UNIVERSITY OF SUSSEX

Department of Informatics
MComp (Hons) Computer Science
Academic year 2015/16

Scalable Distributed Computing using Hashing with Async I/O

Michael Sledge

Motivation

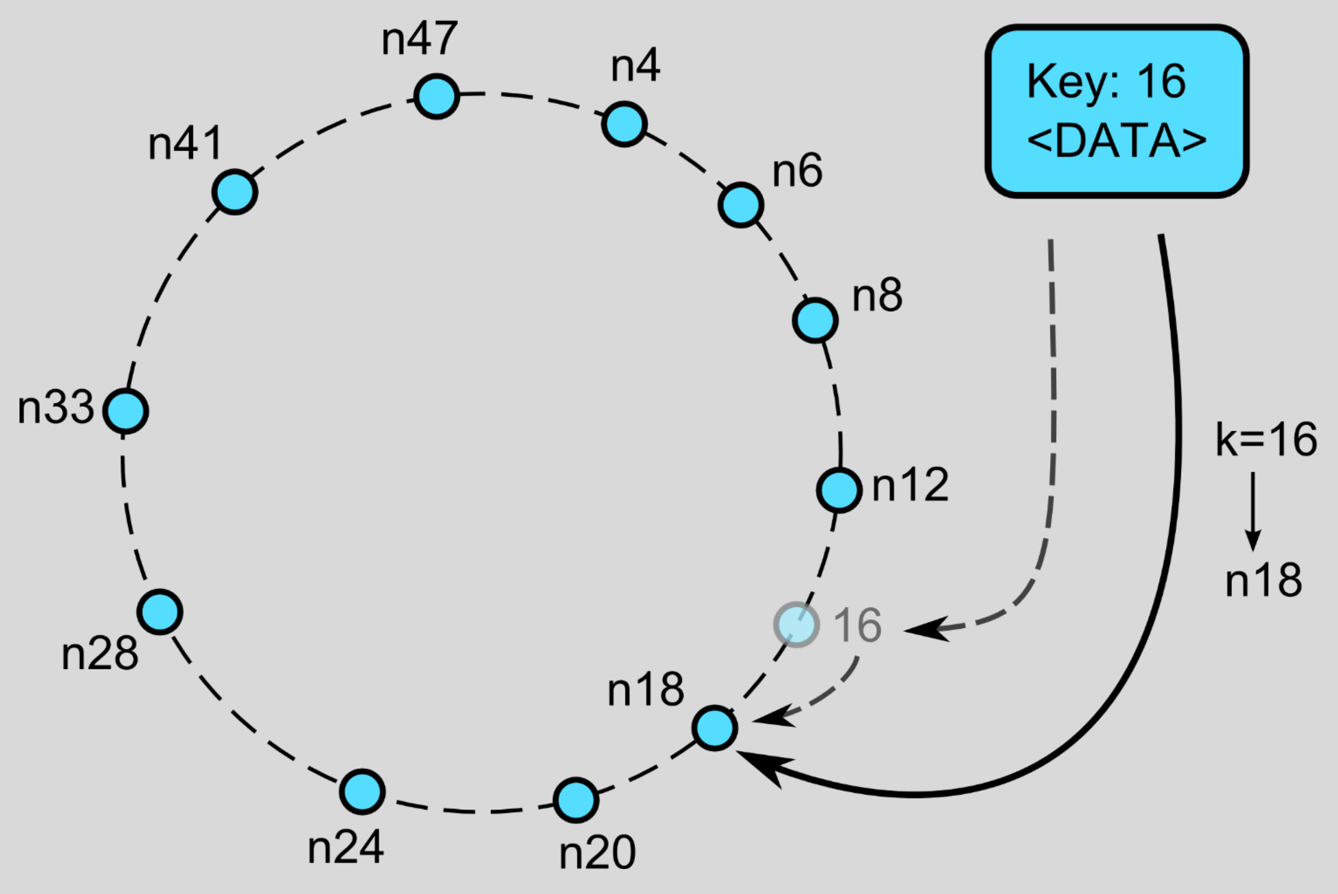
- Distributed computing has many challenges:
  - organising computing nodes;
  - distributing work evenly;
  - Handling changes in node set (failures etc.).

This project uses consistent hashing to create self-organising nodes and distribute workload.

What is it

- A library for creating decentralised peer-to-peer distributed systems
- Creates and maintains a network of nodes
- Accommodates frequent joining and leaving of nodes
- Nodes act like Hashmap ‘buckets’
- Map ‘work’ to nodes using hashes
- Asynchronous network I/O for scalability and performance
- Can be used to build variety of systems

Mapping data to a node



Interface

- Find successor – finds the node a message maps to.
- Send message – sends message to the node it maps to.
- Route message – sends message through nodes to destination

Usage examples

- Can be used to implement DHT (Distributed Hash Table)
- Distributed storage
- Web/Database caching (like Memcached)
- Peer-to-peer communication
- Distributed parallel processing
- Publish–subscribe Network

Specifications

- Based on Chord protocol
- Written in C
- Simple interface
- Scalable and performant

Uses Libevent

- Library for asynchronous I/O using callbacks
- Allows for high scalability with non-blocking I/O