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 pong/0.

pong :-
    print('alive!').

:- event ping/1.

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

- 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

1. Operating System
2. libevent (in C)
3. Network back-end (in C)
4. Runtime dispatcher (in Prolog)
5. De-serialize, dispatch to runtime dispatcher
6. Serialize and send
7. Dispatch event to application handler
Evaluation
D'ARMC

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

Graph showing the speedup of D'ARMC as a function of the number of nodes. The speedup increases with the number of nodes, approaching a median value of around 20 as the number of nodes approaches 70.
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

<table>
<thead>
<tr>
<th>start</th>
<th>int.</th>
<th>succ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[1,2)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>[2,4)</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>[4,0)</td>
<td>0</td>
</tr>
</tbody>
</table>

keys

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>int.</td>
</tr>
<tr>
<td>2</td>
<td>[2,3)</td>
</tr>
<tr>
<td>3</td>
<td>[3,5)</td>
</tr>
<tr>
<td>5</td>
<td>[5,1)</td>
</tr>
</tbody>
</table>

keys

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>int.</td>
</tr>
<tr>
<td>4</td>
<td>[4,5)</td>
</tr>
<tr>
<td>5</td>
<td>[5,7)</td>
</tr>
<tr>
<td>7</td>
<td>[7,3)</td>
</tr>
</tbody>
</table>

keys

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>int.</td>
</tr>
<tr>
<td>4</td>
<td>[4,5)</td>
</tr>
<tr>
<td>5</td>
<td>[5,7)</td>
</tr>
<tr>
<td>7</td>
<td>[7,3)</td>
</tr>
</tbody>
</table>
Chord: Hop Count

![Graph showing hop count distribution with two curves representing different hop counts (100 and 500).]
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

<table>
<thead>
<tr>
<th></th>
<th>DAHL Zyzzyva</th>
<th>C++ Zyzzyva</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single phase</td>
<td>4.5 k req/s</td>
<td>40 k req/s</td>
</tr>
<tr>
<td>Second phase</td>
<td>2.5 k req/s</td>
<td>20 k req/s</td>
</tr>
</tbody>
</table>
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/