Perfume: resource-aware model inference

Tony Ohmann, Michael Herzberg, Sebastian Fiss, Armand Halbert, Marc Palyart, Ivan Beschastnikh, Yuriy Brun

University of Massachusetts, Amherst
University of British Columbia
Motivation: system understanding

- Imagine a poorly documented system that tests network connections

- A developer wants to understand: what causes the system to classify connections as problematic?

- The developer can:
  - examine source code,
  - step through executions, or
  - instrument the system and inspect the console log
Motivation: system understanding

- Imagine a poorly documented system that tests network connections

- A developer wants to understand: what causes the system to classify connections as problematic?

- The developer can:
  - examine source code,
  - step through executions, or
  - instrument the system and inspect the console log
Motivation: system understanding

- Imagine a poorly documented system that tests network connections

- A developer wants to understand: what causes the system to classify connections as problematic?

- The developer can:
  - examine source code,
  - step through executions, or
  - instrument the system and inspect the console log

Perfume: resource-aware model inference
• Complex

• Relevant information spread out

• Describes single executions, not aggregate system behavior
Perfume: inferring a log model

- Perfume model
Perfume: inferring a log model

- **Perfume model**

- Resource in this log: time

- Other example resources: memory, bytes transferred, CPU usage

---

**Console log**

- 19.38.218.11 [31/May/2014:31200.0] "GET HTTP/1.1 /test bandwidth"
- 210.82.199.247 [31/May/2014:31200.1] "GET HTTP/1.1 /test bandwidth"
- 38.151.1.182 [31/May/2014:31200.2] "GET HTTP/1.1 /test bandwidth"
- 95.39.21.28 [31/May/2014:31200.3] "GET HTTP/1.1 /test bandwidth"
- 210.82.199.247 [31/May/2014:31200.8] "GET HTTP/1.1 /broadband"
- 38.151.1.182 [31/May/2014:31200.9] "GET HTTP/1.1 /broadband"
- 19.38.218.11 [31/May/2014:31202.0] "GET HTTP/1.1 /narrowband"
- 210.82.199.247 [31/May/2014:31202.1] "GET HTTP/1.1 /query"
- 38.151.1.182 [31/May/2014:31202.2] "GET HTTP/1.1 /query"
- 95.39.21.28 [31/May/2014:31202.3] "GET HTTP/1.1 /narrowband"
- 38.151.1.182 [31/May/2014:31203.6] "GET HTTP/1.1 /query"
- 38.151.1.182 [31/May/2014:31204.1] "GET HTTP/1.1 /OK"
- 19.38.218.11 [31/May/2014:31205.7] "GET HTTP/1.1 /query"
- 95.39.21.28 [31/May/2014:31206.0] "GET HTTP/1.1 /query"
- 95.39.21.28 [31/May/2014:31206.8] "GET HTTP/1.1 /OK"
- 210.82.199.247 [31/May/2014:31208.3] "GET HTTP/1.1 /query"
- 210.82.199.247 [31/May/2014:31208.8] "GET HTTP/1.1 /problem"
Perfume: inferring a log model

- Perfume model

**Console log**

19.38.218.11 [31/May/2014:31200.0] "GET HTTP/1.1 /test bandwidth"
210.82.199.247 [31/May/2014:31200.1] "GET HTTP/1.1 /test bandwidth"
38.151.1.182 [31/May/2014:31200.2] "GET HTTP/1.1 /test bandwidth"
95.39.21.28 [31/May/2014:31200.3] "GET HTTP/1.1 /test bandwidth"

210.82.199.247 [31/May/2014:31200.8] "GET HTTP/1.1 /broadband"
38.151.1.182 [31/May/2014:31200.9] "GET HTTP/1.1 /broadband"
19.38.218.11 [31/May/2014:31202.0] "GET HTTP/1.1 /narrowband"

210.82.199.247 [31/May/2014:31202.1] "GET HTTP/1.1 /query"
38.151.1.182 [31/May/2014:31202.2] "GET HTTP/1.1 /query"
95.39.21.28 [31/May/2014:31202.3] "GET HTTP/1.1 /narrowband"
38.151.1.182 [31/May/2014:31203.6] "GET HTTP/1.1 /query"
38.151.1.182 [31/May/2014:31204.1] "GET HTTP/1.1 /OK"
19.38.218.11 [31/May/2014:31205.7] "GET HTTP/1.1 /query"
95.39.21.28 [31/May/2014:31206.0] "GET HTTP/1.1 /query"
95.39.21.28 [31/May/2014:31206.8] "GET HTTP/1.1 /OK"

210.82.199.247 [31/May/2014:31208.3] "GET HTTP/1.1 /query"
210.82.199.247 [31/May/2014:31208.8] "GET HTTP/1.1 /problem"
19.38.218.11 [31/May/2014:31208.9] "GET HTTP/1.1 /query"
19.38.218.11 [31/May/2014:31209.7] "GET HTTP/1.1 /OK"

**Resource-aware model**

Perfume: inferring a log model

- Perfume model

**Console log**

19.38.218.11 [31/May/2014:31200.0] "GET HTTP/1.1 /test bandwidth"
210.82.199.247 [31/May/2014:31200.1] "GET HTTP/1.1 /test bandwidth"
38.151.1.182 [31/May/2014:31200.2] "GET HTTP/1.1 /test bandwidth"
95.39.21.28 [31/May/2014:31200.3] "GET HTTP/1.1 /test bandwidth"

210.82.199.247 [31/May/2014:31200.8] "GET HTTP/1.1 /broadband"
38.151.1.182 [31/May/2014:31200.9] "GET HTTP/1.1 /broadband"
19.38.218.11 [31/May/2014:31202.0] "GET HTTP/1.1 /narrowband"

210.82.199.247 [31/May/2014:31202.1] "GET HTTP/1.1 /query"
38.151.1.182 [31/May/2014:31202.2] "GET HTTP/1.1 /query"
95.39.21.28 [31/May/2014:31202.3] "GET HTTP/1.1 /narrowband"
38.151.1.182 [31/May/2014:31203.6] "GET HTTP/1.1 /query"
38.151.1.182 [31/May/2014:31204.1] "GET HTTP/1.1 /OK"
19.38.218.11 [31/May/2014:31205.7] "GET HTTP/1.1 /query"
95.39.21.28 [31/May/2014:31206.0] "GET HTTP/1.1 /query"
95.39.21.28 [31/May/2014:31206.8] "GET HTTP/1.1 /OK"

210.82.199.247 [31/May/2014:31208.3] "GET HTTP/1.1 /query"
210.82.199.247 [31/May/2014:31208.8] "GET HTTP/1.1 /problem"
19.38.218.11 [31/May/2014:31208.9] "GET HTTP/1.1 /query"
19.38.218.11 [31/May/2014:31209.7] "GET HTTP/1.1 /OK"
Perfume: inferring a log model

- **Perfume model**
  - differentiates behavior
    - **broadband** (left) and **narrowband** (right) separated

---

**Console log**

- 19.38.218.11 [31/May/2014:31200.0] "GET HTTP/1.1 /test bandwidth"
- 210.82.199.247 [31/May/2014:31200.1] "GET HTTP/1.1 /test bandwidth"
- 38.151.1.182 [31/May/2014:31200.2] "GET HTTP/1.1 /test bandwidth"
- 95.39.21.28 [31/May/2014:31200.3] "GET HTTP/1.1 /test bandwidth"
- 210.82.199.247 [31/May/2014:31200.8] "GET HTTP/1.1 /broadband"
- 38.151.1.182 [31/May/2014:31200.9] "GET HTTP/1.1 /broadband"
- 19.38.218.11 [31/May/2014:31202.0] "GET HTTP/1.1 /narrowband"
- 210.82.199.247 [31/May/2014:31202.1] "GET HTTP/1.1 /query"
- 38.151.1.182 [31/May/2014:31202.2] "GET HTTP/1.1 /query"
- 95.39.21.28 [31/May/2014:31202.3] "GET HTTP/1.1 /narrowband"
- 38.151.1.182 [31/May/2014:31203.6] "GET HTTP/1.1 /query"
- 38.151.1.182 [31/May/2014:31204.1] "GET HTTP/1.1 /OK"
- 19.38.218.11 [31/May/2014:31205.7] "GET HTTP/1.1 /query"
- 95.39.21.28 [31/May/2014:31206.0] "GET HTTP/1.1 /query"
- 95.39.21.28 [31/May/2014:31206.8] "GET HTTP/1.1 /OK"
- 210.82.199.247 [31/May/2014:31208.3] "GET HTTP/1.1 /query"
- 210.82.199.247 [31/May/2014:31208.8] "GET HTTP/1.1 /problem"
- 19.38.218.11 [31/May/2014:31208.9] "GET HTTP/1.1 /query"
- 19.38.218.11 [31/May/2014:31209.7] "GET HTTP/1.1 /OK"

---

**Resource-aware model**

- Perfume: resource-aware model inference
Perfume: inferring a log model

- Perfume model
  - differentiates behavior
    - broadband (left) and narrowband (right) separated
  - differentiates resource consumption
    - second query: slow (left) and fast (right) separated

Perfume: resource-aware model inference

Console log

Resource-aware model
Perfume: inferring a log model

- **Perfume model**
  - visualizes problematic executions
  - helps answer questions
  - “what causes the system to classify connections as problematic?”

---

**Console log**

- 19.38.218.11 [31/May/2014:31200.0] "GET HTTP/1.1 /test bandwidth"
- 210.82.199.247 [31/May/2014:31200.1] "GET HTTP/1.1 /test bandwidth"
- 38.151.1.182 [31/May/2014:31200.2] "GET HTTP/1.1 /test bandwidth"
- 95.39.21.28 [31/May/2014:31200.3] "GET HTTP/1.1 /test bandwidth"
- 210.82.199.247 [31/May/2014:31200.8] "GET HTTP/1.1 /broadband"
- 38.151.1.182 [31/May/2014:31200.9] "GET HTTP/1.1 /broadband"
- 19.38.218.11 [31/May/2014:31202.0] "GET HTTP/1.1 /narrowband"
- 210.82.199.247 [31/May/2014:31202.1] "GET HTTP/1.1 /query"
- 38.151.1.182 [31/May/2014:31202.2] "GET HTTP/1.1 /query"
- 95.39.21.28 [31/May/2014:31202.3] "GET HTTP/1.1 /narrowband"
- 38.151.1.182 [31/May/2014:31203.6] "GET HTTP/1.1 /query"
- 38.151.1.182 [31/May/2014:31204.1] "GET HTTP/1.1 /OK"
- 19.38.218.11 [31/May/2014:31205.7] "GET HTTP/1.1 /query"
- 95.39.21.28 [31/May/2014:31206.0] "GET HTTP/1.1 /query"
- 95.39.21.28 [31/May/2014:31206.8] "GET HTTP/1.1 /OK"
- 210.82.199.247 [31/May/2014:31208.3] "GET HTTP/1.1 /query"
- 210.82.199.247 [31/May/2014:31208.8] "GET HTTP/1.1 /problem"
- 19.38.218.11 [31/May/2014:31208.9] "GET HTTP/1.1 /query"
- 19.38.218.11 [31/May/2014:31209.7] "GET HTTP/1.1 /OK"

---

**Resource-aware model**

- test bandwidth 0.0
- broadband 0.7
- narrowband 2.0
- query 1.3
- query 3.7
- problem 0.5
- OK 0.8
- query 6.2
- query 1.4
- query 3.2
- OK 0.5
Perfume: inferring a log model

- **Perfume model**
  - visualizes problematic executions
  - helps answer questions
  - “what causes the system to classify connections as problematic?”
Perfume: inferring a log model

- Perfume model
  - visualizes problematic executions
  - helps answer questions
    - “what causes the system to classify connections as problematic?”
  - answer: broadband clients with slow second query

```
19.38.218.11 [31/May/2014:31200.0] "GET HTTP/1.1 /test bandwidth"
210.82.199.247 [31/May/2014:31200.1] "GET HTTP/1.1 /test bandwidth"
38.151.1.182 [31/May/2014:31200.2] "GET HTTP/1.1 /test bandwidth"
95.39.21.28 [31/May/2014:31200.3] "GET HTTP/1.1 /test bandwidth"
210.82.199.247 [31/May/2014:31200.8] "GET HTTP/1.1 /broadband"
38.151.1.182 [31/May/2014:31200.9] "GET HTTP/1.1 /broadband"
19.38.218.11 [31/May/2014:31202.0] "GET HTTP/1.1 /narrowband"
210.82.199.247 [31/May/2014:31202.1] "GET HTTP/1.1 /query"
38.151.1.182 [31/May/2014:31202.2] "GET HTTP/1.1 /query"
95.39.21.28 [31/May/2014:31202.3] "GET HTTP/1.1 /narrowband"
38.151.1.182 [31/May/2014:31203.6] "GET HTTP/1.1 /query"
38.151.1.182 [31/May/2014:31204.1] "GET HTTP/1.1 /OK"
19.38.218.11 [31/May/2014:31205.7] "GET HTTP/1.1 /query"
95.39.21.28 [31/May/2014:31206.0] "GET HTTP/1.1 /query"
95.39.21.28 [31/May/2014:31206.8] "GET HTTP/1.1 /OK"
210.82.199.247 [31/May/2014:31208.3] "GET HTTP/1.1 /query"
210.82.199.247 [31/May/2014:31208.8] "GET HTTP/1.1 /problem"
19.38.218.11 [31/May/2014:31208.9] "GET HTTP/1.1 /query"
19.38.218.11 [31/May/2014:31209.7] "GET HTTP/1.1 /OK"
```
Perfume motivation

Console logs

- rich, low-level descriptions of system behavior
- massive, difficult to interpret

Our solution: summarize log with a resource-aware model

Related work

- Property inference: [Ernst et al. 2001; Yang et al. 2006; Gabel and Su 2008]
- System understanding: [Kunz et al. 1997; Cornelissen et al. 2011]
**Perfume motivation**

**Console logs**
- rich, low-level descriptions of system behavior
- massive, difficult to interpret

**Our solution:** summarize log with a resource-aware model

**Related work**
- **Model inference:** [Biermann and Feldman 1972; Walkinshaw and Bogdanov 2008; Lorenzoli et al. 2008; Beschastnikh et al. 2011; Schur et al. 2013; Ghezzi et al. 2014]
- **Property inference:** [Ernst et al. 2001; Yang et al. 2006; Gabel and Su 2008]
- **Performance:** [Aguilera et al. 2003; Barham et al. 2004; Fonseca et al. 2007; Jiang et al. 2007; Jovic et al. 2011; Sambasivan et al. 2011; Grechanik et al. 2012; Fahland et al. 2013]
- **System understanding:** [Kunz et al. 1997; Cornelissen et al. 2011]

**Our contribution:** Improve model precision and quality with resource usage data
Key insight

Observations

1) behavior depends on resource usage
   - caching
   - timeouts
   - network protocols

2) most runtime logs already contain resource usage data
   - time
   - bytes transferred
   - power/memory/CPU usage

Key insight: **Resource-aware inference ⇒ better models**

Key challenges: 1. precise models
                2. concise models
                3. usable models
**Key insight**

**Observations**

1) behavior depends on **resource** usage
   - caching
   - timeouts
   - network protocols

2) most runtime logs already contain **resource** usage data
   - time
   - bytes transferred
   - power/memory/CPU usage

**Key insight:** **Resource-aware inference** ⇒ **better models**

**Key challenges:** 1. **precise** models
   2. **concise** models
   3. **usable** models
**Perfume approach**

**Goal:** infer a **precise, concise, usable** FSM with resource-use transitions

1. **Parse log**
2. **Mine resource-constrained properties**
3. **Create concise model**
4. **Refine model to remove property violations**
1) Parse log

- User specifies regular expressions to parse the log
- Extract each execution
- Identify events within each execution
- Note resource usage

Parse log

mine resource-constrained properties

create concise model

refine model to remove property violations

Console log

one execution

19.38.218.11 [31/May/2014:31200.0] "GET HTTP/1.1 /test bandwidth"
19.38.218.11 [31/May/2014:31202.0] "GET HTTP/1.1 /narrowband"
19.38.218.11 [31/May/2014:31205.7] "GET HTTP/1.1 /query"
19.38.218.11 [31/May/2014:31208.9] "GET HTTP/1.1 /query"
19.38.218.11 [31/May/2014:31209.7] "GET HTTP/1.1 /OK"

210.82.199.247 [31/May/2014:31200.1] "GET HTTP/1.1 /test bandwidth"
210.82.199.247 [31/May/2014:31200.8] "GET HTTP/1.1 /broadband"
210.82.199.247 [31/May/2014:31202.1] "GET HTTP/1.1 /query"
210.82.199.247 [31/May/2014:31208.3] "GET HTTP/1.1 /query"
210.82.199.247 [31/May/2014:31208.8] "GET HTTP/1.1 /problem"

38.151.1.182 [31/May/2014:31200.2] "GET HTTP/1.1 /test bandwidth"
38.151.1.182 [31/May/2014:31200.9] "GET HTTP/1.1 /broadband"
38.151.1.182 [31/May/2014:31202.2] "GET HTTP/1.1 /query"
38.151.1.182 [31/May/2014:31203.6] "GET HTTP/1.1 /query"
38.151.1.182 [31/May/2014:31204.1] "GET HTTP/1.1 /OK"

95.39.21.28 [31/May/2014:31200.3] "GET HTTP/1.1 /test bandwidth"
95.39.21.28 [31/May/2014:31202.3] "GET HTTP/1.1 /narrowband"
95.39.21.28 [31/May/2014:31206.0] "GET HTTP/1.1 /query"
95.39.21.28 [31/May/2014:31206.8] "GET HTTP/1.1 /OK"
1) Parse log

- User specifies regular expressions to parse the log
- Extract each **execution**
- Identify **events** within each execution
- Note **resource usage**
1) Parse log

- User specifies regular expressions to parse the log
- Extract each **execution**
- Identify **events** within each execution
- Note **resource usage**

```
19.38.218.11 [31/May/2014:31200.0] "GET HTTP/1.1 /test bandwidth"
19.38.218.11 [31/May/2014:31202.0] "GET HTTP/1.1 /narrowband"
19.38.218.11 [31/May/2014:31205.7] "GET HTTP/1.1 /query"
19.38.218.11 [31/May/2014:31208.9] "GET HTTP/1.1 /query"
19.38.218.11 [31/May/2014:31209.7] "GET HTTP/1.1 /OK"
210.82.199.247 [31/May/2014:31200.1] "GET HTTP/1.1 /test bandwidth"
210.82.199.247 [31/May/2014:31200.8] "GET HTTP/1.1 /broadband"
210.82.199.247 [31/May/2014:31202.1] "GET HTTP/1.1 /query"
210.82.199.247 [31/May/2014:31208.3] "GET HTTP/1.1 /query"
210.82.199.247 [31/May/2014:31208.8] "GET HTTP/1.1 /OK"
38.151.1.182 [31/May/2014:31200.2] "GET HTTP/1.1 /test bandwidth"
38.151.1.182 [31/May/2014:31200.9] "GET HTTP/1.1 /broadband"
38.151.1.182 [31/May/2014:31202.2] "GET HTTP/1.1 /query"
38.151.1.182 [31/May/2014:31203.6] "GET HTTP/1.1 /query"
38.151.1.182 [31/May/2014:31204.1] "GET HTTP/1.1 /OK"
95.39.21.28 [31/May/2014:31200.3] "GET HTTP/1.1 /test bandwidth"
95.39.21.28 [31/May/2014:31202.3] "GET HTTP/1.1 /narrowband"
95.39.21.28 [31/May/2014:31206.0] "GET HTTP/1.1 /query"
95.39.21.28 [31/May/2014:31206.8] "GET HTTP/1.1 /OK"
```
2) Mine properties

- Mine observed, resource-constrained properties
  - approximate system’s true properties

- e.g., *narrowband* always followed by *OK* in ≤ 7.7s
- e.g., *broadband* always precedes *problem* in ≥ 8.0s

1 Dwyer et al. ICSE1999
3) Create concise model

- Start with a very **concise**, small model
  - not **precise**
- Transitions with identical event types lead to one state
  - e.g., **query**
4) Refine model

- Identify violations of observed properties
- **Violation:**
  - `narrowband` always followed by `OK` in \( \leq 7.7 \)s

Initial model: **violates** property
4) Refine model

- Identify violations of observed properties
- **Violation:**
  - `narrowband` always followed by `OK` in $\leq 7.7s$

```
Initial model: violates property
```

```
(query) (query) (OK)
3.7 + 6.2 + 0.8 = 10.7
10.7 > 7.7s
```
4) Refine model

- Make model **precise** by removing violations
- **Violation resolved:**
  - *narrowband* always followed by *OK* in \( \leq 7.7 \text{s} \)

```
Initial model: violates property

Refined model: satisfies property
```
4) Refine model

- Make model **precise** by removing violations

- **Violation resolved:**
  - *narrowband* always followed by *OK* in $\leq 7.7s$

![Diagram showing initial and refined models](image-url)
1) Perfume web interface: http://bestchai.bitbucket.org/perfume
Evaluation summary

1) RQ1: Do resource-aware models increase system understanding?
   - user study: Perfume users 12.4% more correct, 12.3% faster

2) RQ2: Can Perfume model real network protocol behavior?
   - TCP case study: model revealed real TCP behavior

3) RQ3: Can Perfume model large-scale website behavior?
   - real estate website case study
   - replicated BEAR results

- Perfume model usability: http://bestchai.bitbucket.org/perfume

---

1 Ghezzi et al. ICSE2014
RQ1: Do resource-aware models increase system understanding?

**Small-scale user study**

- Asked 13 users questions
  - e.g., “Are there traces that result in *Access-Reject* that contain no *Access-Challenge* messages?”
  - e.g., “Does *cache-page* ever occur after *cache-image*?”

- Compared **Perfume** to other system understanding techniques
  - 12.4% more correct, 12.3% faster comprehension than using console logs alone

---

TCP case study

- **Perfume**-inferred model of web browser TCP traffic illustrated:
  1) Timeouts
  2) Buffer pushes (\textit{psh} packets)
  3) Connection resets (\textit{rst} packets)

- Actual TCP properties visible without prior TCP knowledge

---

\(^1\) [http://people.cs.umass.edu/ohmann/perfume/ase2014](http://people.cs.umass.edu/ohmann/perfume/ase2014)
RQ3: Can Perfume model large-scale website behavior?

Website case study¹

- compared Perfume to the BEAR² analysis tool
- Perfume verifies behavior and bug discovered by BEAR

¹ http://people.cs.umass.edu/ohmann/perfume/ase2014

² Ghezzi et al. ICSE2014
Technical contributions

- Temporal, resource-based system properties
- Resource-aware property mining
- Supporting non-monotonic resources
- CEGAR refinement extended for resource use
- Model visualization with resources
Perfume contributions

- Defined resource-based properties
- Developed resource-aware model inference
- Deployed cloud-based UI

- Evaluated Perfume in a pilot user study
  - improved user’s program comprehension
- Evaluated Perfume in two case studies
  - revealed real TCP behavior
  - replicated BEAR results

http://people.cs.umass.edu/ohmann/perfume
Backup: user study models

- Caching web browser
- Connection Tester
- RADIUS protocol
Backup: TCP model
Backup: model inference comparison

(a) Input log
(b) kTails
(c) Synoptic
(d) Perfume

Perfume: resource-aware model inference
9/14/14