Distributed Online Learning of Event Definitions

Nikos Katzouris\textsuperscript{1}, Alexander Artikis\textsuperscript{2,1} and Georios Paliouras\textsuperscript{1}

http://cer.iit.demokritos.gr

\textsuperscript{1}National Center for Scientific Research Demokritos, Athens, Greece
\textsuperscript{2}University of Piraeus, Piraeus, Greece
Complex Event Recognition

Input ►
... □ □ □ □ □ ...
Simple Events
... □ □ □ □ □ ...

Event Recognition System

Recognition ►

Output ■
... ○ ○ ○ ○ ○ ...
Complex Events
... ○ ○ ○ ○ ○ ...

Complex Event Definitions

Simple Events
... □ □ □ □ □ ...

Complex Event Definitions

Complex Events
... ○ ○ ○ ○ ○ ...
Complex Event Recognition

Input ▶

Simple Events

Recognition ▶

Event Recognition System

Complex Event Definitions

initiatedAt(meeting(X, Y), T) ←
happensAt(active(X), T),
happensAt(active(Y), T),
holdsAt(close(X, Y, 25), T).

terminatedAt(meeting(X, Y), T) ←
happensAt(walking(X), T),
not holdsAt(close(X, Y, 25), T).

Output ▶

Complex Events
Complex Event Recognition

Simple Events

- happensAt(active(id₀), 10)
- holdsAt(coord(id₀, 20.88, 11.90), 10)
- happensAt(active(id₁), 10)
- holdsAt(coord(id₁, 22.34, 15.23), 10)

Complex Event Definitions

- initiatedAt(meeting(X, Y), T) ← happensAt(active(X), T), happensAt(active(Y), T), holdsAt(close(X, Y, 25), T).
- terminatedAt(meeting(X, Y), T) ← happensAt(walking(X), T), not holdsAt(close(X, Y, 25), T).
Complex Event Recognition

Simple Events

- happensAt(active(id_0), 10)
- holdsAt(coord(id_0, 20.88, 11.90), 10)
- happensAt(active(id_1), 10)
- holdsAt(coord(id_1, 22.34, 15.23), 10)

Complex Event Definitions

- initiatedAt(meeting(X, Y), T) ←
  - happensAt(active(X), T),
  - happensAt(active(Y), T),
  - holdsAt(close(X, Y, 25), T).
- terminatedAt(meeting(X, Y), T) ←
  - happensAt(walking(X), T),
  - not holdsAt(close(X, Y, 25), T).

Complex Events

- holdsAt(meeting(id_0, id_1), 11)
- holdsAt(meeting(id_0, id_1), 12)
- holdsAt(meeting(id_0, id_1), 13)

Output
Learning for Complex Event Recognition

Input ➤

Simple Events

happensAt(active(id₀), 10)
holdsAt(coord(id₀, 20.88, 11.90), 10)
happensAt(active(id₁), 10)
holdsAt(coord(id₁, 22.34, 15.23), 10)

Recognition ➤

Event Recognition System

Complex Event Definitions

happensAt(meeting(X, Y), T) ←
happensAt(active(X), T),
happensAt(active(Y), T),
holdsAt(close(X, Y, 25), T).

terminatedAt(meeting(X, Y), T) ←
happensAt(walking(X), T),
not holdsAt(close(X, Y, 25), T).

Output ■

Complex Events

holdsAt(meeting(id₀, id₁), 11)
holdsAt(meeting(id₀, id₁), 12)
holdsAt(meeting(id₀, id₁), 13)

Learn this

From These
Complex Event Recognition using the Event Calculus

- Formal, declarative semantics.
- Representation of complex temporal phenomena.
- Representation of complex atemporal phenomena.
- Very efficient reasoning → RTEC.

Direct connections to machine learning → Inductive Logic Programming (ILP).
Complex Event Recognition using the Event Calculus

- Formal, declarative semantics.
- Representation of complex temporal phenomena.
- Representation of complex atemporal phenomena.
- Very efficient reasoning → RTEC.
- Direct connections to machine learning → Inductive Logic Programming (ILP).
## The Event Calculus

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>happensAt($E$, $T$)</td>
<td>Event $E$ occurs at time $T$</td>
</tr>
<tr>
<td>initiatedAt($F$, $T$)</td>
<td>At time $T$ a period of time for which fluent $F$ holds is initiated</td>
</tr>
<tr>
<td>terminatedAt($F$, $T$)</td>
<td>At time $T$ a period of time for which fluent $F$ holds is terminated</td>
</tr>
<tr>
<td>holdsAt($F$, $T$)</td>
<td>Fluent $F$ holds at time $T$</td>
</tr>
</tbody>
</table>
## The Event Calculus

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>happensAt($E$, $T$)</td>
<td>Event $E$ occurs at time $T$</td>
</tr>
<tr>
<td>initiatedAt($F$, $T$)</td>
<td>At time $T$ a period of time for which fluent $F$ holds is initiated</td>
</tr>
<tr>
<td>terminatedAt($F$, $T$)</td>
<td>At time $T$ a period of time for which fluent $F$ holds is terminated</td>
</tr>
<tr>
<td>holdsAt($F$, $T$)</td>
<td>Fluent $F$ holds at time $T$</td>
</tr>
</tbody>
</table>

### Domain-Independent Axioms

\[
\text{holdsAt}(F, T+1) \leftarrow \text{initiatedAt}(F, T).
\]

\[
\text{holdsAt}(F, T+1) \leftarrow \text{holdsAt}(F, T), \not \text{terminatedAt}(F, T).
\]
## The Event Calculus

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>happensAt($E$, $T$)</td>
<td>Event $E$ occurs at time $T$</td>
</tr>
<tr>
<td>initiatedAt($F$, $T$)</td>
<td>At time $T$ a period of time for which fluent $F$ holds is initiated</td>
</tr>
<tr>
<td>terminatedAt($F$, $T$)</td>
<td>At time $T$ a period of time for which fluent $F$ holds is terminated</td>
</tr>
<tr>
<td>holdsAt($F$, $T$)</td>
<td>Fluent $F$ holds at time $T$</td>
</tr>
</tbody>
</table>

### Domain-Independent Axioms

- $\text{holdsAt}(F, T+1) \leftarrow \text{initiatedAt}(F, T)$.  
- $\text{holdsAt}(F, T+1) \leftarrow \text{holdsAt}(F, T)$, not terminatedAt($F$, $T$).

### Domain-Specific Axioms

- $\text{initiatedAt}(\text{meeting}(X, Y), T) \leftarrow \text{happensAt}(\text{active}(X), T)$, 
  $\text{happensAt}(\text{active}(Y), T)$, 
  $\text{holdsAt}(\text{close}(X, Y, 25), T)$.  
- $\text{terminatedAt}(\text{meeting}(X, Y), T) \leftarrow \text{happensAt}(\text{walking}(X), T)$,  
  not holdsAt($\text{close}(X, Y, 25), T$).
Online Inductive Logic Programming

Challenge:
- Inductive Logic Programming algorithms are batch learners.
  - Each candidate in the search space is evaluated on the entire dataset.

Goal:
- Online learning:
  - Examples arrive in a stream.
  - Each example is “seen” once.

Approach:
- Make decisions from subsets of the stream:
  - Decisions are optimal “locally”.
  - Decisions are optimal “globally”...
    - within an error margin $\epsilon$,
    - with probability $1-\delta$. 
The Hoeffding Bound

- $X$ is a random variable.
- $X_1, \ldots, X_N$ are $N$ independent observations of $X$'s values.
- Let $\bar{X}$ be the known, observed mean of $X$.
- Let $\hat{X}$ be the unknown, true mean of $X$. 

Then: $\bar{X} - \epsilon \leq \hat{X} \leq \bar{X} + \epsilon$, with probability $1 - \delta$, where

$$\epsilon = \sqrt{\frac{\ln(1/\delta)}{2N}}$$
The Hoeffding Bound

- $X$ is a random variable.
- $X_1, \ldots, X_N$ are $N$ independent observations of $X$'s values.
- Let $\bar{X}$ be the known, observed mean of $X$.
- Let $\hat{X}$ be the unknown, true mean of $X$.
- Then:
  \[ \bar{X} - \epsilon \leq \hat{X} \leq \bar{X} + \epsilon, \text{ with probability } 1 - \delta, \text{ where } \epsilon = \sqrt{\frac{\ln(1/\delta)}{2N}} \]
Online Rule Learning

Candidate Rules

\[ R_1: 0.345 \]
\[ R_2: 0.232 \]
\[ R_3: 0.145 \]
\[ R_4: 0.612 \]
\[ R_5: 0.325 \]

Find the best candidate across the stream

\[ \bar{X} - \epsilon \leq \hat{X} \leq \bar{X} + \epsilon, \text{ where } \epsilon = \sqrt{\ln \left( \frac{1}{\delta} \right)} \frac{2}{N} \]

Training stream

\[ \text{Find the best candidate across the stream} \]

\[ \bar{X} = \text{score BestRule} - \text{score SecondBestRule} \]

Continue until the number \( N \) of examples makes \( \bar{X} > \epsilon = \sqrt{\ln \left( \frac{1}{\delta} \right)} \frac{2}{N} \)

Then \( \bar{X} - \epsilon > 0 \Rightarrow \hat{X} > 0 \Rightarrow \text{BestRule is indeed the best rule, with probability } 1 - \delta. \]
Online Rule Learning

Candidate Rules

\[ R_1: 0.345 \]
\[ R_2: 0.232 \]
\[ R_3: 0.145 \]
\[ R_4: 0.612 \]
\[ R_5: 0.325 \]

Training stream

Find the best candidate across the stream

As examples stream in...

\[ \bar{X} = \text{score}_{\text{BestRule}} - \text{score}_{\text{SecondBestRule}} \]
Online Rule Learning

Candidate Rules

$R_1: 0.345$

$R_2: 0.232$

$R_3: 0.145$

$R_4: 0.612$

$R_5: 0.325$

Find the best candidate across the stream

Training stream

As examples stream in...

Monitor $\bar{X} = \text{score}_{\text{Best Rule}} - \text{score}_{\text{Second Best Rule}}$

Continue until the number $N$ of examples makes $\bar{X} > \epsilon = \sqrt{\frac{\ln(1/\delta)}{2N}}$
Online Rule Learning

Candidate Rules

\[ R_1: 0.345 \]
\[ R_2: 0.232 \]
\[ R_3: 0.145 \]
\[ R_4: 0.612 \]
\[ R_5: 0.325 \]

Find the best candidate across the stream

Training stream

\[ \tilde{X} - \epsilon \leq \hat{X} \leq \tilde{X} + \epsilon, \text{ where } \epsilon = \sqrt{\frac{\ln(1/\delta)}{2N}} \]

As examples stream in...

Monitor \( \tilde{X} = score_{\text{BestRule}} - score_{\text{SecondBestRule}} \)

Continue until the number \( N \) of examples makes \( \tilde{X} > \epsilon = \sqrt{\frac{\ln(1/\delta)}{2N}} \)

Then

\( \tilde{X} - \epsilon > 0 \implies \)
\( \hat{X} > 0 \implies \)

BestRule is indeed the best rule, with probability \( 1-\delta \).
Online Rule Learning

initiatedAt\( (\text{meet}(X,Y), T) \leftarrow \)

\[ \text{initiatedAt}(\text{meet}(X,Y), T) \leftarrow \text{happensAt}(\text{active}(X), T). \]

\[ \text{initiatedAt}(\text{meet}(X,Y), T) \leftarrow \text{happensAt}(\text{inactive}(Y), T). \]

\[ \text{initiatedAt}(\text{meet}(X,Y), T) \leftarrow \text{happensAt}(\text{inactive}(Y), T). \]

\[ \text{initiatedAt}(\text{meet}(X,Y), T) \leftarrow \text{happensAt}(\text{inactive}(Y), T). \]

\[ \text{initiatedAt}(\text{meet}(X,Y), T) \leftarrow \text{holdsAt}(\text{close}(X,Y,25), T). \]

\[ \text{initiatedAt}(\text{meet}(X,Y), T) \leftarrow \text{holdsAt}(\text{orientation}(X,Y,45), T). \]

\[ \text{initiatedAt}(\text{meet}(X,Y), T) \leftarrow \text{not happensAt}(\text{inactive}(X), T). \]

\[ \text{initiatedAt}(\text{meet}(X,Y), T) \leftarrow \text{not happensAt}(\text{inactive}(Y), T). \]

\[ \text{initiatedAt}(\text{meet}(X,Y), T) \leftarrow \text{not happensAt}(\text{inactive}(Y), T). \]

\[ \text{initiatedAt}(\text{meet}(X,Y), T) \leftarrow \text{not happensAt}(\text{inactive}(Y), T). \]

\[ \text{initiatedAt}(\text{meet}(X,Y), T) \leftarrow \text{not happensAt}(\text{inactive}(Y), T). \]

Bottom Clause \( \perp \):

Training stream

\[ \text{Used } O\left( \frac{1}{\epsilon^2} \ln \frac{1}{\delta} \right) \text{ examples} \]

\[ \text{Used } O\left( \frac{1}{\epsilon^2} \ln \frac{1}{\delta} \right) \text{ examples} \]

\[ \text{Used } O\left( \frac{1}{\epsilon^2} \ln \frac{1}{\delta} \right) \text{ examples} \]
Learning a Theory

- As training examples stream-in...
  - If a positive example is “missed”
    - Add new rule (cover new positives)
  - Gradually expand existing rules. (eliminate negatives)
  - If a rule turns out to be “bad”
    - Remove rule (prune bad rules)
Distributed Learning I: Synchronous Strategy
Distributed Learning I: Synchronous Strategy

Generated rule R

Node 1
Node 2
Node 3
Node 4
Node N
Local Stream
Local Stream
Local Stream
Local Stream
Local Stream...

Local Stream

Node 1
Node 2
Node 3
Node 4
Node N
Local Stream
Local Stream
Local Stream
Local Stream
Local Stream...

Local Stream
Distributed Learning I: Synchronous Strategy

- Generated rule $R$
- Broadcast $R$
- Local Stream

Node 1 → Node 2 → Node 3 → Node 4 → Node N → ... → Local Stream

- Local Stream
Distributed Learning I: Synchronous Strategy

Hoeffding test succeeds for rule R

Node 2

Local Stream

Node 1

Local Stream

Node 3

Local Stream

Node N

Local Stream

Node 4

Local Stream

...
Distributed Learning I: Synchronous Strategy

Node 1

Node 2

Node 3

Node N

Hoeffding test succeeds for rule R

Request evaluation stats for R

Local Stream

Local Stream

Local Stream

Local Stream

Node 1

Node 3

Node N

Node 4
Distributed Learning I: Synchronous Strategy

Node 2

Hoeffding test succeeds for rule R

Send replies

Node 1

Local Stream

Node 3

Local Stream

Node N

Local Stream

Node 4

Local Stream
Distributed Learning I: Synchronous Strategy

Add received stats to local ones and re-assess specialization

Node 2

Local Stream

Add received stats to local ones and re-assess specialization

Node 1

Local Stream

Add received stats to local ones and re-assess specialization

Node 3

Local Stream

Add received stats to local ones and re-assess specialization

Node N

Local Stream
Distributed Learning I: Synchronous Strategy

If R is specialized to R'

Broadcast R'

Node 2

Local Stream

Node 1

Local Stream

Node 3

Local Stream

Node N

Local Stream

Node 4

Local Stream
Each node learns independently from its own training stream.
Distributed Learning II: Asynchronous Strategy

Each node learns independently from its own training stream.

Each node runs the monolithic OLED algorithm.

- Generate new rules.
- Specialize rules.
- prune rules

Local Stream

Node 1

- Generate new rules.
- Specialize rules.
- prune rules

Node 2

Local Stream

Node 3

- Generate new rules.
- Specialize rules.
- prune rules

Node N

Local Stream

Node 4

- Generate new rules.
- Specialize rules.
- prune rules

...
Distributed Learning II: Asynchronous Strategy

Each node learns independently from its own training stream.

If some rule R is good-enough locally...

Broadcast R
Distributed Learning II: Asynchronous Strategy

Return the rules common to the majority of the nodes.
Empirical Evaluation I: Activity Recognition

- Activity recognition using a benchmark dataset (CAVIAR).
  - 28 surveillance videos.
- Input: short-term activities per video frame + contextual information:
  - walking, active, inactive, running.
  - coordinates, orientation, occlusion.
- Learn concepts for *Move* and *Meet*.
- 10-fold cross-validation.
## Empirical Evaluation: Activity Recognition

<table>
<thead>
<tr>
<th>(A)</th>
<th>#Cores</th>
<th>Time (sec)</th>
<th>Speed-up</th>
<th>$F_1$-score</th>
<th>Theory size</th>
<th># Mgs</th>
</tr>
</thead>
<tbody>
<tr>
<td>meeting</td>
<td>1</td>
<td>46</td>
<td>–</td>
<td>0.798</td>
<td>28</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>moving</td>
<td>1</td>
<td>68</td>
<td>–</td>
<td>0.744</td>
<td>21</td>
<td>–</td>
</tr>
<tr>
<td>(B)</td>
<td>meeting</td>
<td>1</td>
<td>7588</td>
<td>–</td>
<td>0.834</td>
<td>36</td>
</tr>
<tr>
<td>moving</td>
<td>1</td>
<td>7898</td>
<td>–</td>
<td>0.758</td>
<td>34</td>
<td>–</td>
</tr>
</tbody>
</table>
Implementation & Future Work

▶ Future Work
  ▶ Evaluate on larger and more demanding datasets.
  ▶ More robust distribution strategies.

▶ Code
  ▶ Scala + akka Actors library.
  ▶ Clingo answer set solver for reasoning.
  ▶ GitHub: http://github.com/nkatzz/OLED

▶ *Part of the presented work was developed after the submission to ILP 2017.*
Appendix I: Learning Sets of Clauses

<table>
<thead>
<tr>
<th>TP</th>
<th>Annotation: holds</th>
<th>Inferred: holds</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP</td>
<td>Annotation: not holds</td>
<td>Inferred: holds</td>
</tr>
<tr>
<td>FN</td>
<td>Annotation: holds</td>
<td>Inferred: not holds</td>
</tr>
</tbody>
</table>

- **TP (True Positive)**: Annotation holds, Inferred holds → All ok!
- **FP (False Positive)**: Annotation not holds, Inferred holds → Incorrectly initiated by clause $R_{init}$ OR No termination clause "fires" OR Specialize $R_{init}$ OR Generate new termination clause
- **FN (False Negative)**: Annotation holds, Inferred not holds → Incorrectly terminated by clause $R_{term}$ OR No initiation clause "fires" OR Specialize $R_{term}$ OR Generate new initiation clause

**Initiation Learner**
- Reward all clauses that correctly initiate the TP

**Termination Learner**
- Reward all clauses that correctly allow the TP to persist
Appendix I: Learning Sets of Clauses

<table>
<thead>
<tr>
<th>TP</th>
<th>Annotation</th>
<th>Inferred</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>holds</td>
<td>holds</td>
<td>All ok!</td>
</tr>
<tr>
<td>FP</td>
<td>not holds</td>
<td>holds</td>
<td>Incorrectly initiated by clause $R_{init}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Specialize $R_{init}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>No termination clause “fires”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Generate new termination clause</td>
</tr>
<tr>
<td>FN</td>
<td>holds</td>
<td>not holds</td>
<td>Incorrectly terminated by clause $R_{term}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Specialize $R_{term}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>No initiation clause “fires”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Generate new initiation clause</td>
</tr>
</tbody>
</table>

Initiation Learner

Reward all clauses that correctly initiate the TP

Termination Learner

Reward all clauses that correctly allow the TP to persist

Input stream
### Appendix I: Learning Sets of Clauses

<table>
<thead>
<tr>
<th>TP</th>
<th>Annotation</th>
<th>Inferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>holds</td>
<td>holds</td>
</tr>
</tbody>
</table>

**All ok!**

<table>
<thead>
<tr>
<th>FP</th>
<th>Annotation</th>
<th>Inferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP</td>
<td>not holds</td>
<td>holds</td>
</tr>
</tbody>
</table>

Incorrectly initiated by clause $R_{init}$

Specialize $R_{init}$

Generate new termination clause

OR

<table>
<thead>
<tr>
<th>FN</th>
<th>Annotation</th>
<th>Inferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>FN</td>
<td>holds</td>
<td>not holds</td>
</tr>
</tbody>
</table>

Incorrectly terminated by clause $R_{term}$

Specialize $R_{term}$

Generate new initiation clause

**OR**

**Initiation Learner**

Reward all clauses that correctly initiate the TP

**Termination Learner**

Reward all clauses that correctly allow the TP to persist

---

Input stream
Appendix I: Learning Sets of Clauses

TP: Annotation holds, Inferred holds → All ok!

FP: Annotation not holds, Inferred holds → Incorrectly initiated by clause $R_{\text{init}}$

FP: Annotation not holds, Inferred holds → No termination clause “fires”

FN: Annotation holds, Inferred not holds → Incorrectly terminated by clause $R_{\text{term}}$

FN: Annotation holds, Inferred not holds → No initiation clause “fires”

Initiation Learner
Penalize all clauses that incorrectly initiate the FP

Termination Learner
Generate new termination clause

Input stream

FP

FP
Appendix I: Learning Sets of Clauses

TP  | Annotation: holds | Inferred: holds |
---  |------------------|----------------|
     | All ok!          |                |

FP  | Annotation: not holds | Inferred: holds |
---  |-----------------------|----------------|
     | Incorrectly initiated by clause $R_{\text{init}}$ |                |
     | Specialize $R_{\text{init}}$ | Generate new termination clause |
     | OR                      |                |

FN  | Annotation: holds | Inferred: not holds |
---  |------------------|---------------------|
     | Incorrectly terminated by clause $R_{\text{term}}$ |                |
     | Specialize $R_{\text{term}}$ | Generate new initiation clause |
     | OR                      |                |

Initiation Learner
Generate new initiation clause

Termination Learner
Penalize all clauses that generate the FN

Input stream
## Appendix II: Maritime Surveillance Experiments

<table>
<thead>
<tr>
<th>#Cores</th>
<th>Time (sec)</th>
<th>Speed-up</th>
<th>F$_1$-score</th>
<th>Theory size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>sync</td>
<td>async</td>
<td>sync</td>
</tr>
<tr>
<td>highSpeedIn</td>
<td>1</td>
<td>532</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sync</td>
<td>async</td>
<td>sync</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>116</td>
<td>105</td>
<td>4.58</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>77</td>
<td>70</td>
<td>6.9</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>65</td>
<td>54</td>
<td>8.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sync</td>
<td>async</td>
<td>sync</td>
</tr>
<tr>
<td></td>
<td>lowSpeed</td>
<td>1</td>
<td>164</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sync</td>
<td>async</td>
<td>sync</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>52</td>
<td>42</td>
<td>3.15</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>48</td>
<td>43</td>
<td>3.41</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>51</td>
<td>54</td>
<td>3.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sync</td>
<td>async</td>
<td>sync</td>
</tr>
<tr>
<td></td>
<td>stopped</td>
<td>1</td>
<td>98 test</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sync</td>
<td>async</td>
<td>sync</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>56</td>
<td>51</td>
<td>1.75</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>49</td>
<td>47</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>51</td>
<td>52</td>
<td>1.92</td>
</tr>
</tbody>
</table>