



TERRACOTTA

Krzysztof Kliś

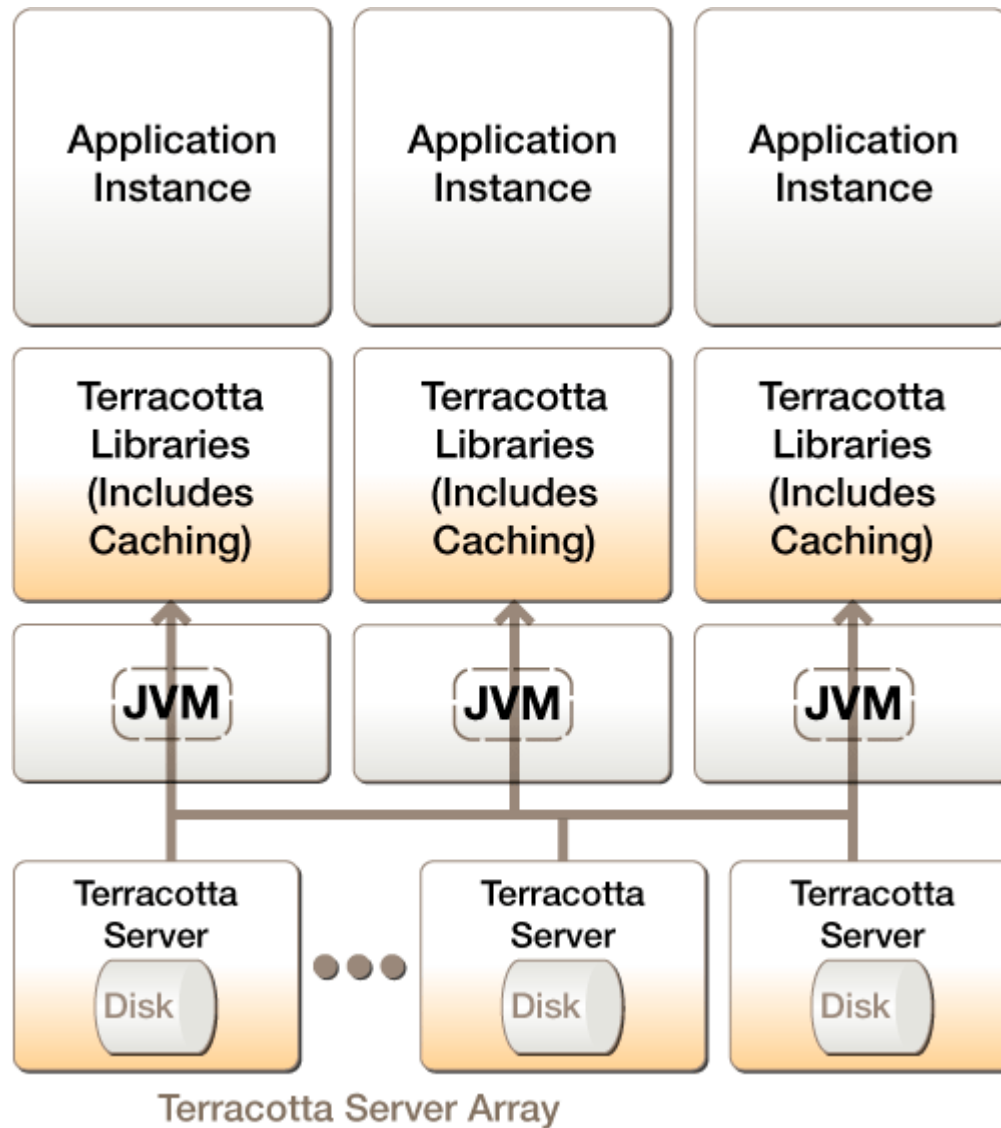
krzysztof.klis@gmail.com

Czym jest Terracotta?

Terracotta jest oprogramowaniem open source pozwalającym na proste skalowanie w poziomie aplikacji działających na platformie Java, bez konieczności wykorzystywania dodatkowego kodu lub zewnętrznych baz danych.

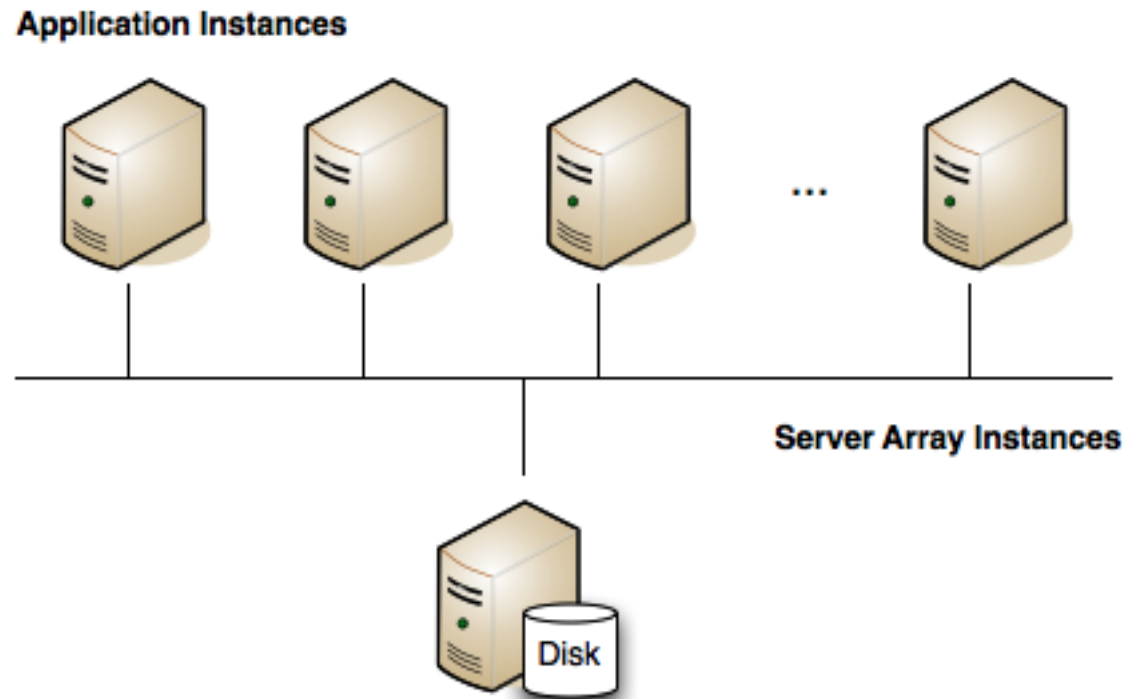
Terracotta wykorzystuje technologię Network-Attached Memory (NAM) do łączenia maszyn wirtualnych Javy w klastry, a samo klastrowanie infrastruktury programowej odbywa się na poziomie poniżej warstwy aplikacji.

Architektura



Pojedynczy serwer obiektów

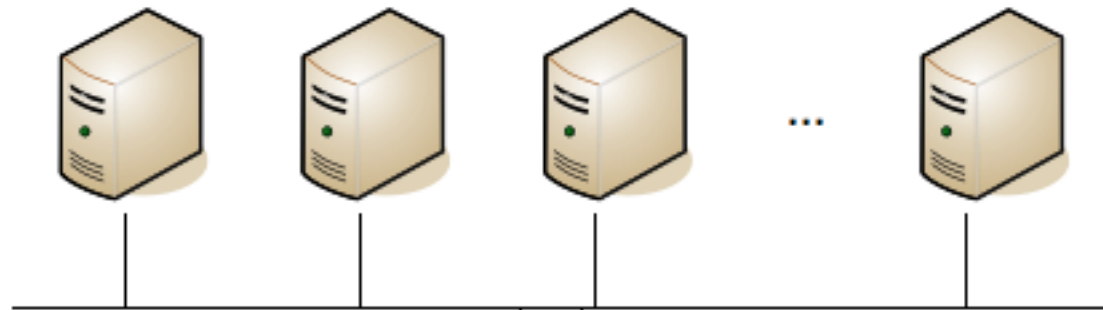
Fig. 3: Reliable Single Server — Data Saved to Disk



Scenariusz z serwerem zapasowym

Fig. 4: Available and Reliable – Failover, Data Saved

Application Instances



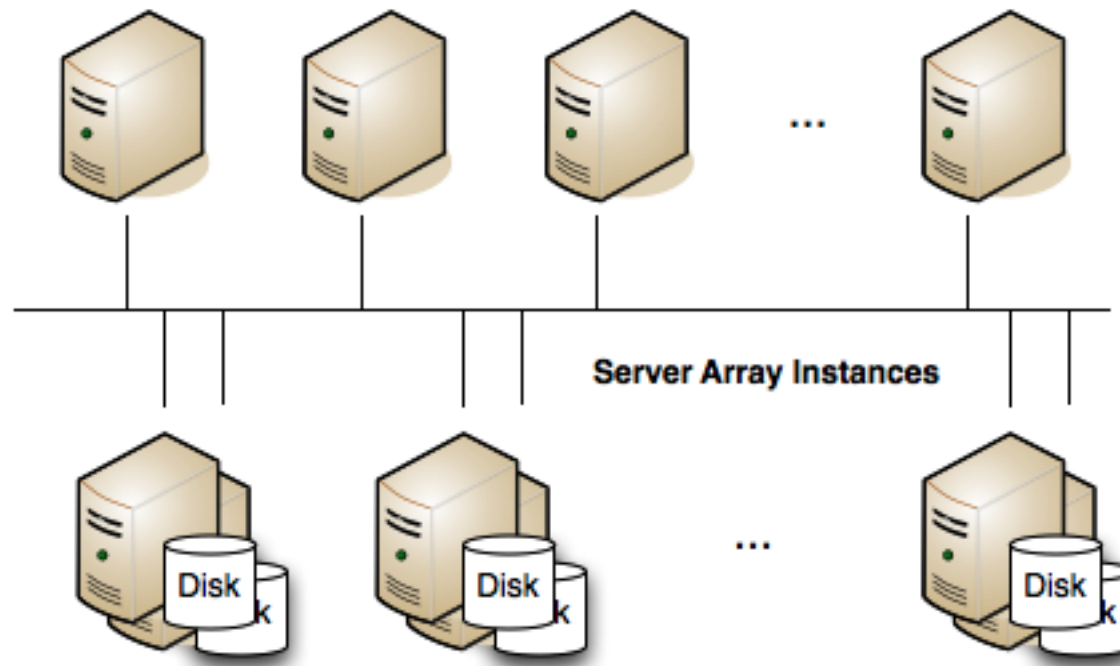
Server Array Instances



Macierz serwerów Terracotty

Fig. 5: Available, Reliable, and Scalable — Failover, Data Saved, Capacity Raised

Application Instances



Przykładowy kod w Javie

```
public class Test<T> {  
  
    private List<T> list = new ArrayList<T>();  
  
    public void add(T obj) {  
        synchronized (list) {  
            list.add(obj);  
        }  
    }  
  
    public List<T> get() {  
        return list;  
    }  
  
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<con:tc-config xmlns:con="http://www.terracotta.org/config">
  <servers>
    <server host="%i" name="localhost">
      <dso-port>9510</dso-port>
      <jmx-port>9520</jmx-port>
      <data>terracotta/server-data</data>
      <logs>terracotta/server-logs</logs>
      <statistics>terracotta/cluster-statistics</statistics>
    </server>
  </servers>
  <clients>
    <logs>terracotta/client-logs</logs>
    <statistics>terracotta/client-statistics/%D</statistics>
  </clients>
  <application>
    <dso>
      <instrumented-classes>
        <include>
          <class-expression>pl.poznan.jug.terracotta.Test</class-expression>
        </include>
      </instrumented-classes>
      <roots>
        <root>
          <field-name>pl.poznan.jug.terracotta.Test.list</field-name>
        </root>
      </roots>
      <locks>
        <autolock>
          <method-expression>void pl.poznan.jug.terracotta.Test.add(*)</method-expression>
          <lock-level>write</lock-level>
        </autolock>
      </locks>
    </dso>
  </application>
</con:tc-config>
```

```
public class TestPut {  
  
    public static void main(String[] args) throws InterruptedException {  
        Test<Integer> t = new Test<Integer>();  
        for (int i = 0; i < 20; i++) {  
            t.add(i);  
            System.out.println(i);  
            Thread.sleep(3000);  
        }  
    }  
}
```

```
public class TestGet {  
  
    public static void main(String[] args) throws InterruptedException {  
        Test<Integer> t = new Test<Integer>();  
        while (true) {  
            System.out.println(t.get());  
            Thread.sleep(1000);  
        }  
    }  
}
```

Co można klastrować?

- Dowolne obiekty, które można utworzyć w JVM.
- Obiekty standardowych klas Javy są klastrowane przez Terracottę bez dodatkowych zabiegów ze strony programisty.
- Obiekty własnych klas mogą być klastrowane o ile zostaną zintegrowane z Terracottą za pomocą mechanizmu Terracotta Integration Modules.

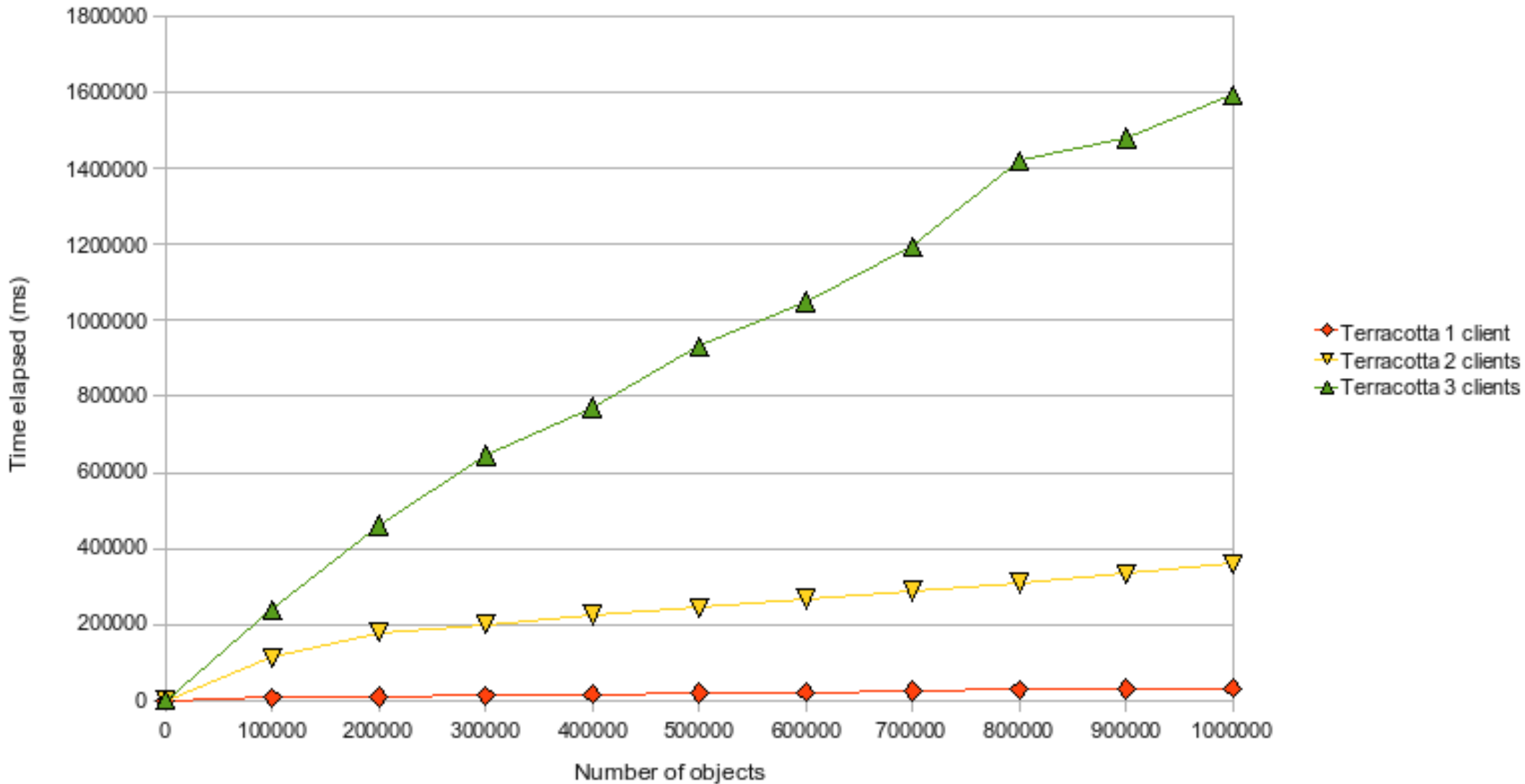
Terracotta Integration Modules

- Apache Commons Collections
- Apache Struts
- BEA WebLogic
- CGLIB
- EHCache
- Geronimo
- Glassbox
- GlassFish
- Hibernate
- iBatis
- IBM JDK
- IBM WebSphere
- IBM WebSphere CE
- JBoss Application Server
- JBoss Cache
- Jetty
- Lucene
- Maven - Terracotta Plugin
- Quartz
- Rife
- Spring
- Sun Hotspot
- Tree Map Cache
- Tomcat
- Wicket

Nie tylko Java

- Clojure (v1.0 – Paul Stadig)
- Groovy (Grails Terracotta Plugin)
- PHP (Quercus + Resin)
- Ruby (JMaglev – Fabio Kung)
- Scala (Scala Actors TIM - Jonas Bonér)
- Scheme (JScheme, Kawa – ja :-))

Wydajność



| | Client #1 | Client #2 | Client #3 |
|-----------|------------------|------------------|------------------|
| 100 000 | 79 970 | 356 630 | 277 430 |
| 200 000 | 144 546 | 627 075 | 609 564 |
| 300 000 | 209 773 | 740 245 | 982 688 |
| 400 000 | 278 809 | 849 758 | 1 183 495 |
| 500 000 | 358 900 | 1 104 308 | 1 334 393 |
| 600 000 | 440 485 | 1 259 875 | 1 446 086 |
| 700 000 | 515 677 | 1 513 364 | 1 556 718 |
| 800 000 | 589 474 | 1 911 247 | 1 761 099 |
| 900 000 | 690 115 | 1 913 275 | 1 841 012 |
| 1 000 000 | 958 560 | 1 915 311 | 1 911 430 |

Terracotta Public License

- Oparta na Mozilla Public License v1.1.
- Usunięta definicja użytku komercyjnego.
- Termin „oryginalny kod” dotyczy zarówno kodu źródłowego jak i kodu wykonywalnego.
- Termin „modyfikacja” obejmuje również dodanie nowego pliku do Terracotty.
- W przypadku dystrybucji oprogramowania zawierającego Terracottę należy w widocznym miejscu interfejsu użytkownika umieścić frazę „Powered by Terracotta”. Wymóg ten nie dotyczy aplikacji hostowanych korzystających z Terracotty.

Przydatne odnośniki

- Terracotta Inc. <http://www.terracotta.org/>
- Terracotta Forge Labs <http://www.terracotta.org/confluence/display/labs/>
- Terracotta Eclipse Plugin <http://download.terracotta.org/eclipse/update>
- Grails Terracotta Plugin <http://www.grails.org/Terracotta+Plugin>
- Terracotta for Quercus / Resin <http://wiki.caucho.com/Terracotta>
- Blog Alexa Millera <http://tech.puredanger.com/>
- Blog Jonasa Bonéra <http://jonasboner.com/>
- Blog Paula Stadiga <http://paul.stadig.name/>
- Blog Fabio Kunga <http://fabiokung.com/>
- Mój blog :-)
<http://weblambdazero.blogspot.com/>