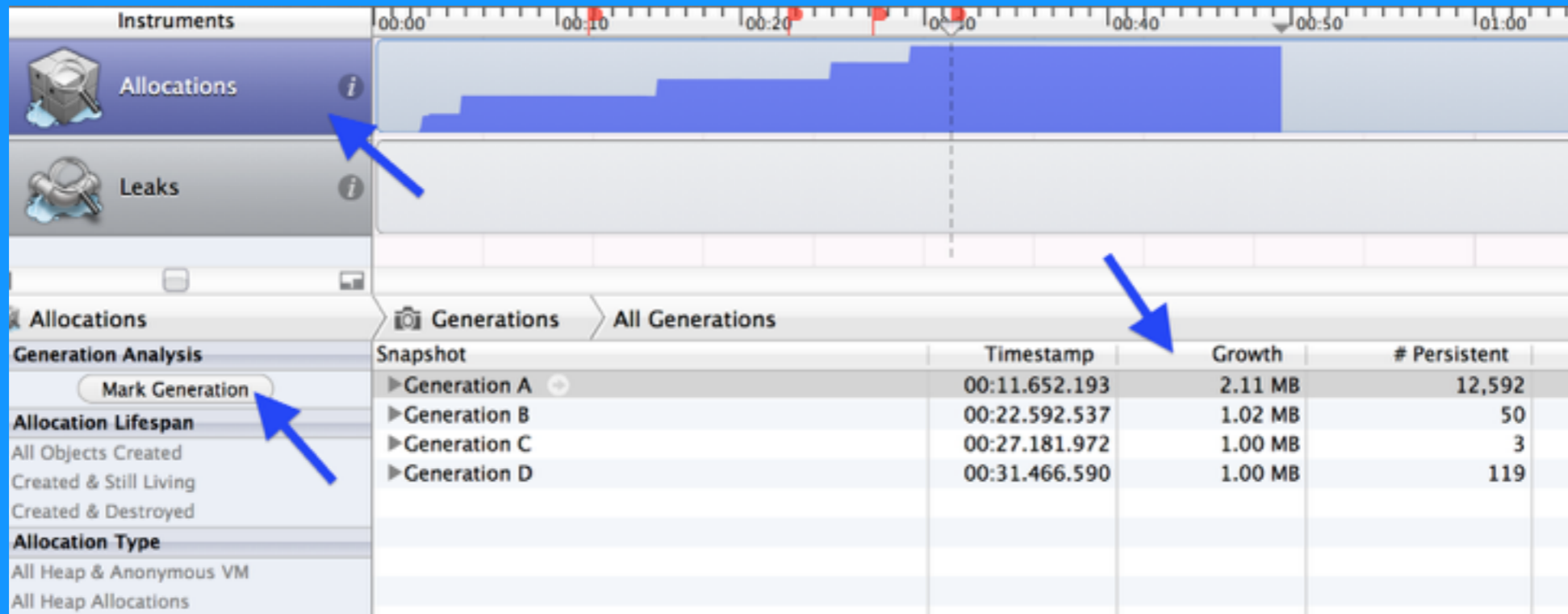
#	Type	Details
1	Anna - 4 nodes	Simple Cycle

Graph visualization:

```
graph TD; Anna -- "Danny* _likesDanny" --> Danny; Danny -- "Jessica* _likesJessica" --> Jessica; Jessica -- "Jonny* _likesJonny" --> Jonny; Jonny -- "Anna* _likesAnna" --> Anna;
```

# Leaks

Pinpointing memory leaks and retain cycles

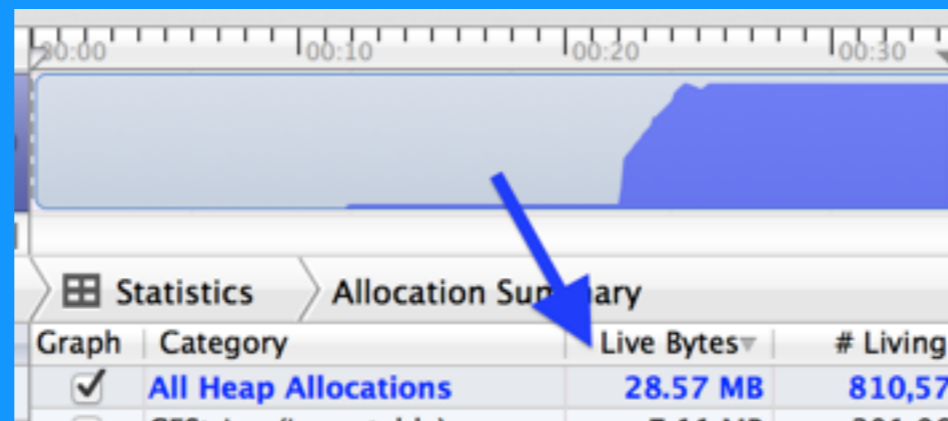


# Allocations

Generations identify objects kept in memory at a given time

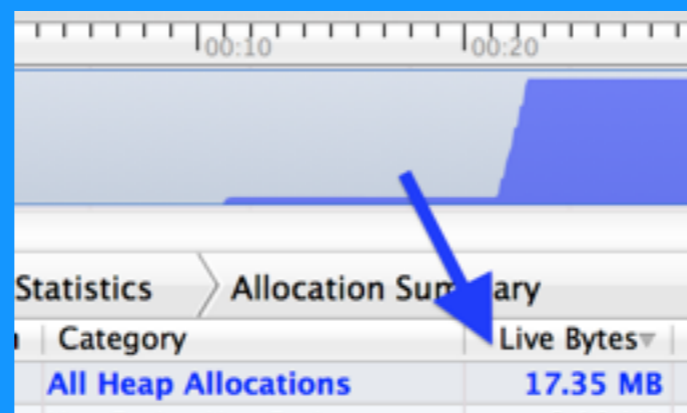
**Memory**  
**vs**  
**Core Data**

```
NSFetchRequest *fr = [[NSFetchRequest alloc] initWithEntityName:@"UserEntity"];
NSArray *userData = [self.context executeFetchRequest:fr error:nil];
```





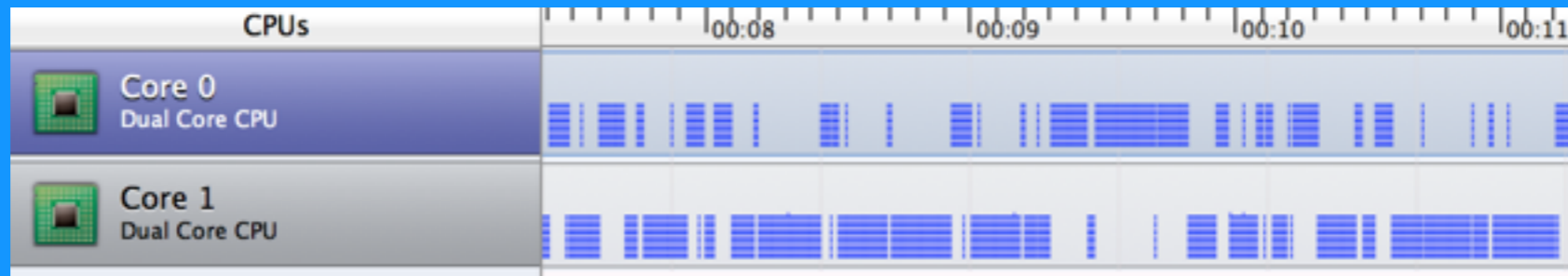
```
NSFetchRequest *fr = [[NSFetchRequest alloc] initWithEntityName:@"UserEntity"];  
fr.propertiesToFetch = @[@"lastName"];  
NSArray *userData = [self.context executeFetchRequest:fr error:nil];
```



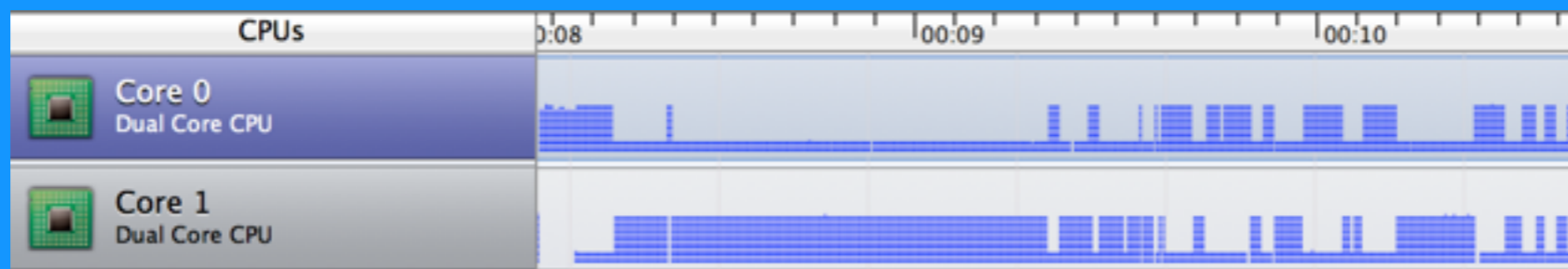
# Core Data Cache

CPU

Non-concurrent:



Concurrent:



@ayoy  
dominik@getbase.com