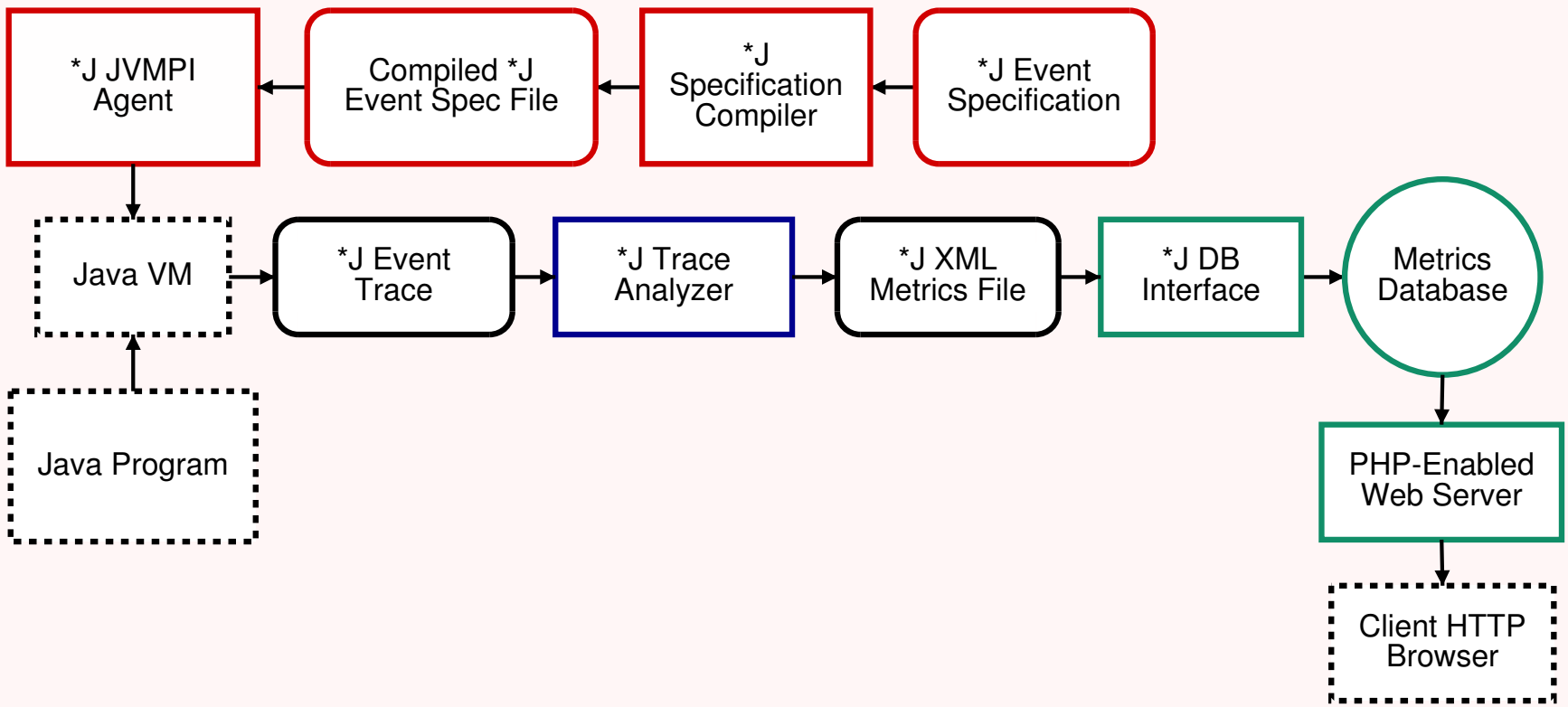
# An Introduction to the *J Dynamic Analysis Framework

Bruno Dufour (dufour@cs.rutgers.edu)

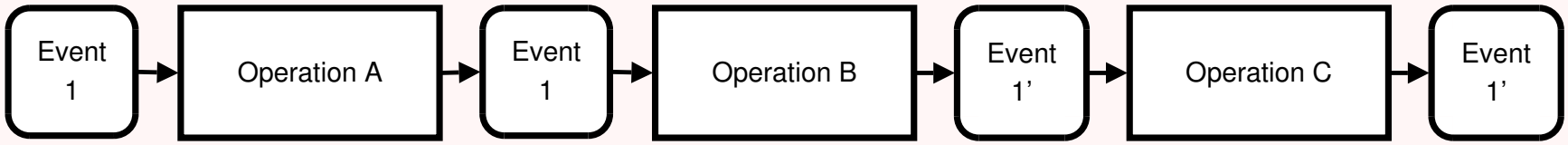
PROLANGS research group, Rutgers University DCS

# *J Overview


*J JVMPI Agent
Compiled *J Event Spec File
*J Specification Compiler
*J Event Specification
Java VM
*J Event Trace
*J Trace Analyzer
*J XML Metrics File
*J DB Interface
Metrics Database
Java Program
PHP-Enabled Web Server
Client HTTP Browser

# *J Computation Model

- Ordered event pipe through which runtime events are propagated:




- Operations can mutate, replace or delete events
- *J provides a number of built-in operations, including:
  - Services (common operations)
    - ID resolver
    - Call Stack Manager
    - Instruction resolver
    - . . .
  - Printers
    - Dynamic call graph printer
    - Class printer
    - . . .

# Compiling the Profiling Agent

- Extract source:

```
% tar xvzf /links/downloads/starj-0.1-beta8.tar.gz
% cd starj-0.1-beta8
```

- Set `JAVA_HOME` variable to point to your Java SDK

```
% export JAVA_HOME=/opt/sun-jdk-x86-1.4.2.11
```

  Note: `$JAVA_HOME/include/jvmpi.h` must exist.

- Compile the profiling agent

```
% cd agent/
% make
% cd ..
```

# Compiling the Profiling Agent (2)

- Compilation warnings, if any, can be ignored
- *May* work on MS Windows after the following line has been commented out from `src/agent/starj_config.h`

```
#define STARJ_ENABLE_PIPE
```

- Other configuration options that can be turned off if needed:

```
#define STARJ_USE_INLINE
#define STARJ_USE_COLOURS
```

# Setting Environment Variables

- Profiling agent needs all system jars explicitly in the `CLASSPATH`:

```
% export CLASSPATH=$PWD:\
$PWD/analyzer/lib/starj.jar:\
$JAVA_HOME/jre/lib/rt.jar
```

  Note: you can use `starj.ClassLocator` to look for classes in system jars, e.g.:

```
% java starj.ClassLocator java.lang.String
[...]
*J> Class 'java.lang.String' found in
'/opt/sun-jdk-x86-1.5.0.04/jre/lib/rt.jar'.
```

- Set `LD_LIBRARY_PATH` so that the JVM will find the *J agent library:

```
% export LD_LIBRARY_PATH=$PWD/agent/lib
```

# The Event Specification

```
default {
    recorded: yes;
    env_id: yes;
}
event Class* {
    class_id: yes;
}
event ClassLoad {
    class_name: yes;
    source_name: yes;
    methods: yes;
}
event JVM* {}
```

```
event MethodEntry2 {
    method_id: yes;
    obj_id: yes;
}
event MethodExit {
    method_id: yes;
}
event ThreadStart {
    thread_env_id: yes;
    thread_name: yes;
    group_name: yes;
    parent_name: yes;
}
event ThreadEnd {}
```

# Compiling the Event Specification

- Invoke the *J event specification compiler:

```
% cd spec
% cp ~/courses/516/p2/dyncg.sj .
% java starj.spec.Compiler dyncg.sj
*J> Parsing file: 'dyncg.sj'
*J> Parsing successful
*J> Writing output to: 'dyncg.spec'
*J> Compilation successful
% cd ..
```

- To see the fully expanded specification with resolved dependencies:

```
% java starj.spec.Extractor dyncg.spec
```

# Tracing a Java Application

```
% java -Xint -Xrunsj:specfile=spec/dyncg.spec,\
cp=$CLASSPATH,gzip=true,file=allyourbase.trace \
AllYourBase
*J Warning> Optimize mode will be turned off due to
requested fields
*J Agent> Initialization completed
All your base are belong to us!
*J Agent> Completed program execution

% file allyourbase.trace
allyourbase.trace: gzip compressed data, from Unix
```

# Invoking the Trace Analyzer

```
% java starj.Main -p toolkits.printers.class \
enabled,file:allyourbase.loaded_classes allyourbase.trace
*J Warning> Unknown trace attribute: 'starj.bytecode.tags'
*J> Collecting event dependencies
*J> Disabling: InstructionResolver (Dependency on event 36 cannot be satisfied)
*J> Disabling: CallSiteResolver (Dependency on event 36 cannot be satisfied)
*J> Collecting operation dependencies
*J> Disabling: PropagationManager (missing dependency: CallSiteResolver)
*J> Disabling unnecessary service: CallStackManager
*J> Disabling unnecessary service: IDResolver
*J> Pruning operation lists
*J> Flattening hierarchy
*J> Initializing operations
*J> Processing trace ...
*J> 14754 events processed in 00h00m00.167s
*J> Finalizing
*J> Processing completed successfully
```

# Extending *J – Custom Printer

```
public class ThreadCGPrinter extends AbstractPrinter {
    public ThreadCGPrinter(String name, String description) {
        super(name, description);
    }

    public ThreadCGPrinter(String name, String description, PrintStream out) {
        super(name, description, out);
    }

    public void init() {
        super.init();
        // You initialization code here
    }

    [...]
```

# Extending *J – Custom Printer (2)

```
public OperationSet operationDependencies() {
    OperationSet ops = super.operationDependencies();
    ops.add(IDResolver.v());
    return ops;
}

public EventDependencySet eventDependencies() {
    EventDependencySet dep_set = super.eventDependencies();
    FieldMask method_mask = new TotalMask(Constants.FIELD_RECORDED
            | Constants.FIELD_METHOD_ID | Constants.FIELD_ENV_ID);
    dep_set.add(new EventDependency(Event.METHOD_ENTRY2, method_mask, true,
           new EventDependency(Event.METHOD_ENTRY, method_mask, true)));
    dep_set.add(new EventDependency(Event.METHOD_EXIT, method_mask, true));
    return dep_set;
}

[...]
```

# Extending *J – Custom Printer (3)

```
    public void apply(EventBox box) {
        Event event = box.getEvent();

        // Process event
    }

    public void done() {
        PrintStream out = this.out;

        // Output the call graph(s)
    }
}
```

# Extending *J – Driver Class

```
public class Main {
    public static void main(String[] args) {
        // Get a reference to the printer pack
        RootPack root = Scene.v().getRootPack();
        Pack prn_pack = (Pack) root.getByName("toolkits.printers");

        // Add our own printer to it
        prn_pack.add(new ThreadCGPrinter("threadprn",
                "Per-thread CG printer"));

        // Delegate execution to *J
        starj.Main.main(args);
    }
}
```

# Using the Driver

```
% java Main --element-help toolkits.printers.threadprn
Operation 'threadprn':
----------------------

    Per-thread CG printer

Configuration          Description
-------------          -----------
enabled                Configures whether this element is enabled
                       or not
file                   Sets the name of a file to use as output
```