Diagnosing Intermittent Performance Problems

Baron Schwartz

Today's Agenda

- Diagnosing intermittent MySQL problems
  - What kind of problems are we talking about?
  - Why are they hard to solve?
  - What approaches can solve them successfully?
  - What tools can help you do it more quickly?
  - How can you set up and use these tools?
  - How do you interpret the results?
  - Case Studies

Intermittent Problems

- Happen at random times
- Hard to observe in action
- No obvious reason

What Kinds Of Problems?

- In general, we see three kinds
  - Randomly slow query
  - Sudden error message
  - Server-wide “stalls”

What Kinds Of Problems?

- Real customer examples:
  - “My server seems to freeze for ten seconds to a minute at random times. Suddenly, everything clears up again. It seems to happen for no reason.”
  - “I get sporadic 'too many connections' errors. Increasing max_connections doesn't help. This is not related to my peak load.”

How Hard Can It Be?

- It's hard to troubleshoot when you can't see it.
  - “Our graphs show this happens for 1 to 3 minutes once or twice a week.”
- It's hard to get support when it's not reproducible.
  - “Our support staff thinks that we are imagining it.”
  - “We filed a bug, but it was closed because we can't create a test case.”

How Hard Can It Be?

- It can go on forever.
  - “We've been working on this for nearly 5 months.”

Why Does This Happen?

- More CPUs
- More memory
- More popularity
- Cloud computing

How Not To Do It

- DON'T try to use “tuning scripts”
- DON'T try to change server settings
- DON'T try rebooting everything
- DON'T do $random_stab_in_the_dark
- DON'T try upgrading or replacing components

The Fruits of Trial-And-Error

- “I think this might be related to your networking. Can you try buying a new switch?”

The Fruits of Trial-And-Error

- “I think this might be related to your networking. Can you try buying a new switch?”
- “Oh, that didn't solve it? Hmmm..... let me think.”

The Fruits of Trial-And-Error

- “I think this might be related to your networking. Can you try buying a new switch?”
- “Oh, that didn't solve it? Hmmm..... let me think.”
- “I saw someone else on the Internet with a problem like this. They said that switching from Debian to Red Hat fixed it. Can you try that?”

The Fruits of Trial-And-Error

- “It still happens? Oh wow. What version of Java are you using? Can you [upgrade|downgrade] that?”

The Fruits of Trial-And-Error

... Time passes...

- “Sorry, I really don't know. Well, this is a free forum, so at least this didn't cost you anything.”

Measure, Measure, Measure

- You cannot fix what you cannot measure.

How Do I Measure?

- Ideally, measure in three ways:
- Completely.
- Correctly timed.
- Correctly scoped.

What Should I Measure?

- Everything.
  - Yes, it's a lot of data.
  - Schwartz's Law: whatever you don't measure is what you needed to measure.

I Never See It Happen

- You need automatic tools watching for it.
- We've developed good tools for this.

Using Percona Toolkit

- Percona Toolkit = Maatkit + Aspersa
- The primary tools for this are:
  - pt-stalk: wait for something to happen, then gather tons of diagnostic data for a short time
  - pt-sift: look for needles in the pt-stalk haystack

Finding a Trigger

- Find a *reliable* way to detect the problem
- Getting this right is the foundation!
- Use this as a trigger for pt-stalk.

Example

```
$ mysqladmin ext -i1 | awk '/Queries/{q=$4-
qp;qp=$4}/Threads_connected/{tc=$4}/Threads_running/{printf
"%5d %5d %5d\n", q, tc, $4}'

  798   136     7
  767   134     9
  828   134     7
  683   134     7
  784   135     7
  614   134     7
  108   134     24
  187   134     31
  179   134     28
 1179   134     7
 1151   134     7
 1240   135     7
 1000   135     7
```

Example

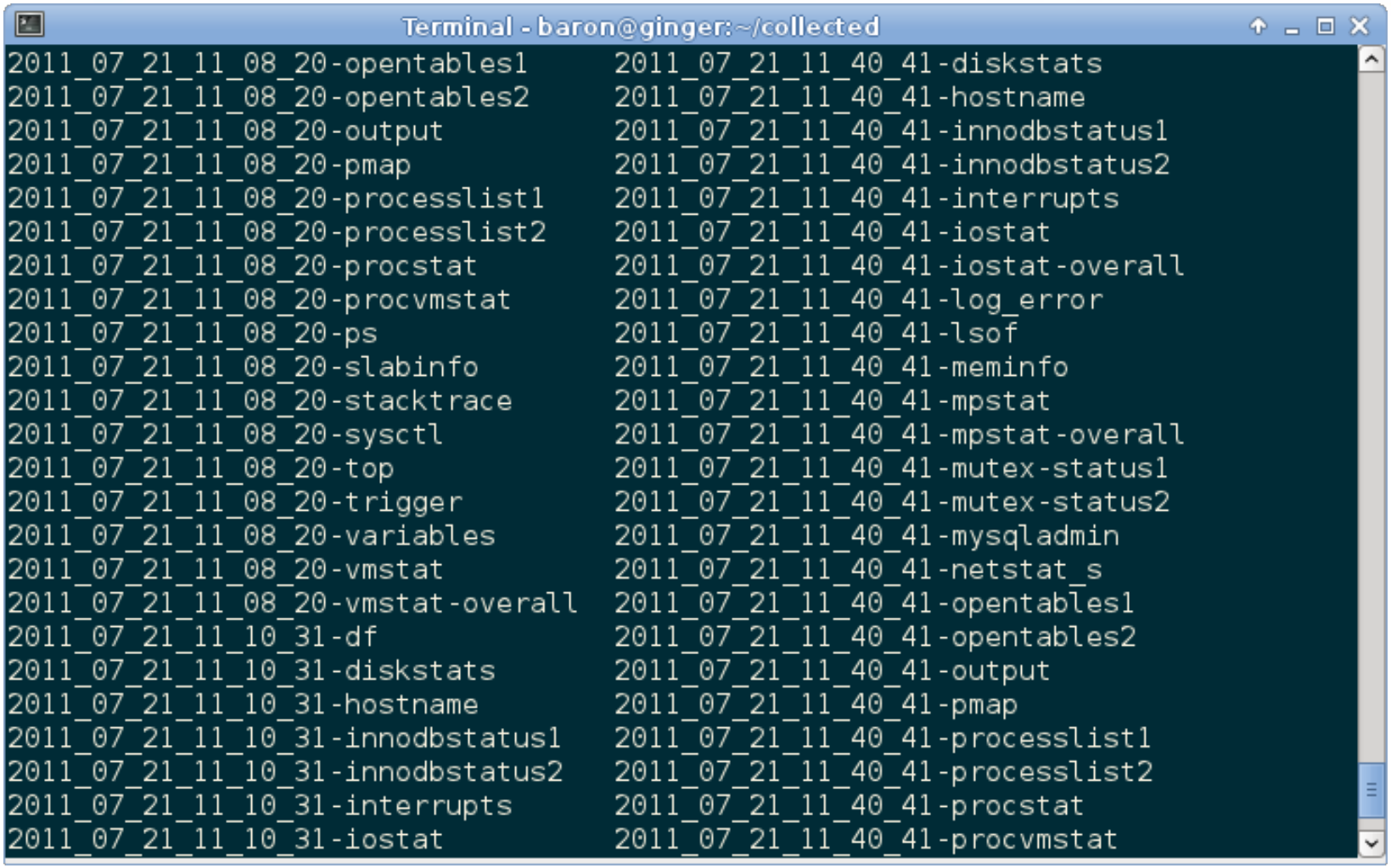
```
$ mysqladmin ext -i1 | awk '/Queries/{q=$4-
qp;qp=$4}/Threads_connected/{tc=$4}/Threads_running/{printf
"%5d %5d %5d\n", q, tc, $4}'

  798   136     7
  767   134     9
  828   134     7
  683   134     7
  784   135     7
  614   134     7
  108   134    24
  187   134    31
  179   134    28
 1179   134     7
 1151   134     7
 1240   135     7
 1000   135     7
```

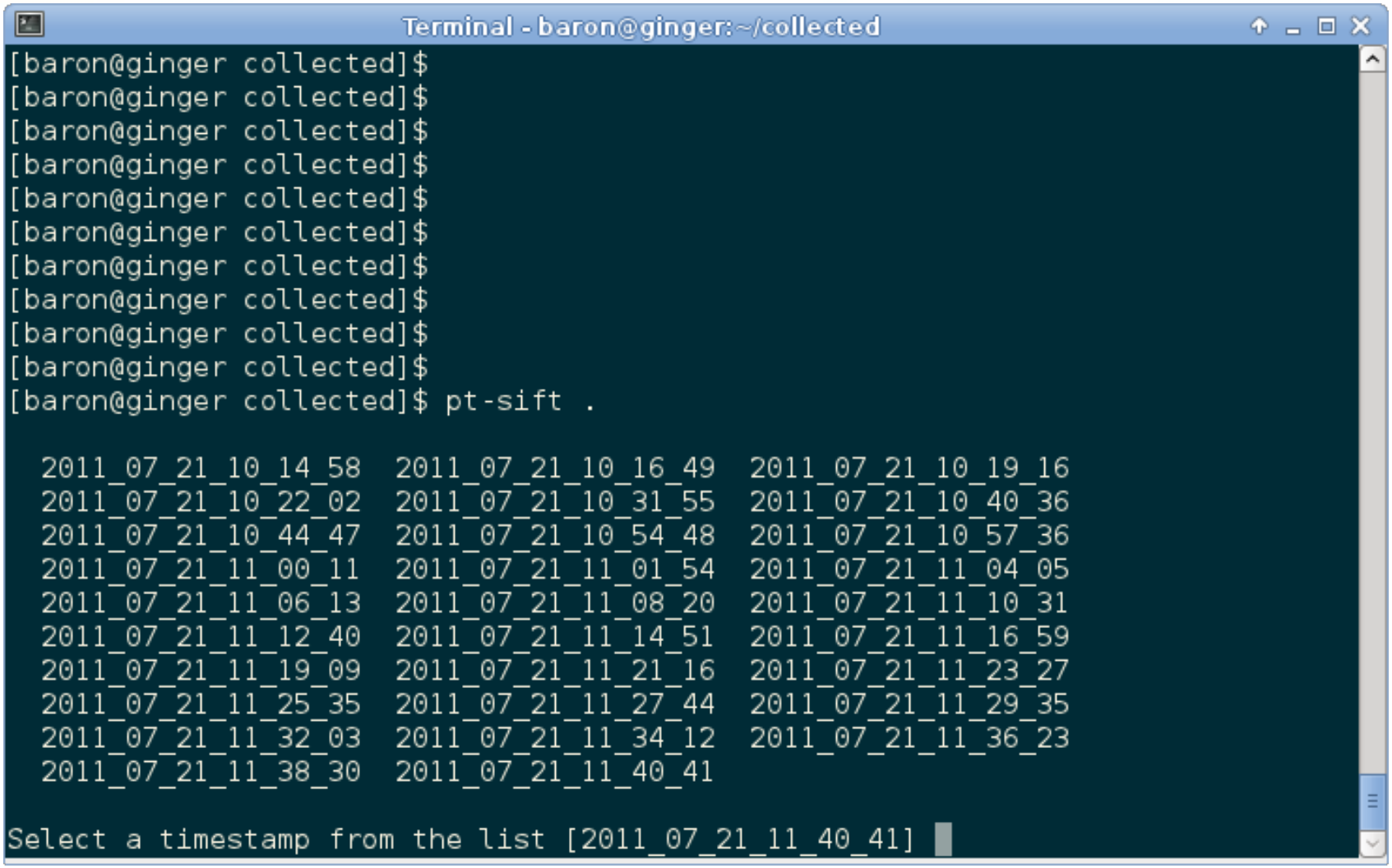
Configuring pt-stalk

- Set --threshold=15
- Set --variable=Threads_running
- Then run pt-stalk as root
- You may need to install and enable:
  - GDB for backtraces (wait analysis)
  - Oprofile for server profiling

Looking At The Data



Using pt-sift



Case Study