Java Technologies

IBM Monitoring and Diagnostic Tools for Java™ ...

Brian Peacock
IBM Java Technology Centre

Agenda

- **Introduce the family of Java consumability tools**
- **Usage scenarios**
- **Health Center**
  - Overview
  - Demo

The IBM Monitoring and Diagnostic Tools for Java™ - family

- **Diagnostics Collector**
- **Health Center**
- **Dump Analyzer**
- **Garbage Collection and Memory Visualizer (GCMV)**

Plus ...

Close relatives

- IBM Diagnostics Guide
- IBM Guided Activity Assistant (IGAA)
- Eclipse Memory Analyzer Tool (MAT)
- Diagnostics Tool Framework for Java (DTFJ)
- IBM Support Assistant (ISA)

Scenarios - “How do I ... ?”

- **Diagnostics Guide**
  - The manual you all know and love
  - Tells you about Java options
  - Explains diagnosis procedures
- **IGAA**
  - Guides you through problem scenarios
    - eg. I have OutOfMemory ... what do I do now ?
  - Also available in InfoCenter format

Scenarios – Artifact Analysis

- **Diagnostics Collector**
  - Where is that dump/trace ?
  - What do you mean I forgot to send you xxxx ?
- **Dump Analyzer**
  - Err ... what does this dump actually show me ?
- **GCMV**
  - Garbage Collection ... that's an internal JVM thing that I don't need to worry about ... isn't it ?
- **MAT**
  - OpenSource heap analyzer
  - IBM provide adapters to read IBM Java dumps

Scenarios – misc items

- **DTFJ**
  - Common API to read Java dumps
- **IBM Support Assistant**
  - You want an IBM tool ... this is the place to go
  - You have a software problem ... this is the place to go

Scenarios – Live Monitoring and Analysis

- **Health Center**

Typical customer questions

- What is my Java application doing ?
- Why is it doing that ?
- Why is my application going so slowly ?
- How can I make it go faster ?
- Is my application scaling well ?
- Is our algorithm sensible ?
- Do we need to tune the JVM ?

... and some pretty tricky questions ...

- **Is my configuration sensible ?**
- **Is the system stable ?**
- **Have I got a memory leak ?**
- **Is the application about to crash ... leading to a very awkward conversation with my boss ?**

... which boil down to ...

- **Is my application healthy ?**
- **If not, how can I fix it ?**

DO *you* have any similar questions that you want our tooling to answer ?

Health Center

- **New tooling from the IBM Java Technology Center**
  - Currently available in beta
- **Provides a view inside a running Java application**
  - What the application is doing
  - Performance analysis
  - Configuration recommendations
  - Configuration recommendations
    - (so important it's worth mentioning it twice)

Target Environments

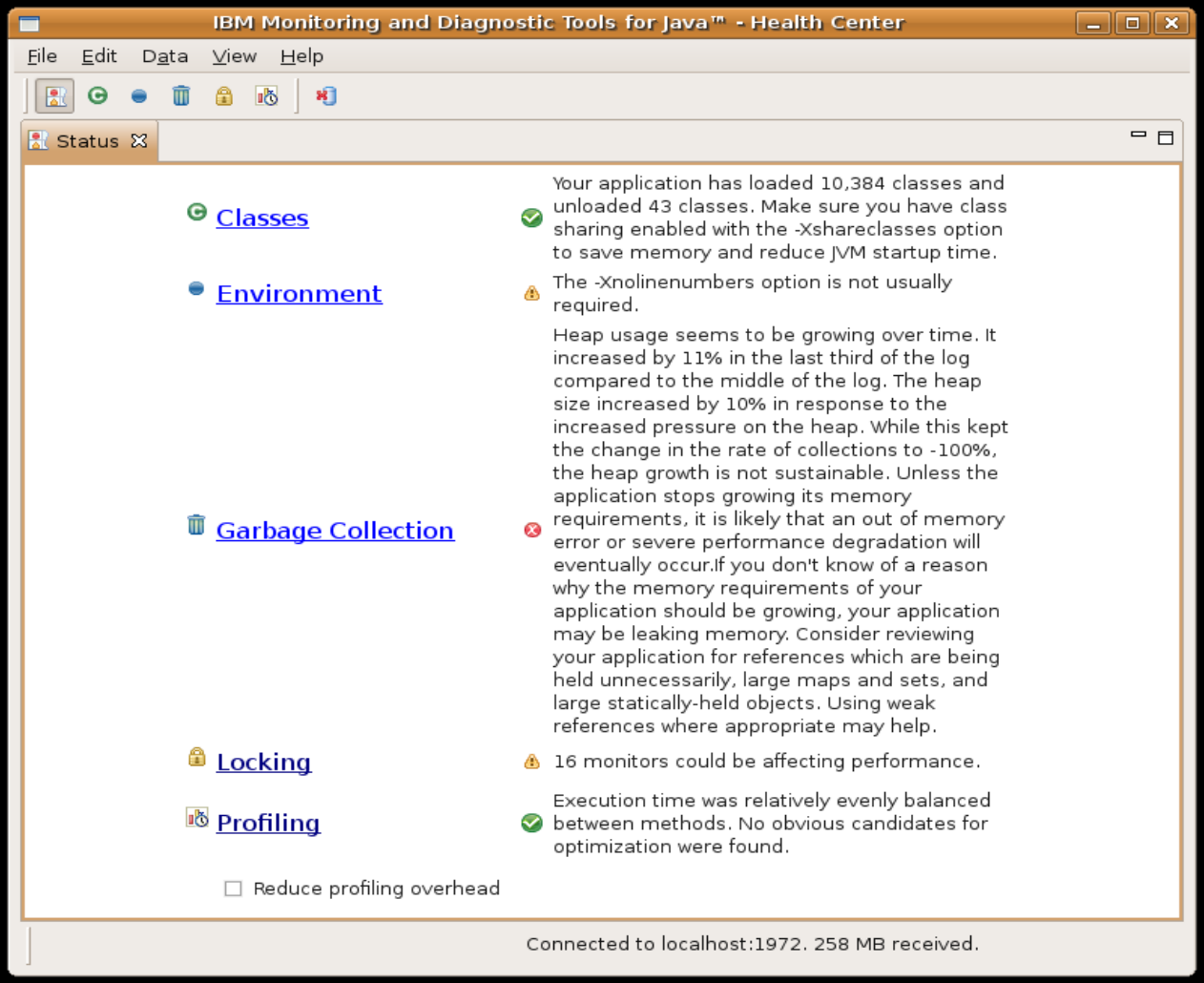
- **Use during the development phase to detect and resolve:**
  - Performance problems
  - Functional issues
- **Use in production to detect and resolve:**
  - Configuration problems
  - Stability issues

How does it work ?

- **Small agent runs on the target JVM**
  - Minimal overhead (circa 3%)
  - Supports all IBM Java platforms
  - Requires Java 5 onwards
- **Data is fed to client which does the analysis**
  - Windows or Linux based client
  - Analyzes/monitors various parts of Java subsystem
  - Makes recommendations based on data received
  - Dynamically updated as more data is received

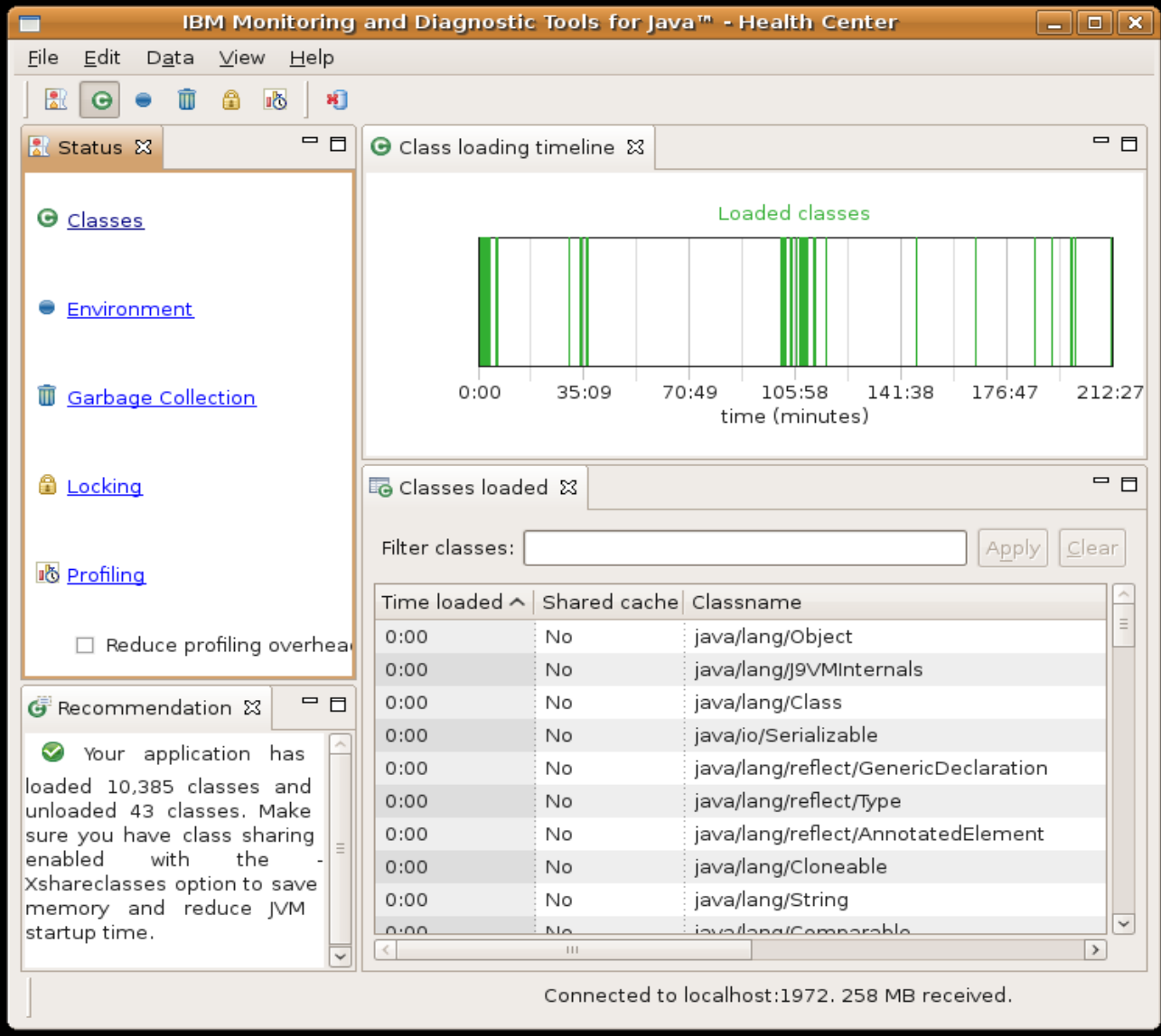
Initial screen – overview and recommendations

- Recommendation engine provides
  - Visual indicator of status
  - Explanation of potential problems and solutions



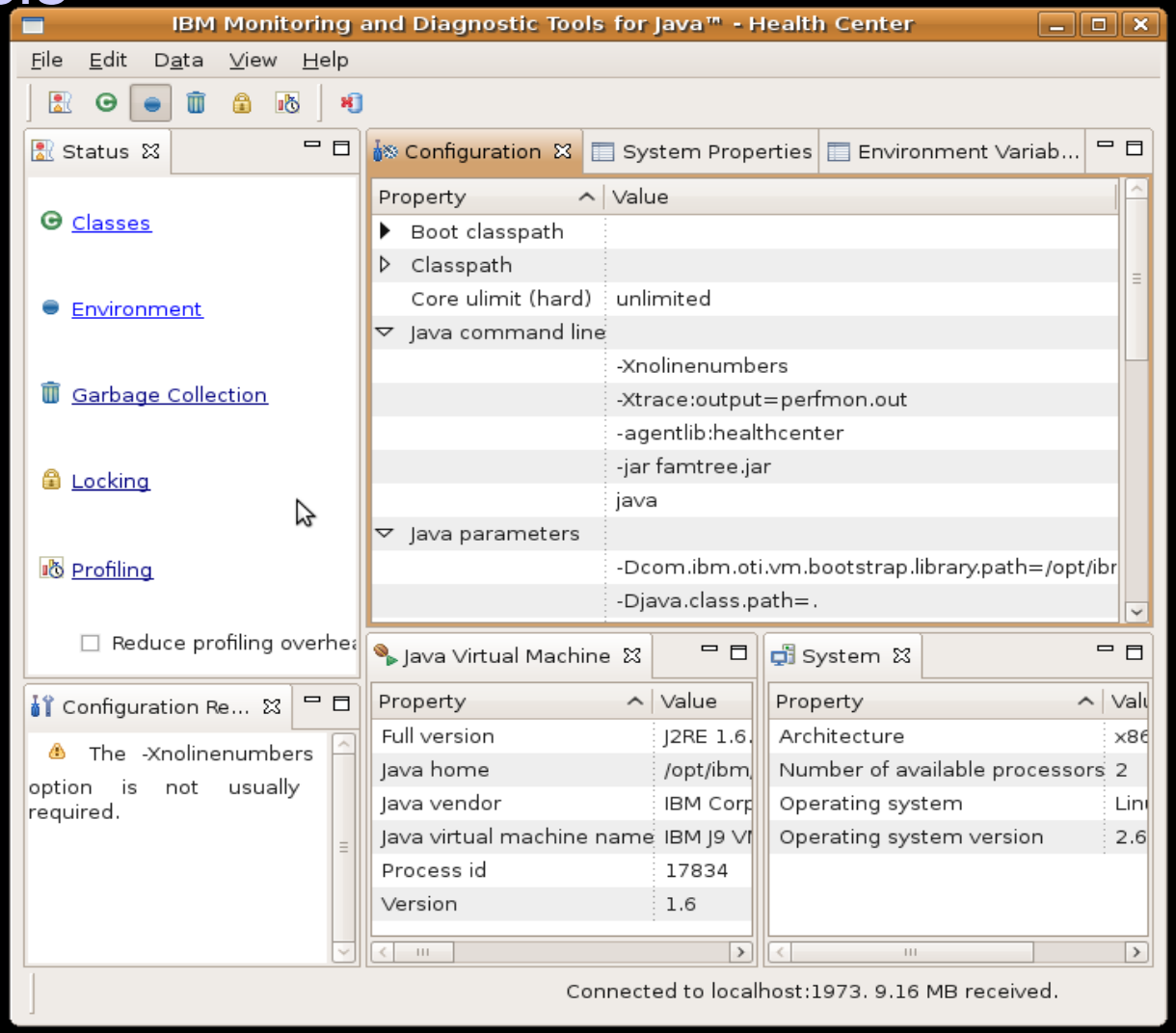
Classloading

- Shows all loaded classes
- Shows load time
- Visualizes classloading activity
- Identifies shared classes
- Makes recommendations



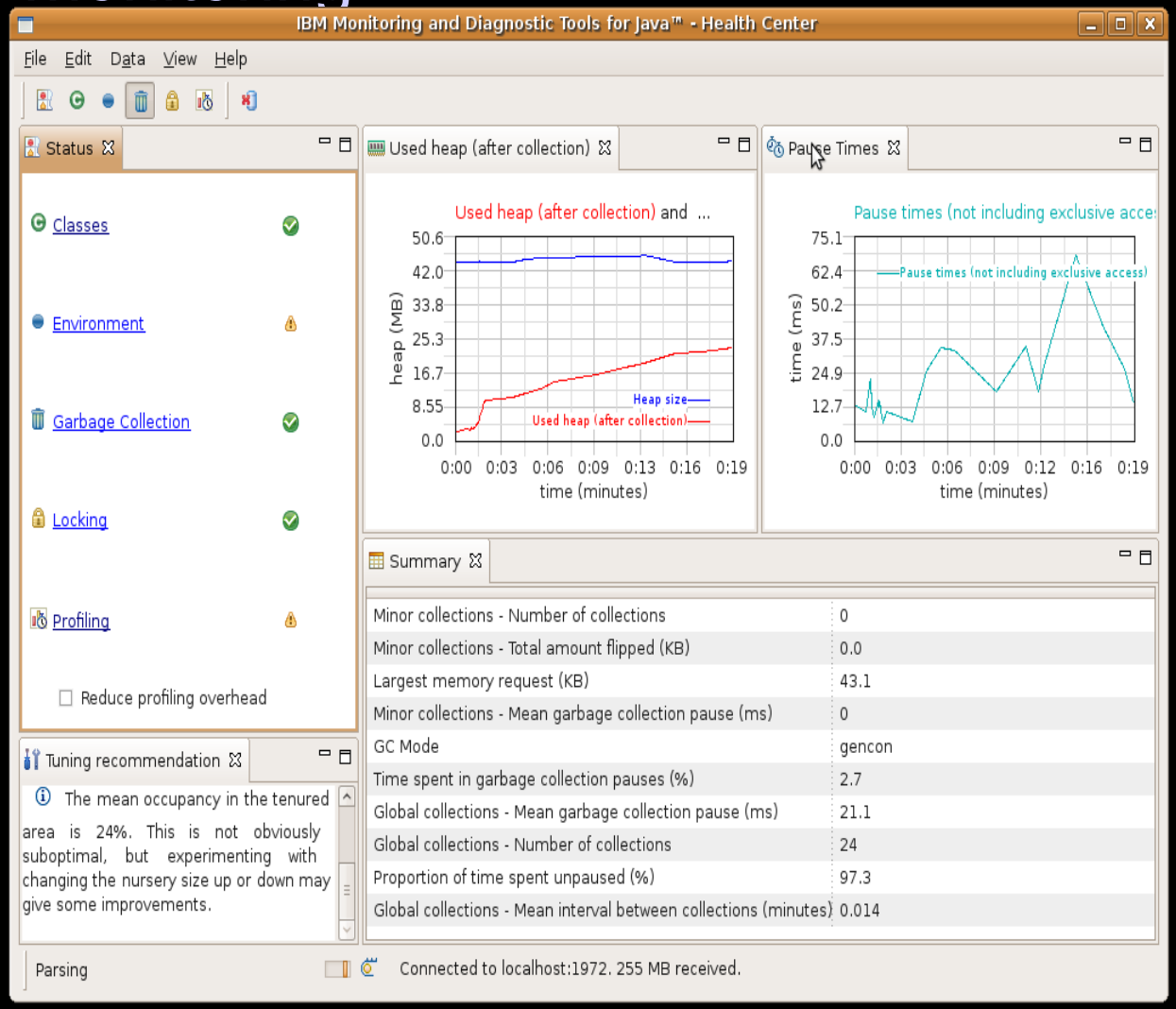
Environment analysis

- Detects unsupported Java options
- Detects options which may hurt performance or serviceability
- Useful for remote diagnosis of configuration-related problems



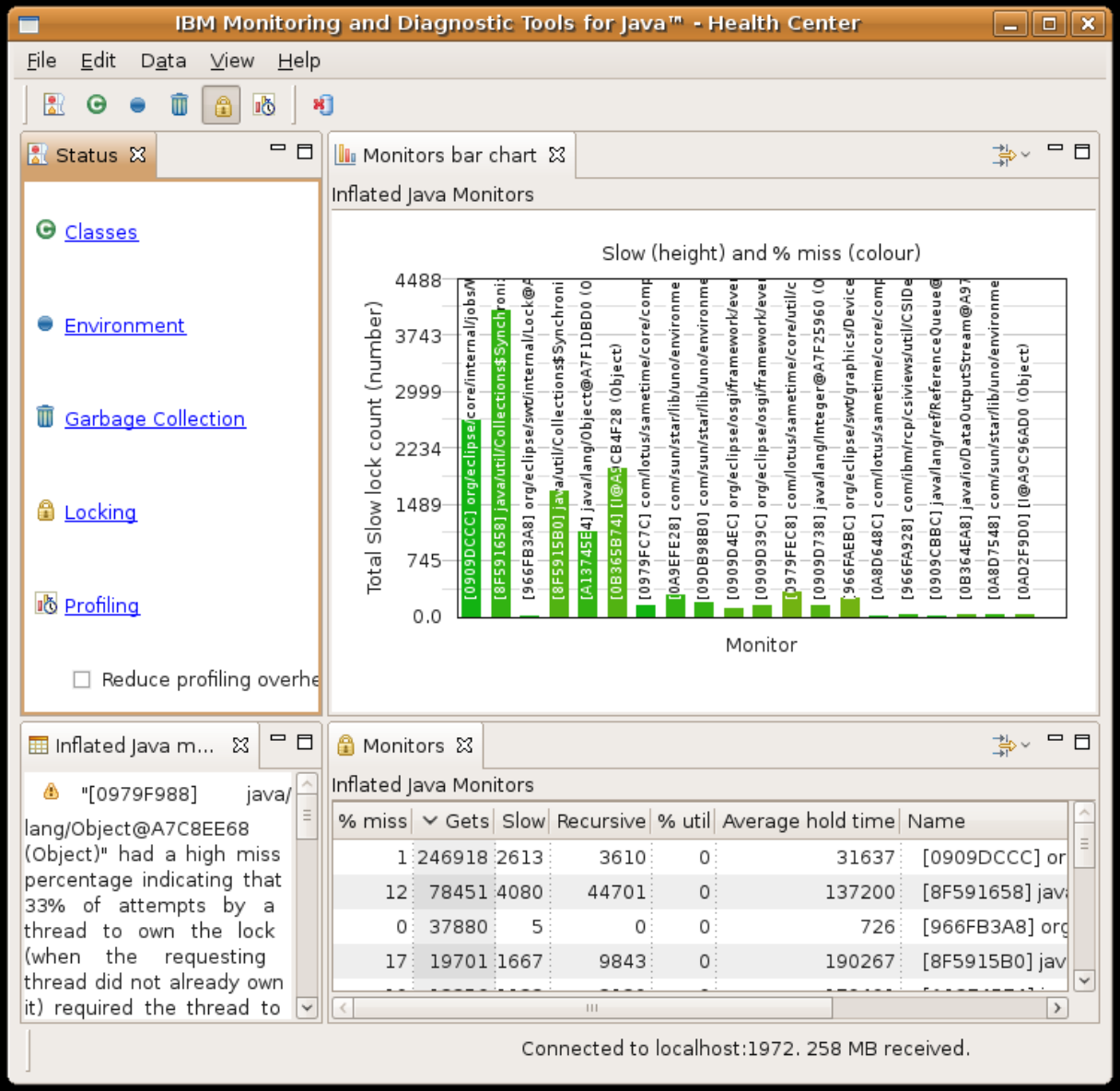
Garbage collection monitoring

- Visualizes heap usage and gc pause times over time
- Identifies memory leaks
- Suggests command-line and tuning parameters
- Same recommendation logic as GCMV



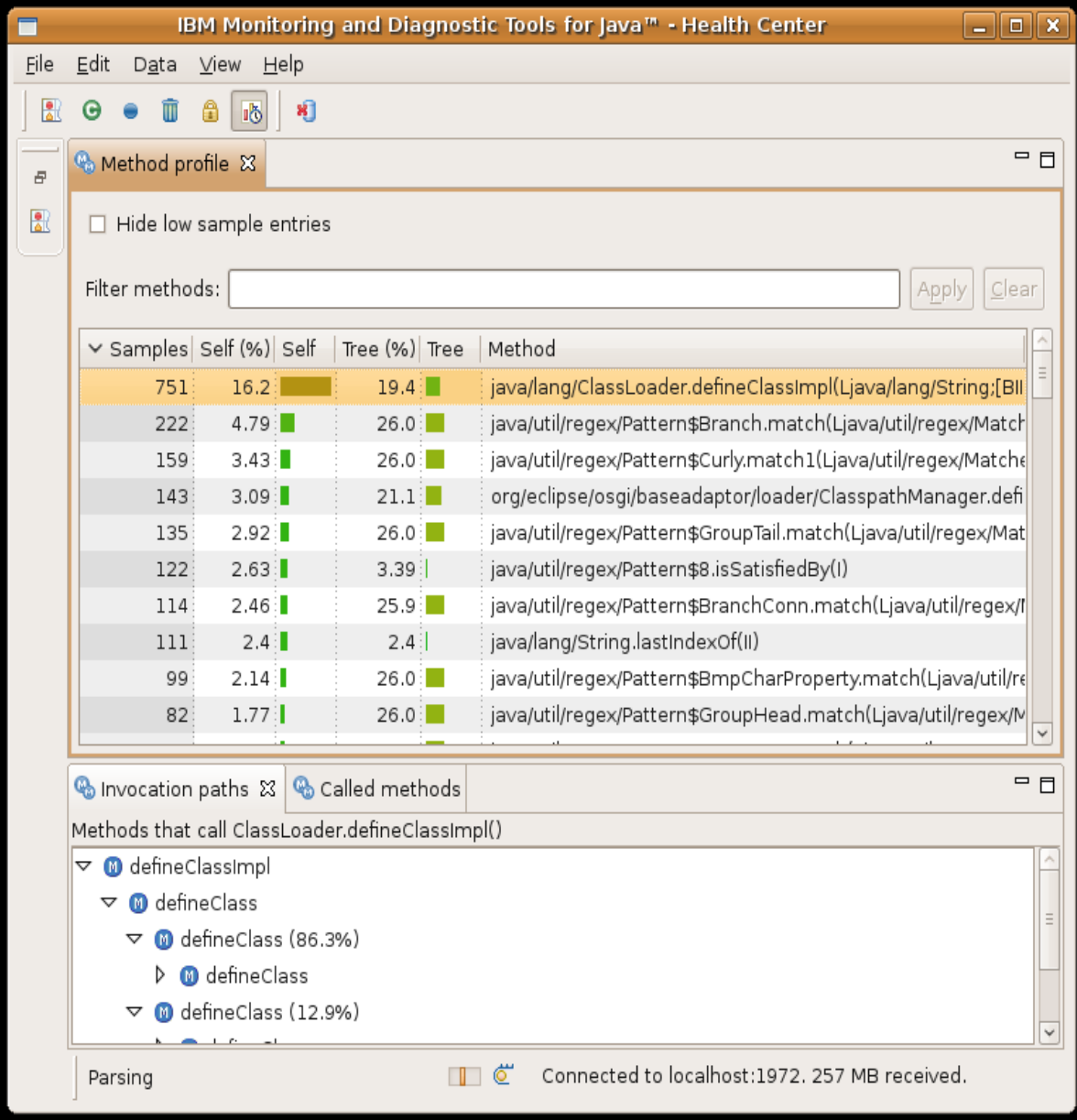
Lock analysis

- Always-on lock monitoring
- Quickly allows the usage of all locks to be profiled
- Helps to identify points of contention in the application that are preventing scaling



Method profiling

- Always-on profiling offers insight into application activity
- Identifies the hottest methods in an application
- Full call stacks to identify where methods are being called from and what methods they call
- No bytecode instrumentation, no recompiling



Availability

- Requires Java 5 or Java 6
- GA-level (suitable for production systems) from Java 5 SR10 or Java 6 SR5
- Agent ships with VM from Java 5 SR9 or Java 6 SR3
- Beta-level function available from Java 5 SR8 and Java 6 SR1

Useful links

- **Direct link**
  - http://www.ibm.com/developerworks/java/jdk/tools/index.html
- **Or ...**
  - Use your favourite search tool to look for ...
    - IBM Java
  - Take first link which will be to IBM DeveloperWorks
  - Select
    - “Monitoring and diagnostic tools”

Health Center demo