



Performance Anti-patterns

Kirk Pepperdine
kirk@kodewerk.com



Our Typical Story

- **Customer JoGoSlo Ltd calls us in desperation**
 - ~1 Million Euros invested
 - Users complain about poor performance
 - Customers consider abandoning the project
- **Developers in a panic**
 - Several man months already invested with no results
 - Can not reproduce the problem in QA
 - Still had some ideas what to do
 - However management has lost confidence
- **We are given 5 days to diagnose problem**
 - How can we approach this problem?

Our Typical Story

- **What we are told to look for**



Our Typical Story

- **What we found**

**Performance
Antipatterns**

Kirk Pepperdine

- **~20 years experience with benchmarking and performance**
- **Engaged worldwide to solve Java performance problems**
- **Co-author of <http://www.javaperformancetuning.com>**
- **Editor at TSS**
- **Member of Sun Java Champion program**





**Java Performance
Tuning**
Oct 23-27, Chania Crete

cretesoft.com

Java Patterns - Java 5 - Java Performance - Ruby on Rails

Performance Anti-patterns

- **Patterns and Antipatterns**
- **Pattern as an Antipattern**
- **Anti-pattern: Shot in the dark**
- **Anti-pattern: No-stress testing**
- **Anti-pattern: Datalite**
- **Questions**

Patterns

- **A software design pattern is a solution to a problem in a context**
 - Useful if the solution occurs over and over again
 - GOF; “descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context”
- **Elements of a pattern**
 - Pattern name
 - Problem
 - Solution
 - Consequences

Patterns

- **Defined by Pattern Language of Patterns (PLoP)**
 - **Template for describing a pattern**
 - **Describes the progression of a solution to proto-pattern to pattern**
- **Patterns Category Catalog**
 - **<http://c2.com/cgi-bin/wiki?CategoryPattern>**
- **Very useful to create compositions of patterns**

Anti-patterns

- **An anti-pattern is a solution to a problem that leads to negative consequences**
 - **Very harmful to create composition of anti-patterns**
 - Evidenced by JoeGoSlo
- **Elements of an anti-pattern**
 - **Anti-pattern name**
 - **Symptoms**
 - **Consequences**
 - **Refactored Solution (could be a pattern)**
- **Context sensitive**
 - **Patterns can morph into anti-pattern (DTO?)**

DTO as an Anti-pattern

- **EJB cannot live outside of the EJB container**
 - **Must call many get operations to get state**
 - **Violation of encapsulation**
 - **Makes poor use of network resource**
 - **Revolving Door antipattern**
- **DTO to the rescue**
 - **Wrap all data in a POJO and send in one shot**
 - **Solves the Revolving Door problem**
 - **Introduces**
 - **New level of marshalling**
 - **Parallel object hierarchy**

DTO as an Anti-pattern

- **DTO in POJO land**
 - **Moves DTO out of context**
 - **Creates a parallel object hierarchy**
 - **Violates DRY**
 - **Increased maintenance**
 - **Introduces yet another level of marshalling/serialization**
 - **Performance antipattern**
- **DTO abuse**
 - **Align mis-matched object schemas**
 - **Design antipattern**

Anti-pattern Resources

- **Introduced in Anti-patterns**
 - <http://www.antipatterns.com/>
- **Loosely documented in the Bitter series**
 - Bitter Java, Bitter EJB (Bruce Tate)
- **Portland Patterns Repository**
 - Not much on performance antipatterns
- **Antipatterns Category Catalog**
 - <http://c2.com/cgi-bin/wiki?AntiPatternsCatalog>
- **Supported by the Antipattern template**
 - <http://c2.com/cgi-bin/wiki?AntiPatternTemplate>

Anti-Pattern: Shot in the Dark

- *Name:* **Shot in the Dark**
- *Most Applicable Scale:* **Application**
- *Refactored Solution Name:* **Measure, don't guess**
- *Refactored Solution Type:* **Tooling, Training**
- *Root Causes:* **Ignorance, Haste (motivated by stress)**
- *Unbalanced forces:* **Management of Performance**
- *Anecdotal Evidence:* **We can't reproduce the problem in QA but I think it is caused by feature "X".**

Shot in the Dark

- **What we found at JoeGoSlo**
 - A poorly performing web based application
 - Unable to reproduce problem in QA environment
- **Tuning was left to developers**
 - Did what developers do, look at code and fix it
 - Conjecture that problem was with database interactions
 - Ignored key pieces of evidence
 - Database group claimed database was not root of problem
 - Batch processes worked without issues
- **Developed a poor track record by guessing at possible problems**

Shot in the Dark Side Effects

- **Finger pointing**
 - **DB group claims DB works fine**
 - **Networking claimed network was fine**
 - **And so on**
- **Loss of credibility**
 - **Management has lost trust**
 - **Users have lost faith**
 - **Developers lose the ability to affect change**
- **Pressure from users + Pressure from management = Lots of stress**

Shot in the Dark Side Effects

● **Effects of Stress**

- **Learning and problem solving capabilities are impaired**
 - **During a stressful event, catecholamines suppress activity in areas at the front of the brain inhibiting the ability to handle complex social or intellectual tasks and behaviors**
- **Why Zebras don't get Ulcers (Robert M. Sapolsky)**
- **Other "tribal" behaviors**

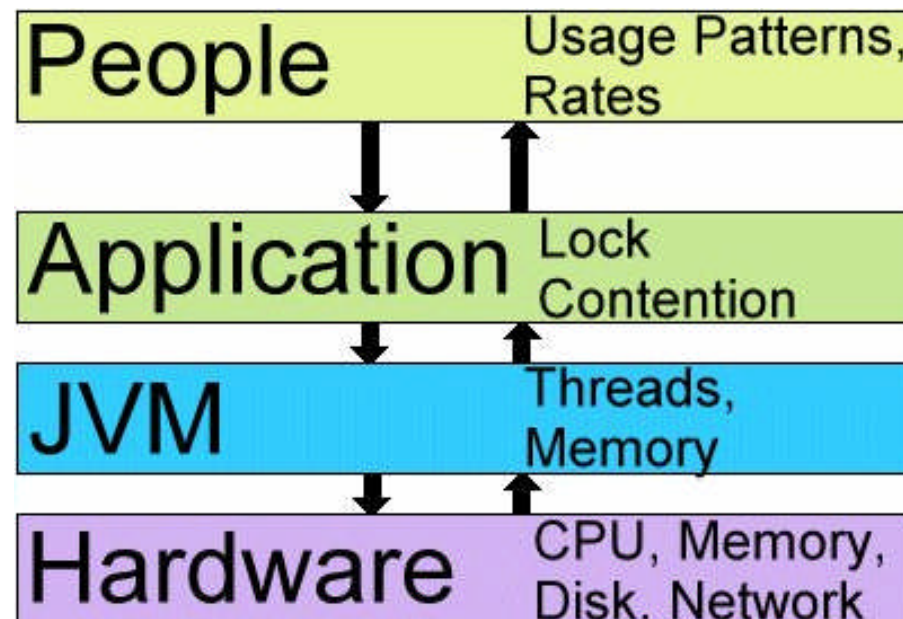
● **Skills needed are impaired by the situation**

- **Developers rely on what they already know**
 - **They look at code**
 - **May or may not discover the problem**

Shot in the Dark Unbalanced Forces

- **Performance management**

- **Need to consider the entire system including dynamic forces**



Shot in the Dark: Refactored Solution

Measure, Don't Guess

Shot in the Dark: Refactored Solution

- **Solid measurements help you**
 - **Focus efforts**
 - **Facilitate planning**
 - **Instill confidence**
 - **Deflect finger pointing**
 - **Reduce stress**
- **How**
 - **By showing you what needs to be done**

Shot in the Dark

IronEyeSQL

File View Server Help

Connect Disconnect Config Purge Import Export About

Legend: Preparation (green), Execution (yellow), Retrieval (orange), Slowest (red), Most Run (blue), Both (purple)

Filtering (click to open)

SQL	Count	Avg Time	Max Time
SELECT LASTUSERID, OIDFIRMA, OIDYEARVALIDFROM, OIDBASEC...	30735	2	47
SELECT LASTUSERID, OIDFIRMA, OIDYEARVALIDFROM, OIDBASEC...	30735	2	47
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	30726	2	78
SELECT LASTUSERID, OIDCOSTCENTRE, OIDBUDGETCT, CALCU...	1031	10	16
SELECT LASTUSERID, OIDCOMPANY, OIDROOT, GROUPTYPE, CO...	228	2	16
SELECT LASTUSERID, OIDGROUP, OIDCOSTTYPE, ITEMPOSITION ...	226	2	16
SELECT LASTUSERID, OIDPARENT, OIDLANGUAGE, TEXT, OBJEC...	73	1	16
SELECT LASTUSERID, OIDGROUP, OIDCOSTCENTRE, ITEMPOSITI...	73	3	16
SELECT LASTUSERID, OIDGROUP, OIDCOSTCENTRE, ITEMPOSITI...	73	3	16
SELECT LASTUSERID, OIDCOSTTYPE, OIDCOSTCENTREBASE, OB...	73	2	16
SELECT LASTUSERID, OIDCOMPANY, MANAGER, VALIDFROM, VALI...	73	2	16
SELECT LASTUSERID, OIDCOSTCENTRE, CODE, SHORTNAME, VA...	73	1	16
SELECT LASTUSERID, SETTINGUSER, SETTINGFILE, SETTINGSEC...	61	2	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	1	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	2	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	2	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	1	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	2	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	2	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	3	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	2	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	3	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	2	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	2	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	2	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	2	16
SELECT LASTUSERID, OIDBUDGETROW, OIDCURRENCY, BALANC...	54	2	16

SQL Statement Syntax: OIDACCUMLATIVECC, ISEXTENDED AUDIT, ISCHANGEABLE, ISAUTOREVALUATION, TARGETCOSTSCIS, CONCURRENTCULC, OBJECTTS, OBJECTID FROM KORESETUP_V WHERE (OIDFIRMA = ?)

Rows Returned: max avg min

Time Performance: max avg min

Count: no. of executions

Data loaded from V:\barcelona\logstvarial\server\spy4.log

Not Connected

Shot in the Dark

IronEyeSQL

File View Server Help

Connect Disconnect Config Purge Import Export About

Legend: Preparation (light blue), Execution (light green), Retrieval (yellow), Slowest (light blue with icon), Most Run (light green with icon), Both (yellow with icon)

Filtering (click to open)

SQL	Count	Avg Time	Max Time
SELECT LASTUSERID, OIDGROUP, OI...	1	1250	1250
SELECT LASTUSERID, OIDGROUP, OI...	1	1250	1250
SELECT LASTUSERID, OIDGROUP, OI...	1	1250	1250
SELECT LASTUSERID, OIDGROUP, OI...	5	1213	1250
SELECT LASTUSERID, OIDGROUP, OI...	1	1157	1157
SELECT LASTUSERID, OIDGROUP, OI...	1	1156	1156
SELECT LASTUSERID, OIDGROUP, OI...	1	297	297
SELECT LASTUSERID, OIDGROUP, OI...	1	297	297
SELECT LASTUSERID, OIDGROUP, OI...	1	156	156
SELECT LASTUSERID, OIDGROUP, OI...	1	156	156
SELECT LASTUSERID, OIDPARENT, OI...	2	31	31
SELECT LASTUSERID, LANGUAGE, RES...	1	31	31
SELECT LASTUSERID, OIDPARENT, OI...	4	24	31
SELECT LASTUSERID, OIDPARENT, OI...	4	16	31
SELECT LASTUSERID, CODE, SHORTNA...	4	16	31
SELECT LASTUSERID, OIDPARENT, OI...	4	16	31
SELECT LASTUSERID, OIDPARENT, OI...	4	16	32
SELECT LASTUSERID, CODE, ADDRESS...	4	16	31
SELECT LASTUSERID, OIDPARENT, OI...	4	16	31
SELECT LASTUSERID, OIDPARENT, OI...	4	16	31
SELECT LASTUSERID, OIDPARENT, OI...	4	16	31
SELECT LASTUSERID, CODE, SHORTNA...	4	16	16
SELECT LASTUSERID, OIDPARENT, OI...	4	16	31
SELECT LASTUSERID, CODE, SHORTNA...	4	16	31
SELECT LASTUSERID, CODE, SHORTNA...	4	16	31
SELECT LASTUSERID, LCID, SHORTNA...	2	16	16
SELECT LASTUSERID, OIDGROUP, OI...	1	16	16

executed 2 time(s)

SQL Statement Syntax: `OIDACCUMLATIVECC, ISEXTEDEDAUDIT, ISCHANGEABLE, ISAUTOREVALUATION, TARGETCOSTSCIS, CONCURRENTCUCALC, OBJECTTS, OBJECTID FROM KORESETUP_V WHERE (OI...`

Rows Returned: max avg min

Time Performance: max avg min

Count: no. of executions

Data loaded from V:\barcelona\logs\varial\server\spy4.log

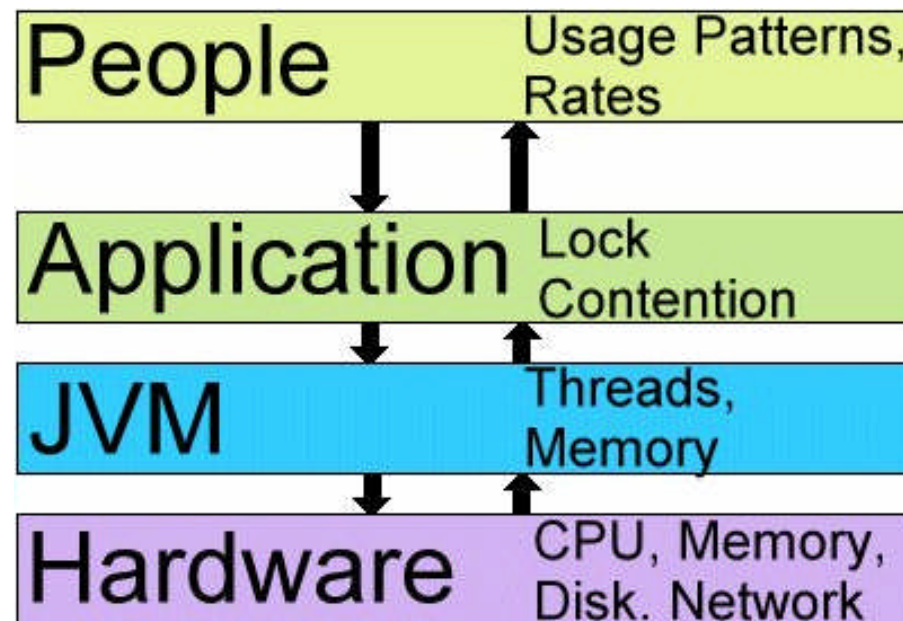
Not Connected

Shot in the Dark: Refactored Solution

- **Make sure you**
 - **Review or set the performance targets**
 - **Know what the people are doing**
 - **Visit floor**
 - **Examine logs**
 - **Perform a layer by layer performance investigation**
 - **Make measurements at each layer**
 - **Start with hardware**
 - **Work up through the stack until overextended resource is identified**

Shot in the Dark Refactored Solution

- **Each layer will eliminate a set of constraining resources**



Composition of Antipatterns

- **JoeGoSlo attempted to measure but failed**
 - **Why?**
- **Faced a composition of performance antipatterns**
 - **Demonstrated in upcoming sections**
- **Masked by Shot in the Dark**
 - **Needed skills were impaired**
 - **Learning**
 - **Problem solving**
 - **Reduced to relying on known**
 - **Wrong Skill set**

Shot in the Dark

```
public static void testDataset(String dataset) throws IOException {  
    DataInputStream rdr = new DataInputStream( dataset);  
    long starttime = System.currentTimeMillis();  
    int truecount = 0;  
    String s;  
    try {  
        while ((s = rdr.readUTF()) != null) {  
            if (checkInteger(s)) truecount++;  
        }  
    } catch (EOFException e) {}  
    rdr.close();  
    System.out.println(truecount + " (count); time " +  
        (System.currentTimeMillis() - starttime));  
}
```

Shot in the Dark

```
public static boolean checkInteger(String testInteger) {  
    try {  
        Integer theInteger = new Integer(testInteger); // fails if not a number  
        return  
            (theInteger.toString() != "") && // not empty  
            (theInteger.intValue() > 10) && // greater than ten  
            ((theInteger.intValue() >= 2) &&  
            (theInteger.intValue() <= 100000)) && // 2 >= X <= 100000  
            (theInteger.toString().charAt(0) == '3'); // first digit is 3  
    } catch (NumberFormatException err) {  
        return false;  
    }  
}
```

Anti-pattern No-stress Testing

- *Name:* **No-stress Testing**
- Most Applicable Scale: **Application, Component**
- *Refactored Solution Name:* **Stress testing**
- *Refactored Solution Type:* **Tooling, Training**
- *Root Causes:* **Haste, Sloth**
- *Unbalanced forces:* **Management of Performance**
- *Anecdotal Evidence:* **Performance was ok when I tried it in development**

No-stress Testing

- **What we found at JoeGoSlo**
 - **Use of browser to stress test application**
 - **Difficult to maintain load**
 - **Test were unrepeatable**
 - **Difficult to reproduce problem**
 - **Difficult to verify success**
- **Introduction of a stress test harness**
 - **Controls load**
 - **Results in repeatable tests**
 - **Increase chances of success**

Antipatterns

- **Test Harness Features**
 - **Easily scripted to support many users doing many different things**
 - **Supports randomization of inputs**
 - **Throttles request rates**
 - **Randomized request rates**
 - **Reports on response times**
 - **Varies load**
 - **Can generate high loads**
- **Introduced Apache JMeter with a mentoring program**

Apache JMeter

The screenshot displays the Apache JMeter graphical user interface. On the left, a tree view shows the test plan structure: Test Plan, Thread Group, HTTP Request, Browser-derived headers, Wildcard parameters, /tips/keyword, Browser-derived headers, Keyword parameters, Gaussian Random Timer, WorkBench, and HTTP Proxy Server. The main panel is titled 'HTTP Request' and contains the following configuration fields:

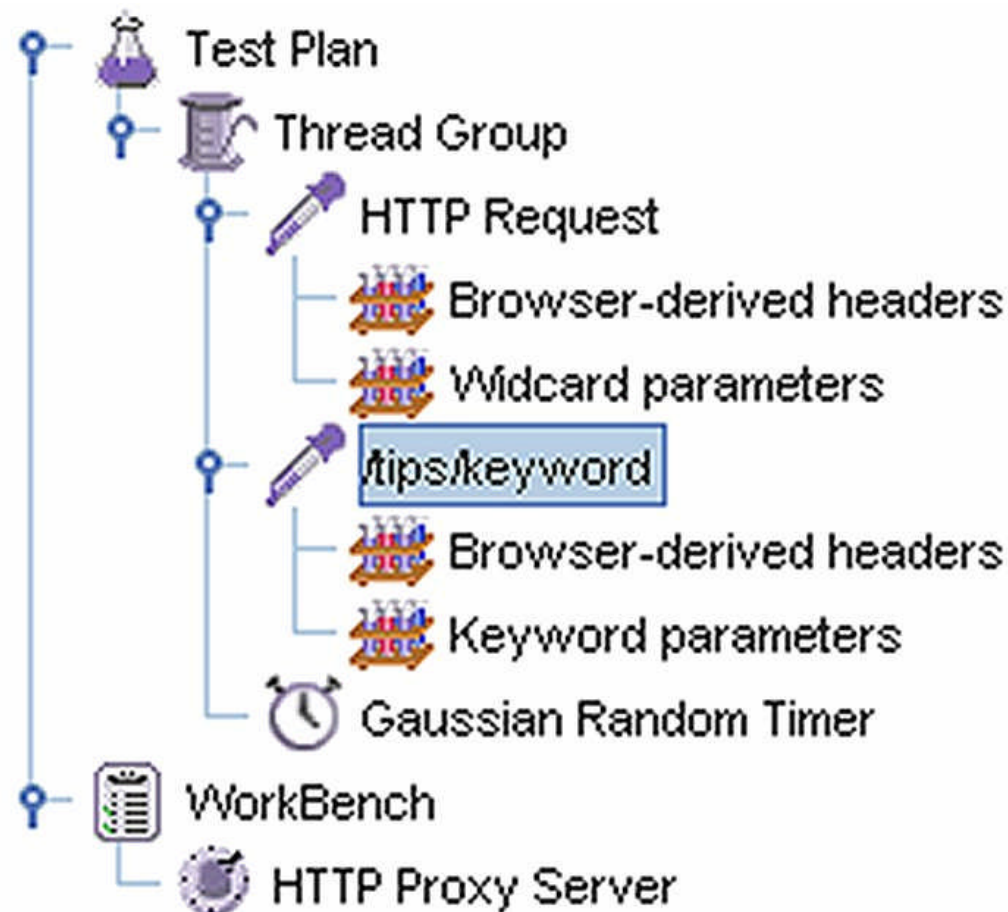
- Name:** /tips/keyword
- Web Server:** Server Name or IP: localhost, Port Number: 8080
- HTTP Request:** Protocol: http, Method: GET (selected), POST
- Path:** /tips/keyword
- Redirect Automatically, Follow Redirects, Use KeepAlive
- Send Parameters With the Request:**

Name:	Value	Encode?	Include Equals?
keyword	\${keyword}	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Buttons: Add, Delete

- Send a File With the Request:** Filename: [text box], Browse... button
- Parameter Name: [text box]
- MIME Type: [text box]
- Optional Tasks:** Retrieve All Embedded Resources from HTML Files, Use as Monitor

Apache JMeter



Demo: Benchmarking with JMeter

● **Tips application**

- **Returns a list of documents that contain keyword**
- **Two types**
 - **Wildcard performs an in-memory search**
 - **Keyword searching in a database**

● **Performance requirement**

- **Queries need to complete in under 1 second**

Demo: Benchmarking with JMeter

- **JoeGoSlo still can't see a problem**
 - **Look for next antipattern**
 - **Observation, QA database is a subset of that in production**

Anti-pattern: Datalite

- **Name:** Datalite
- **Most Applicable Scale:** Application, Component
- **Refactored Solution Name:** Realistic test environment
- **Refactored Solution Type:** Configuration
- **Root Causes:** Haste, Sloth, Complexity
- **Unbalanced forces:** Management of Performance
- **Anecdotal Evidence:** We can't reproduce the problem in QA.

Antipattern: Datalite

- **Production environment?**
 - **Not desirable and usually not an option**
- **QA environment should**
 - **Perfectly resemble your production environment**
 - **Data sizes, memory sizes, cache sizes, disk speeds, network speeds, should be the same**
 - **May need to consider the “when”**
 - **Account for interference from other system**
 - **Needs to a controlled mix-in.**
 - **Sometimes have to add external elements to test**
 - **Maybe able to simulate but this must be done with care**

Antipattern: Datalite

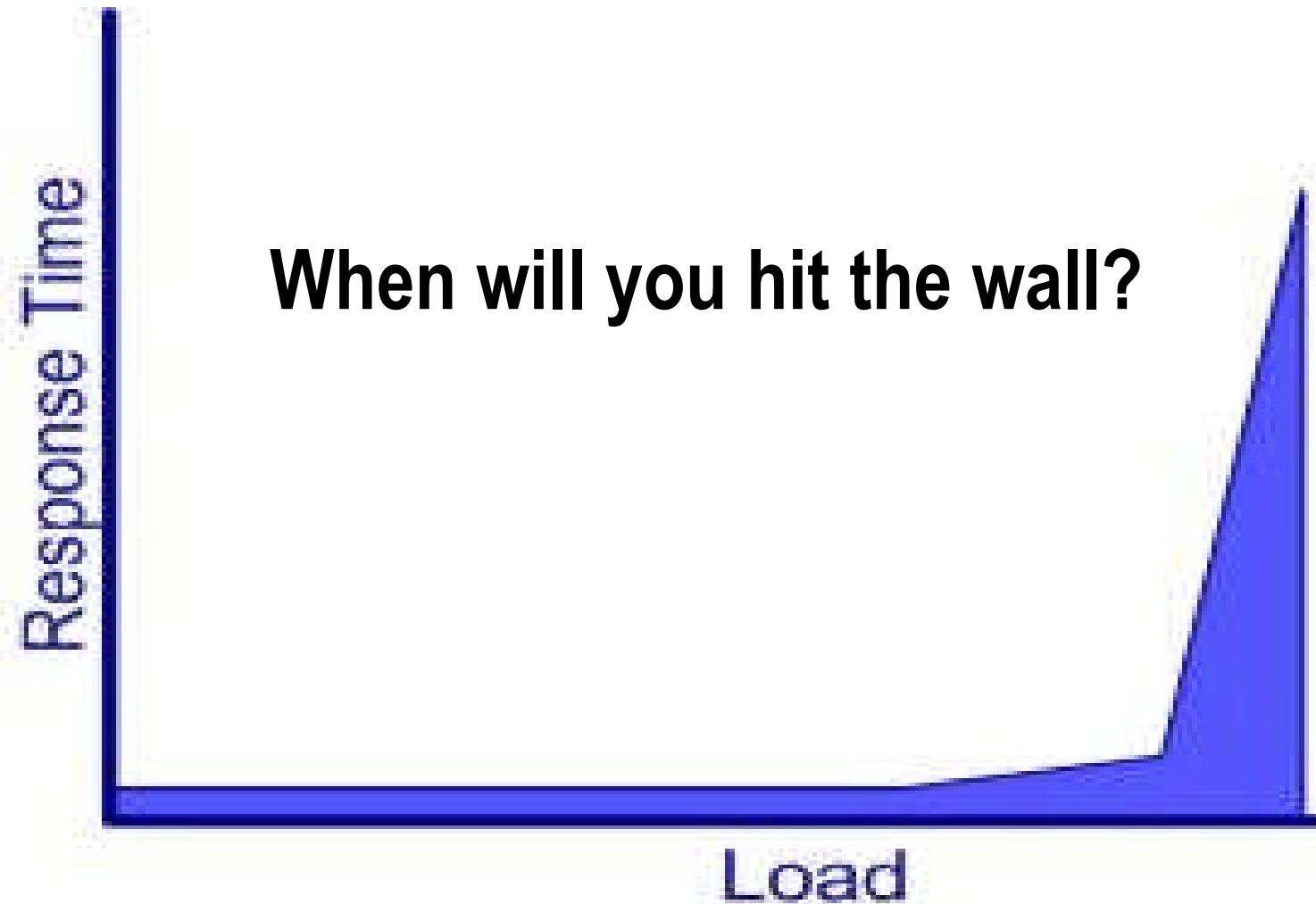
● Caching

- **Protects your application from an underlying slower technology**
- **Increases probability query will not have to visit disk**
- **Reduces response times**
- **May reduce the effects of I/O (network disk), GC**

● Don't extrapolate!

- **Difficult to know when you will hit the wall**
- **Consider effect of moving from gigabit to 10Megabit network when application uses 15Megabits of bandwidth**
 - **Shifts the bottleneck**

Extrapolation



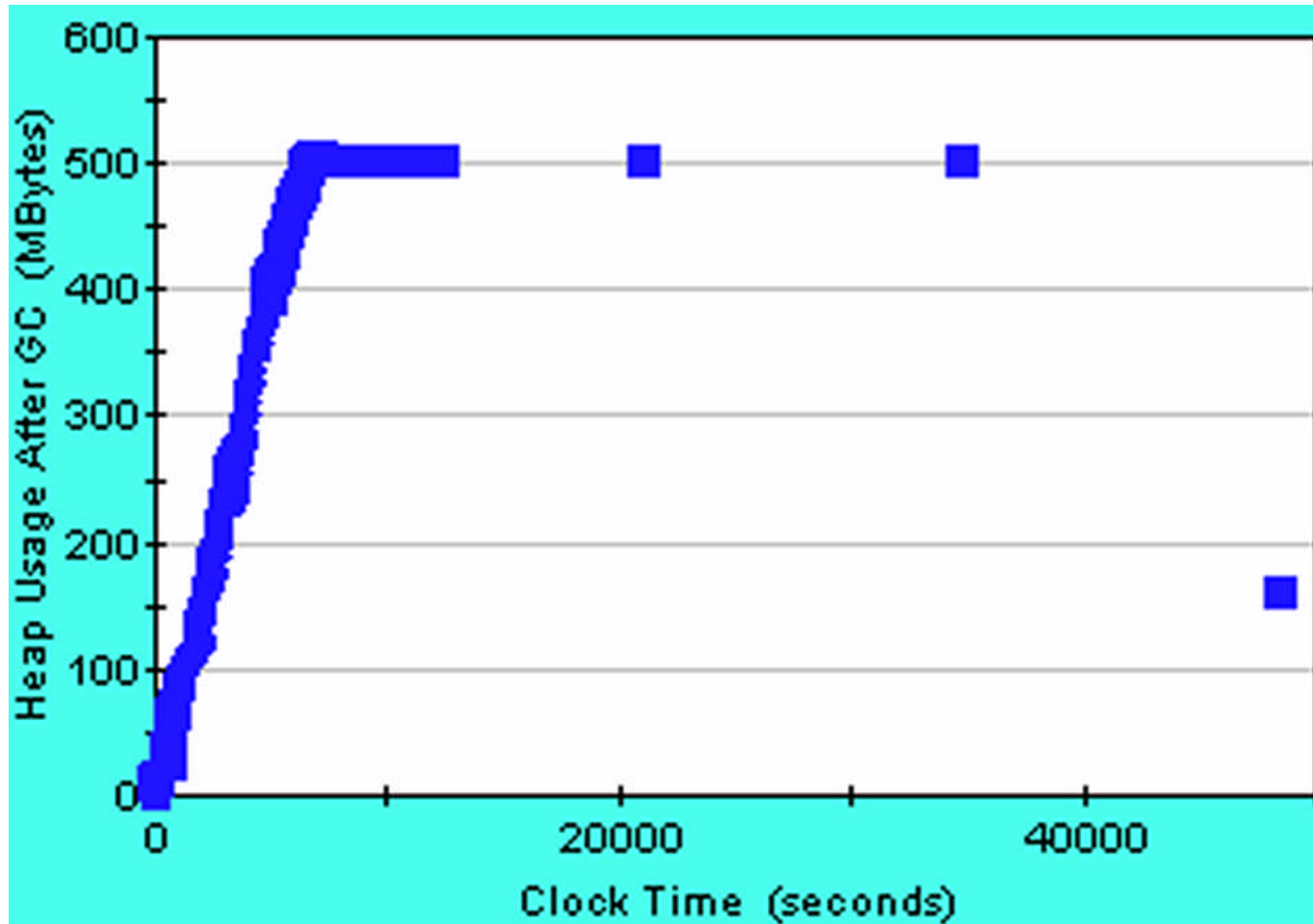
Antipattern: Datalite

- **Test with a complete system**
 - **Stress test harness takes role of users**
 - **Application, JVM are configured identically**
 - **Includes external systems (i.e. database)**
 - **Hardware is identical**
- **Problem is reproducible**
 - **Implies problem is identifiable and can be fixed**

Antipattern: Datalite

- **What we found at JoeGoSlow**
 - **Developers focused on O/R mapping code**
 - **Code was ugly**
 - **Worried about the effects of ripple-loading**
 - **Reduced dataset in the database**
 - **Less data loaded per query**
 - **Testing environment hid the real problem**
 - **They ignored one insignificant but key piece of information**
 - **The application sometimes recovered**
- **We fixed JoeGoSlo's data source and retested**
 - **Turned on low impact memory management monitoring**
 - **-verbose:gc (-Xloggc:gc.log)**

Antipattern: Datalite



JoeGoSlo no more

- **Performance degradation was due to memory starvation**
 - **Confirmed initial hypothesis**
- **Root cause was improper use of HttpSession**
 - **Complete surprise to everyone**
 - **Refocused efforts in the right direction**
 - **Solution was implemented in minutes**
 - **Solution was validated in minutes**

Joegoslow Composition of Anti-patterns

- **Anti-pattern: Datalite**
 - Used small dataset in QA
 - Replicated product database into the QA environment
- **Anti-pattern: No-stress testing**
 - Used browser to stress application
 - Introduced JMeter to apply real repeatable stress
- **Anti-pattern: Shot in the Dark**
 - Guessed that problem was with ripple loading
 - Measured that problem was a session leak

Questions

kirk@javaperformancetuning.com
www.javaperformancetuning.com