Applying Prolog to Develop Distributed Systems

Nuno Lopes, Juan Navarro,
Andrey Rybalchenko, Atul Singh

Motivation

- Software model checker implemented in Prolog: ARMC
- Some (real-world) inputs took 2 weeks to run
- So we wanted a distributed version of ARMC

- Challenged to implement a BFT protocol concisely
- DS group tried P2 (Datalog) before without success (expressiveness and efficiency issues)

Example: Ping

```
init :-
    neighbor(N),
    my_address(Me),
    send(N, ping(Me)).
```

```
:- event ping/1.

ping(Addr) :-
    send(Addr, pong).
```

```
:- event pong/0.

pong :-
    print('alive!').
```

- **Network-driven query execution**
- **Messages = Prolog Tuples**

Example: Recurrent Ping

```
init :-
    alarm(ping_all, 5000, true).


:- alarm ping_all/0.

ping_all :-
    my_address(Me),
    sendall(N, neighbor(N), ping(Me)).
```

DAHL Interface

```
:- event PredSpec1, ..., PredSpecN.
  e.g.,
    :- event q/2.
    q(X, Y) :- Body.
```

Important for:

- Security
- Program Analysis

DAHL Interface

:- event PredSpec1, ..., PredSpecN.

e.g.,

```
:- event q/2.
q(X, Y) :- Body.
```

:- alarm PredSpec1, ..., PredSpecN.

e.g.,

```
:- alarm ping/1.
q(X) :-
    alarm(ping(X), 1000).
ping(Addr) :-
    send(Addr, ping).
```

Triggers local events in reactive systems

DAHL Predicates

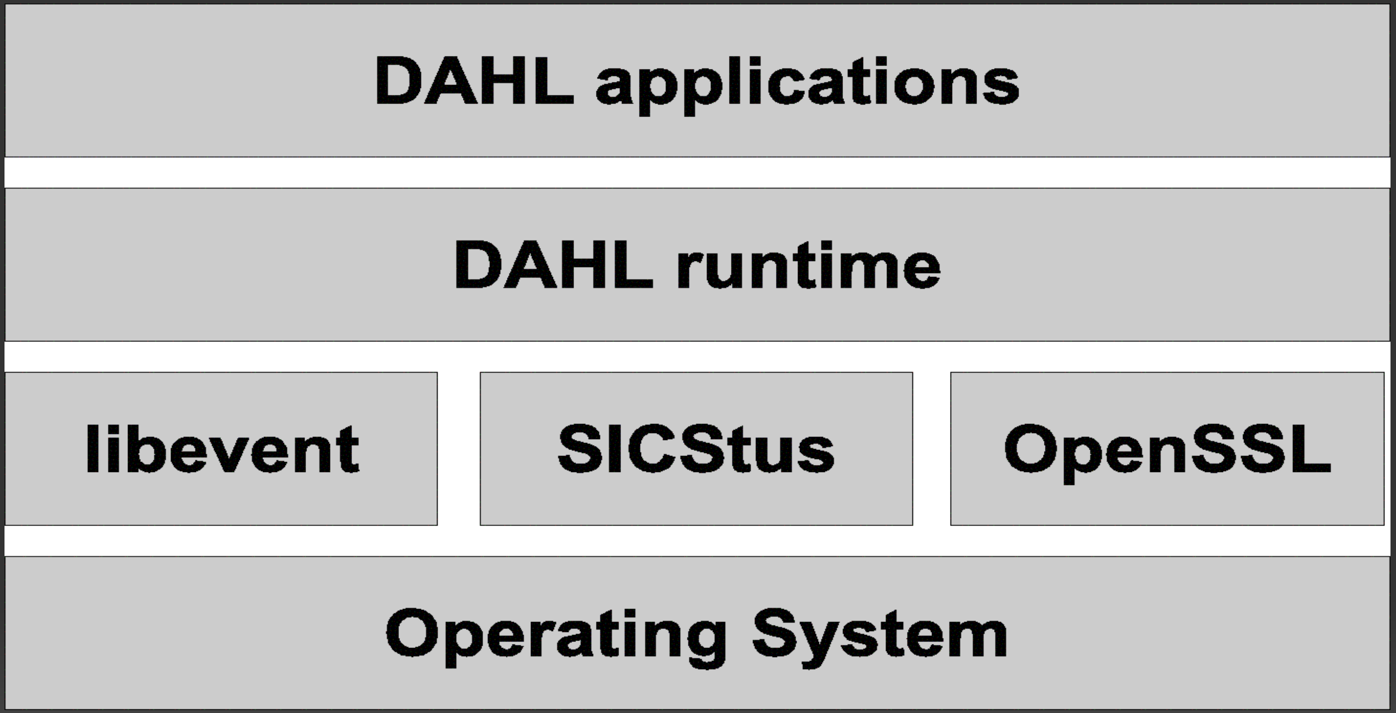
- send(Address, Message)
- sendall(Address, Generator, Message)
- my_address(Address)

- alarm(Message, MSecs)
- alarm(Message, MSecs, Recur)

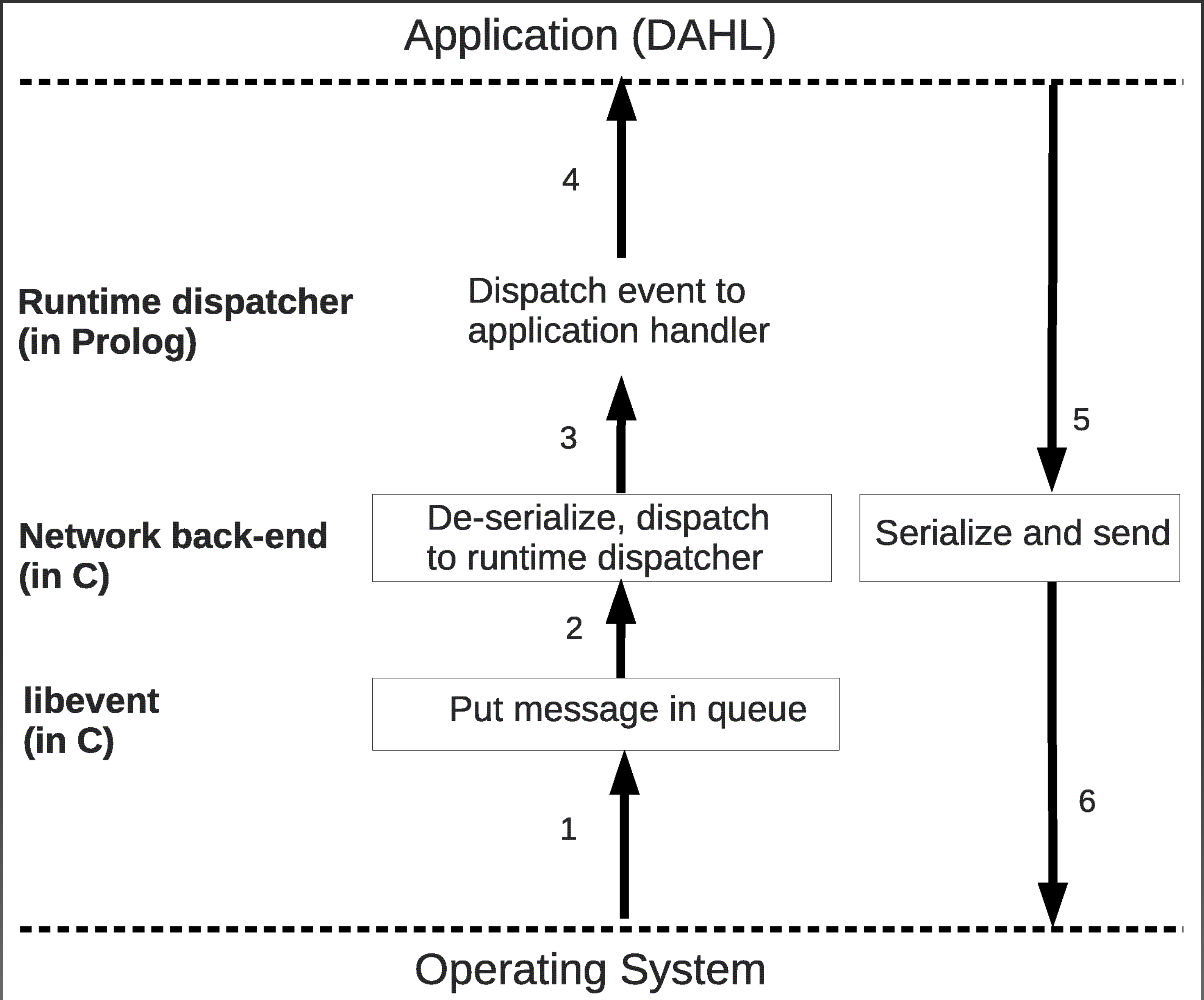
- send_signed(Address, Message)
- signed_by(Address, Signature)
- signed_by(Address)
- signed/0

Implementation

DAHL Software Stack



Event Handling Mechanism



Application (DAHL)

4

Runtime dispatcher (in Prolog)

Dispatch event to application handler

3

5

Network back-end (in C)

De-serialize, dispatch to runtime dispatcher

Serialize and send

2

libevent (in C)

Put message in queue

1

6

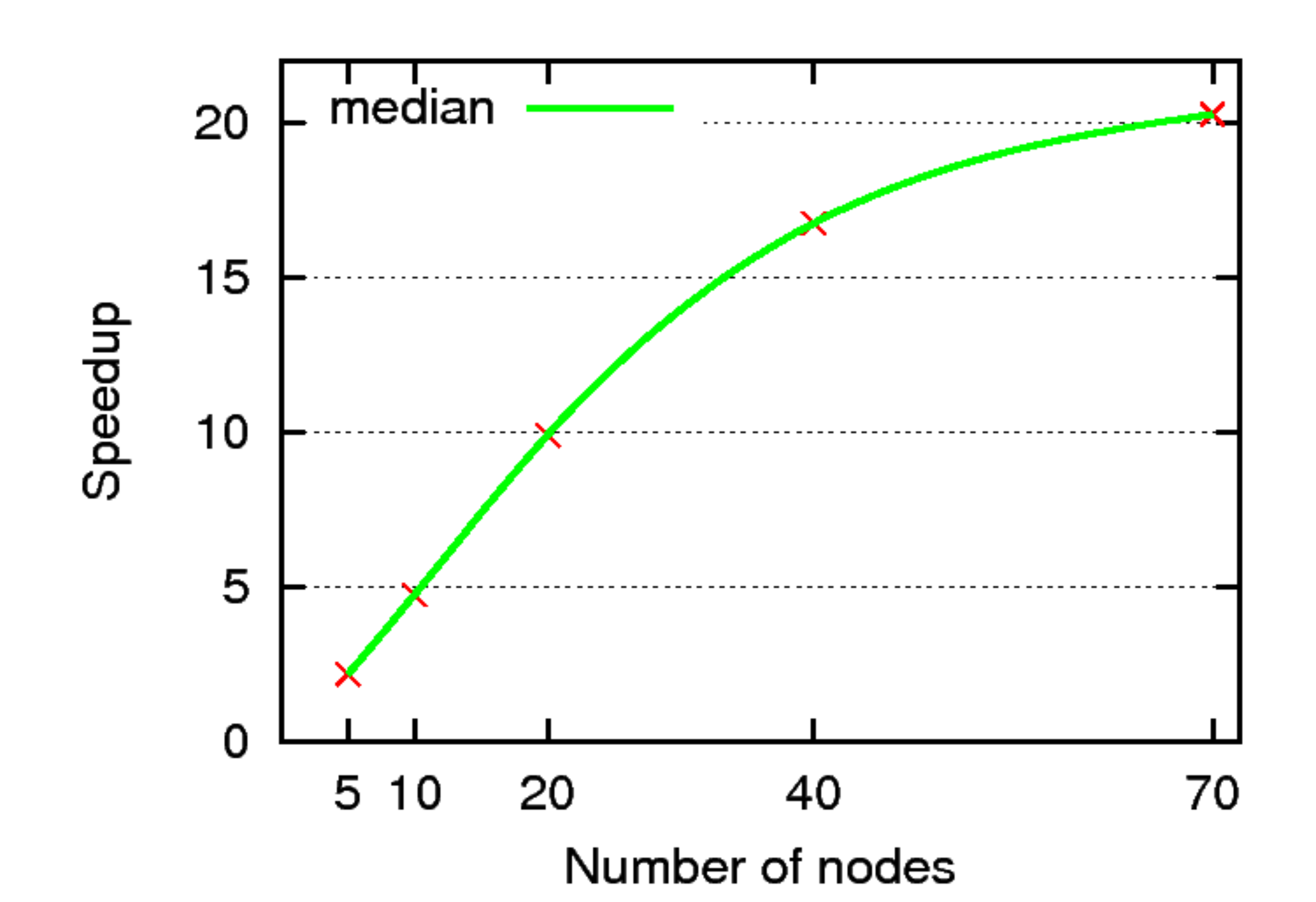
Operating System

Evaluation

D'ARMC

- Distributed software model checker based on ARMC
- Does abstraction refinement through linear interpolation
- Mostly a BFS search

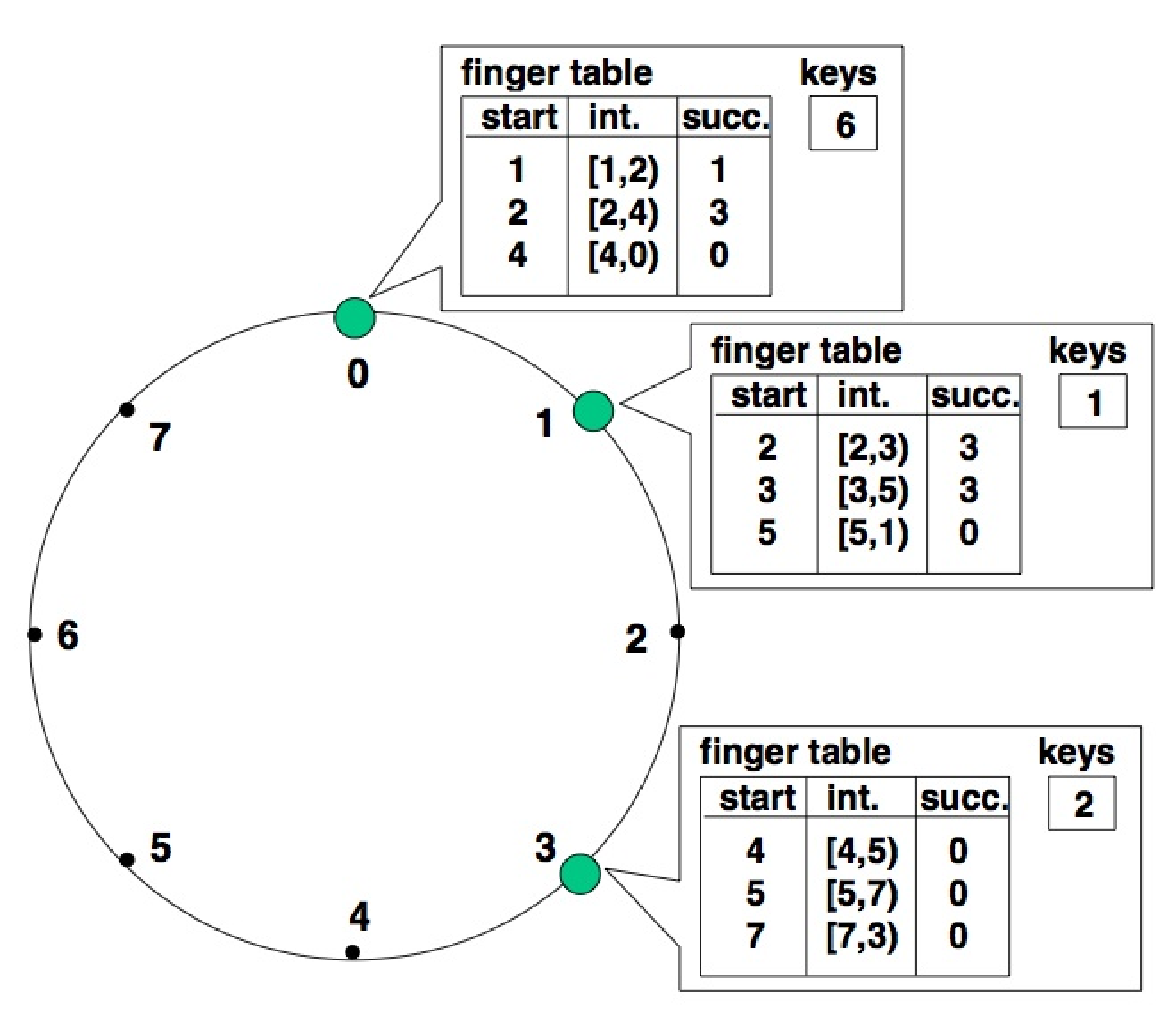
D'ARMC: Speedup



Chord

- A distributed hashtable (aka P2P overlay)
- Nodes organized in a logical ring
- Lookups bounded by the logarithm of the nodes
- ~200 lines of code

Sample Chord Ring



finger table

start	int.	succ.
1	[1,2)	1
2	[2,4)	3
4	[4,0)	0

keys

6

finger table

start	int.	succ.
2	[2,3)	3
3	[3,5)	3
5	[5,1)	0

keys

1

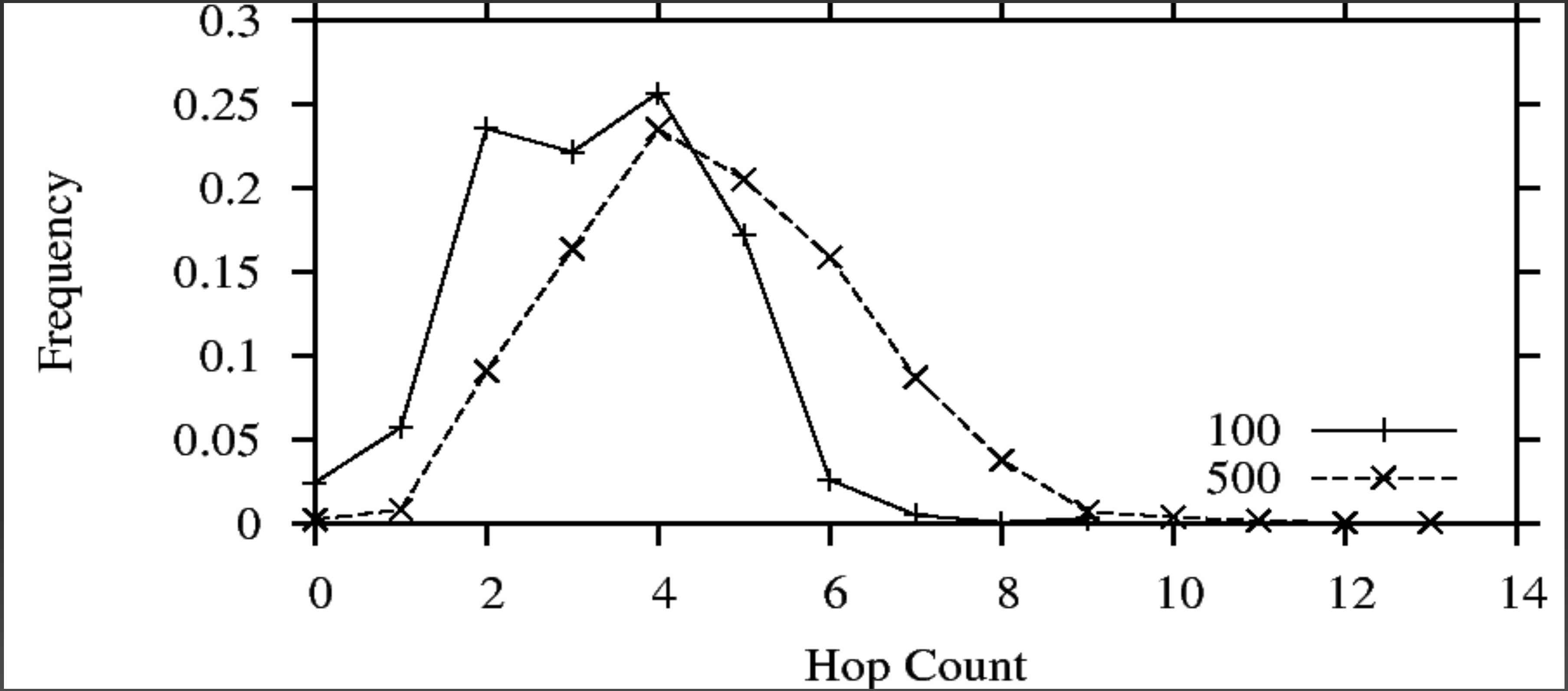
finger table

start	int.	succ.
4	[4,5)	0
5	[5,7)	0
7	[7,3)	0

keys

2

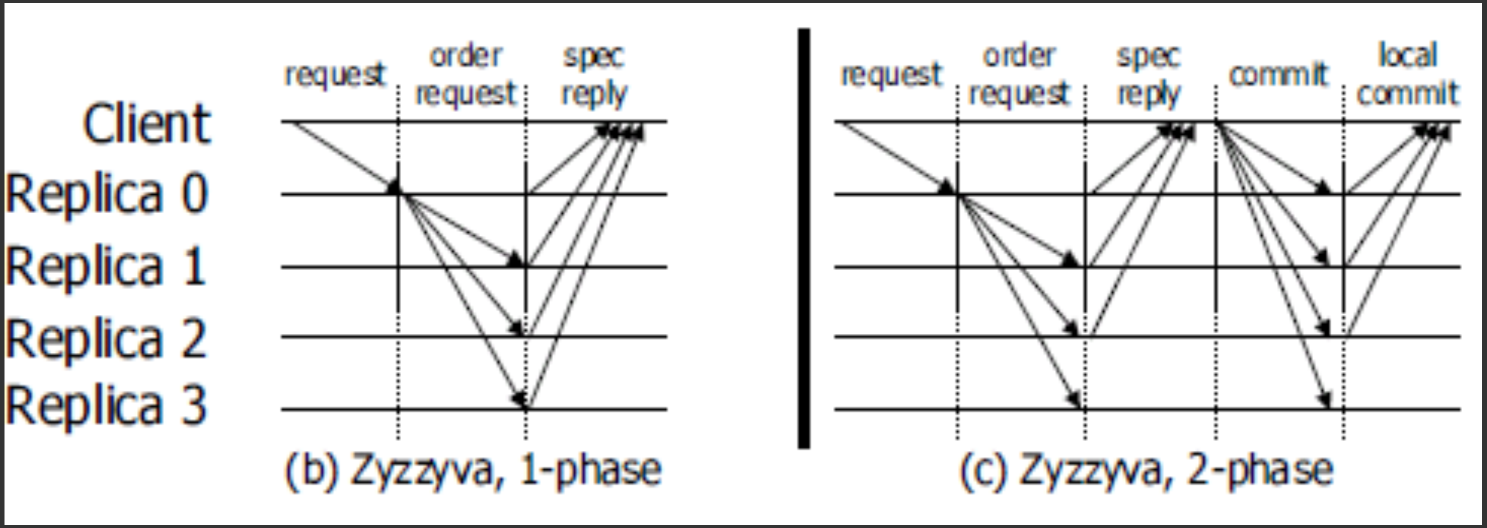
Chord: Hop Count



Zyzzyva

- A complex Byzantine Fault Tolerance (BFT) protocol
- Needs *3f+1* replicas to tolerate *f* faults
- Operates in optimistic way

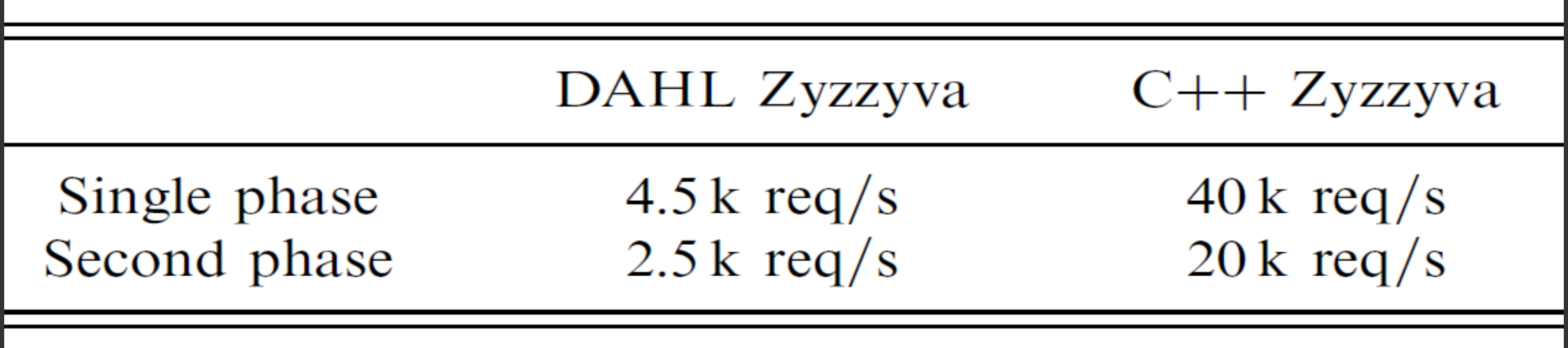
Zyzzyva



(b) Zyzzyva, 1-phase

(c) Zyzzyva, 2-phase

Zyzzyva: Raw Throughput



	DAHL Zyzzyva	C++ Zyzzyva
Single phase	4.5 k req/s	40 k req/s
Second phase	2.5 k req/s	20 k req/s

Conclusions

We presented DAHL:

- An extension to Prolog to implement distributed systems
- An event-driven query executor
- Real applications running today

Grab your copy today:
http://www7.in.tum.de/tools/dahl/