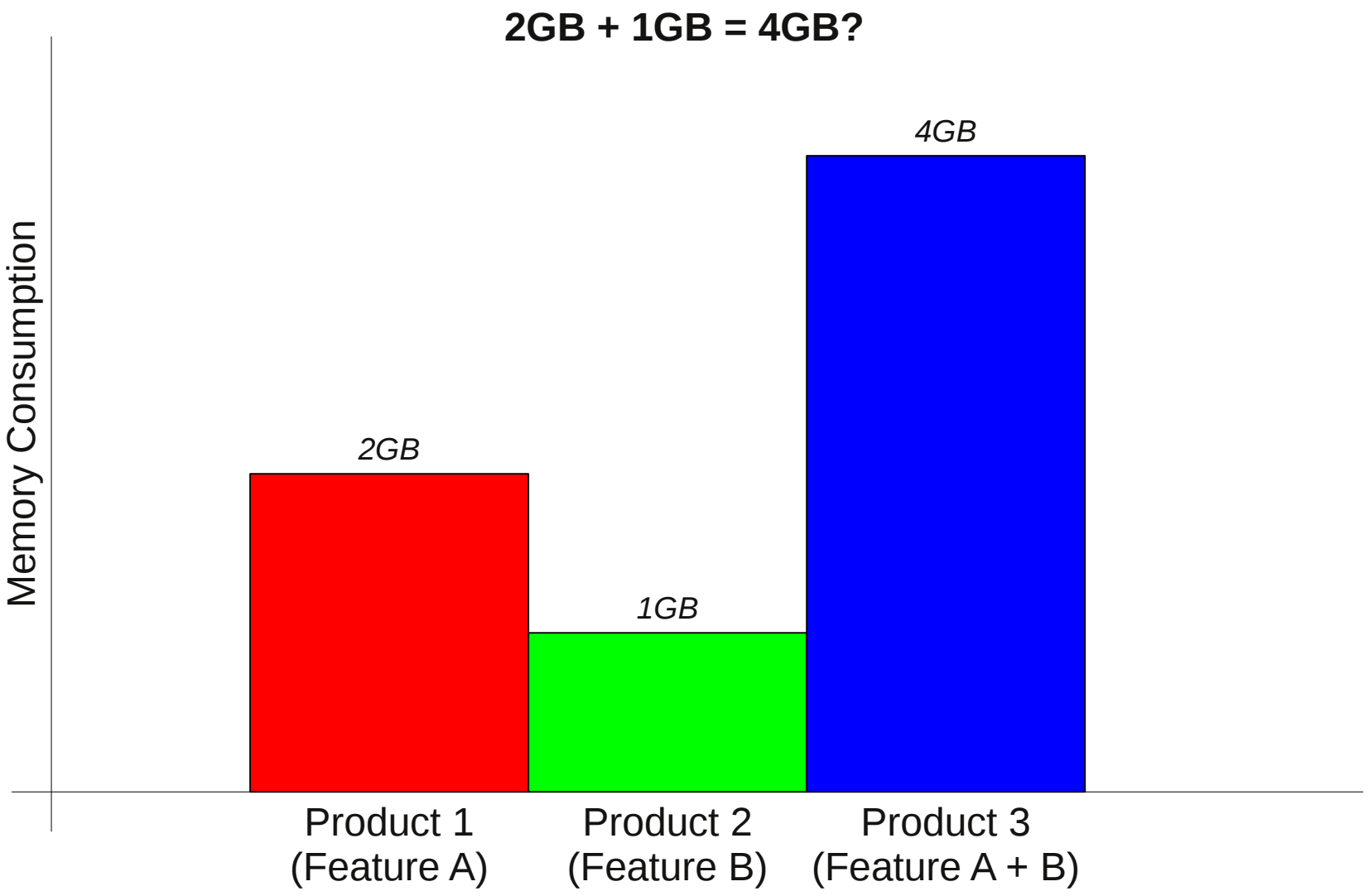
Where has all my Memory gone? Determining Memory Characteristics of Product Variants using VM-Level Monitoring

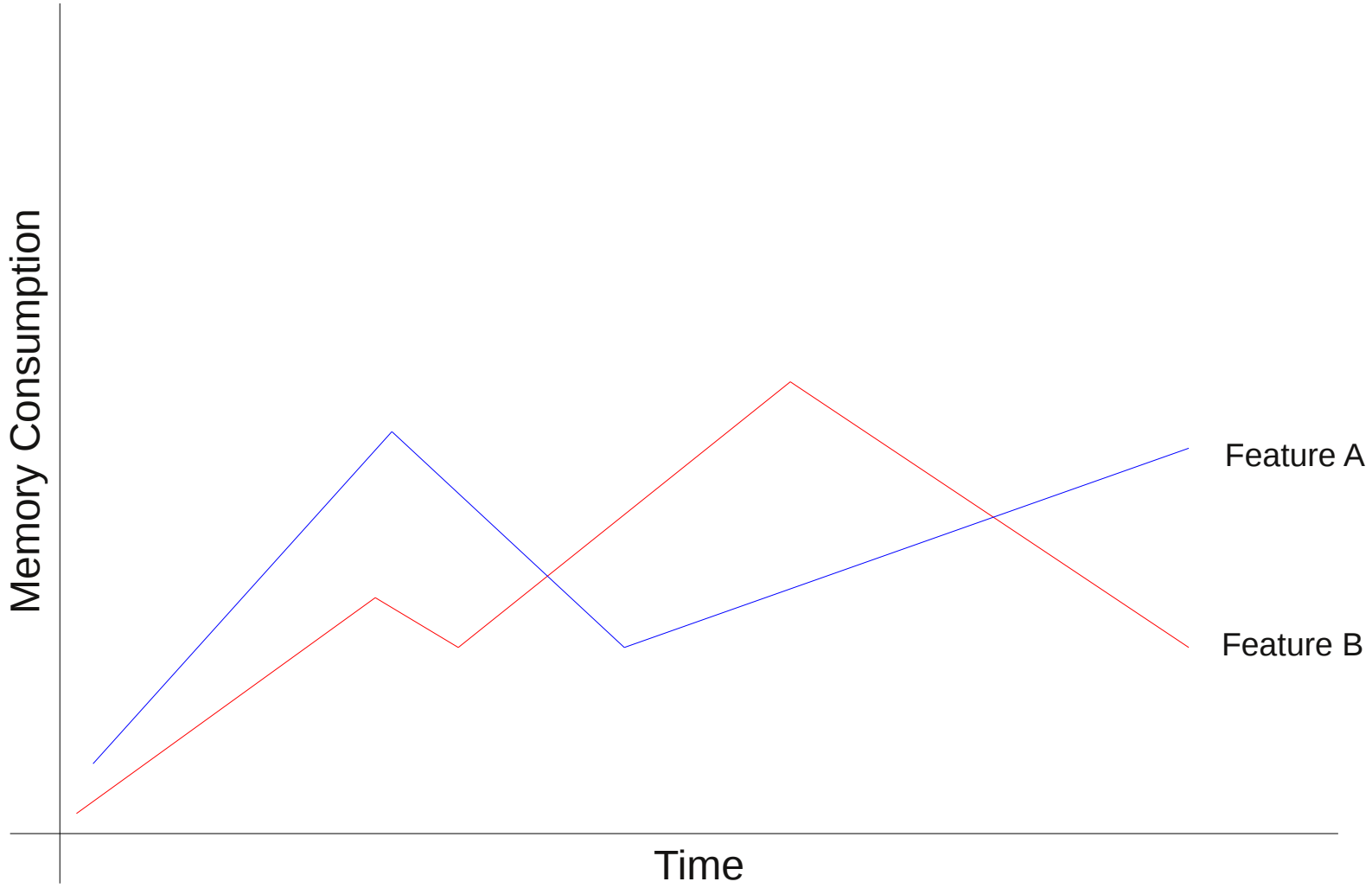
Philipp Lengauer, Verena Bitto, Florian Angerer
Paul Grünbacher, Hanspeter Mössenböck

2014-01-23

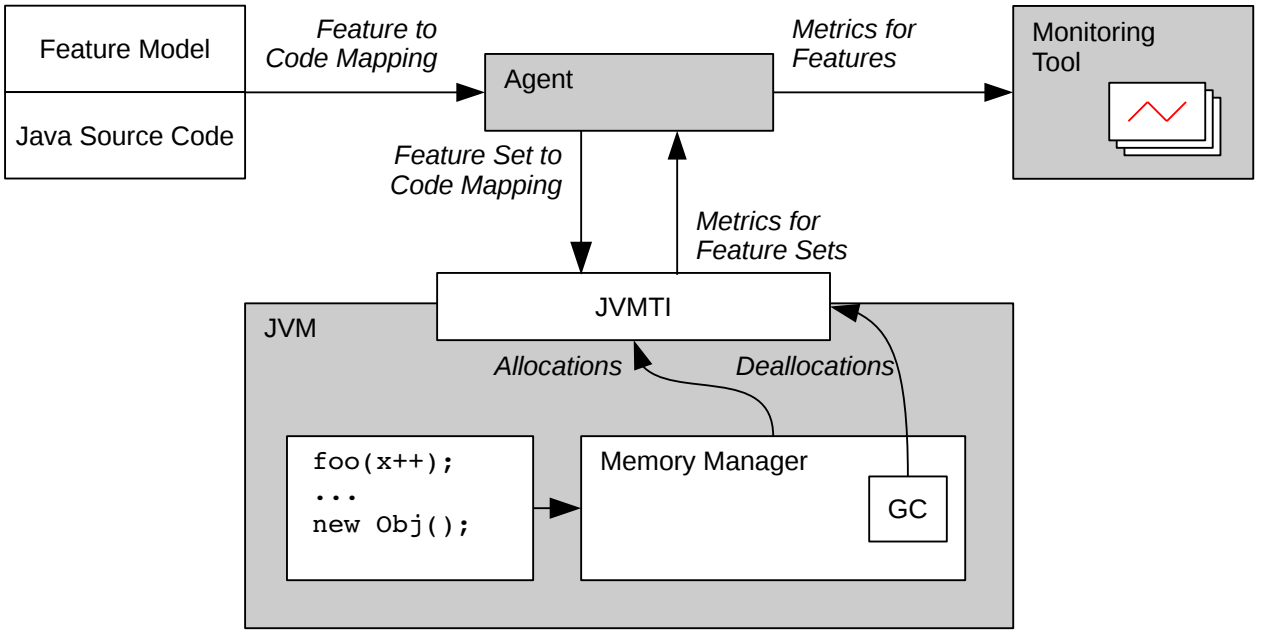
Motivation – Performance Prediction + Interaction



Motivation – Feature-based Online Monitoring + Real-time Anomaly Detection



Approach



Provide feature-to-code mapping:

```
register_get_feature_callback(featureID
(*get_feature)(jclass clazz)
```

Access per-feature memory statistics:

```
get_feature_memory_usage(featureID feature,
jlong* allocations, jlong* deallocations)
```

based on OpenJDK 8

Allocations and Deallocations

Feature Model:
A → Feature 1
B → Feature 2

```
class A {
     ...
     b = new B(0,1);
     ...
}

class B {
     int x, y;
     B(int x, int y) {
          ...
     }
}
```

b->feature = "Feature 1"
allocations[b->feature] += b->size();

b

header
"Feature 1"
0 *(x : int)*
1 *(y : int)*

```
class A {
     ...
     b = null;
     System.gc();
     ...
}
```

b

header
"Feature 1"
0 *(x : int)*
1 *(y : int)*

deallocations[b->feature] += b->size();

Unassigned Code

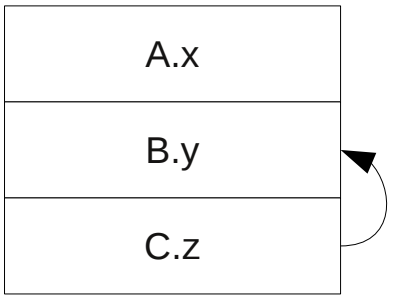
```
class A {
     void x() {
          B.y();
     }
}

class B {
     void y() {
          C.z();
     }
}

class C {
     void z() {
          new A();
     }
}
```

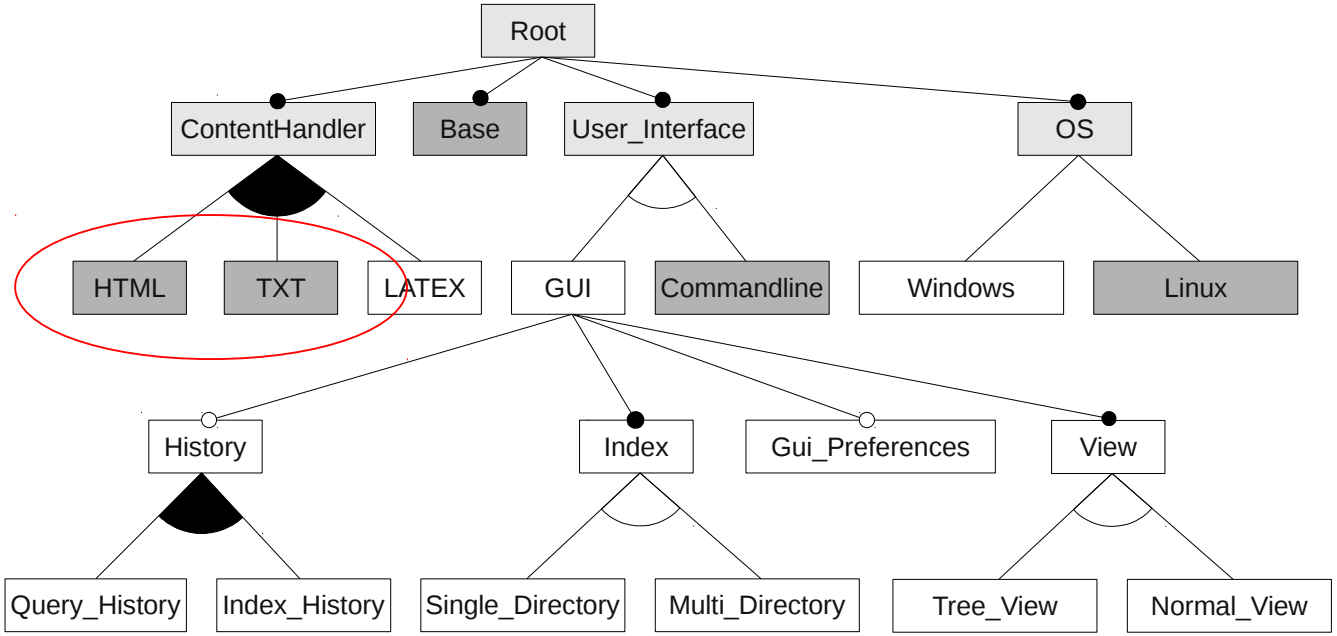
Feature Model:
A → Feature 1
B → Feature 2
C → ?

Stack:



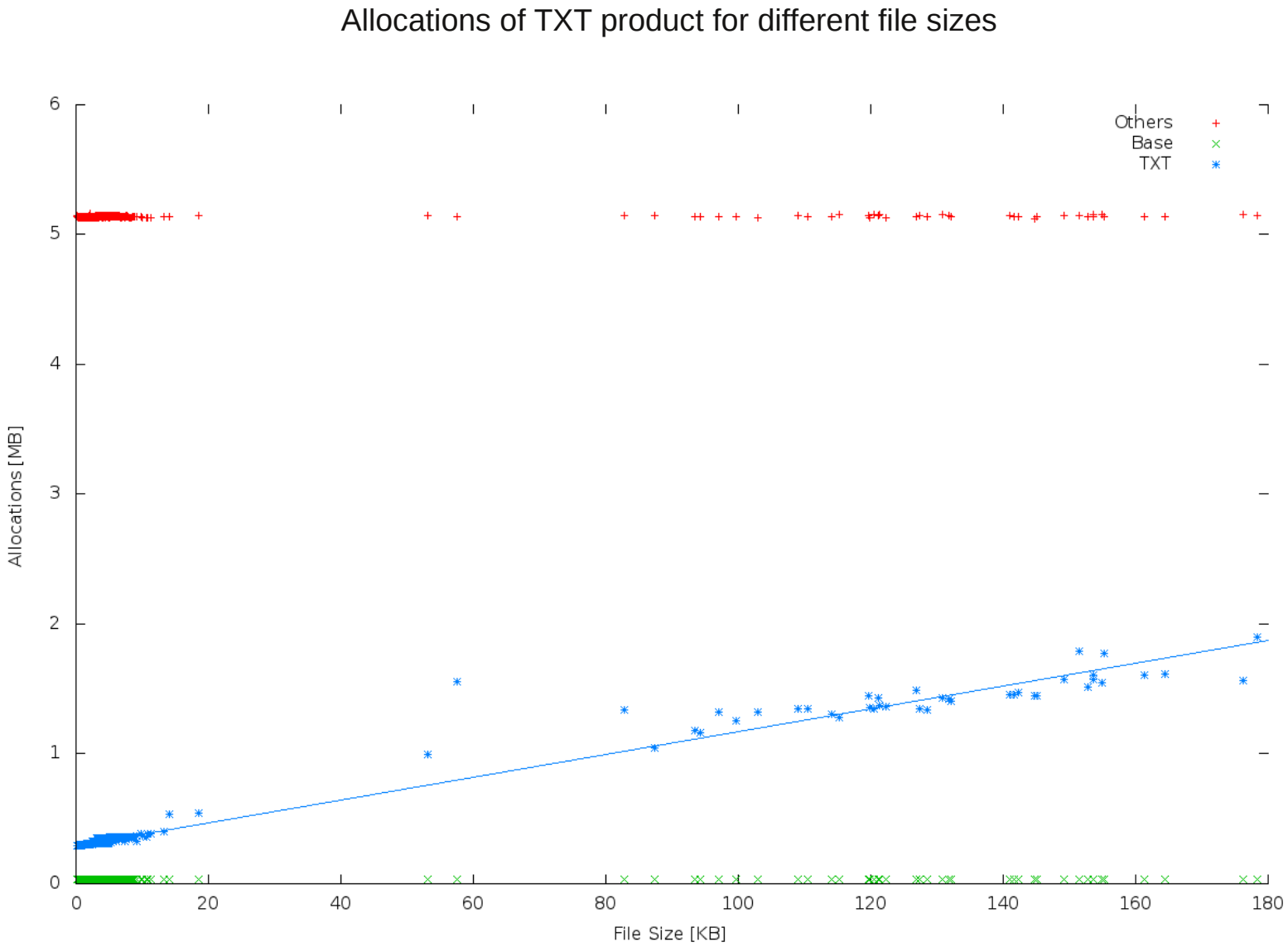
assign “Feature 2”

DesktopSearcher

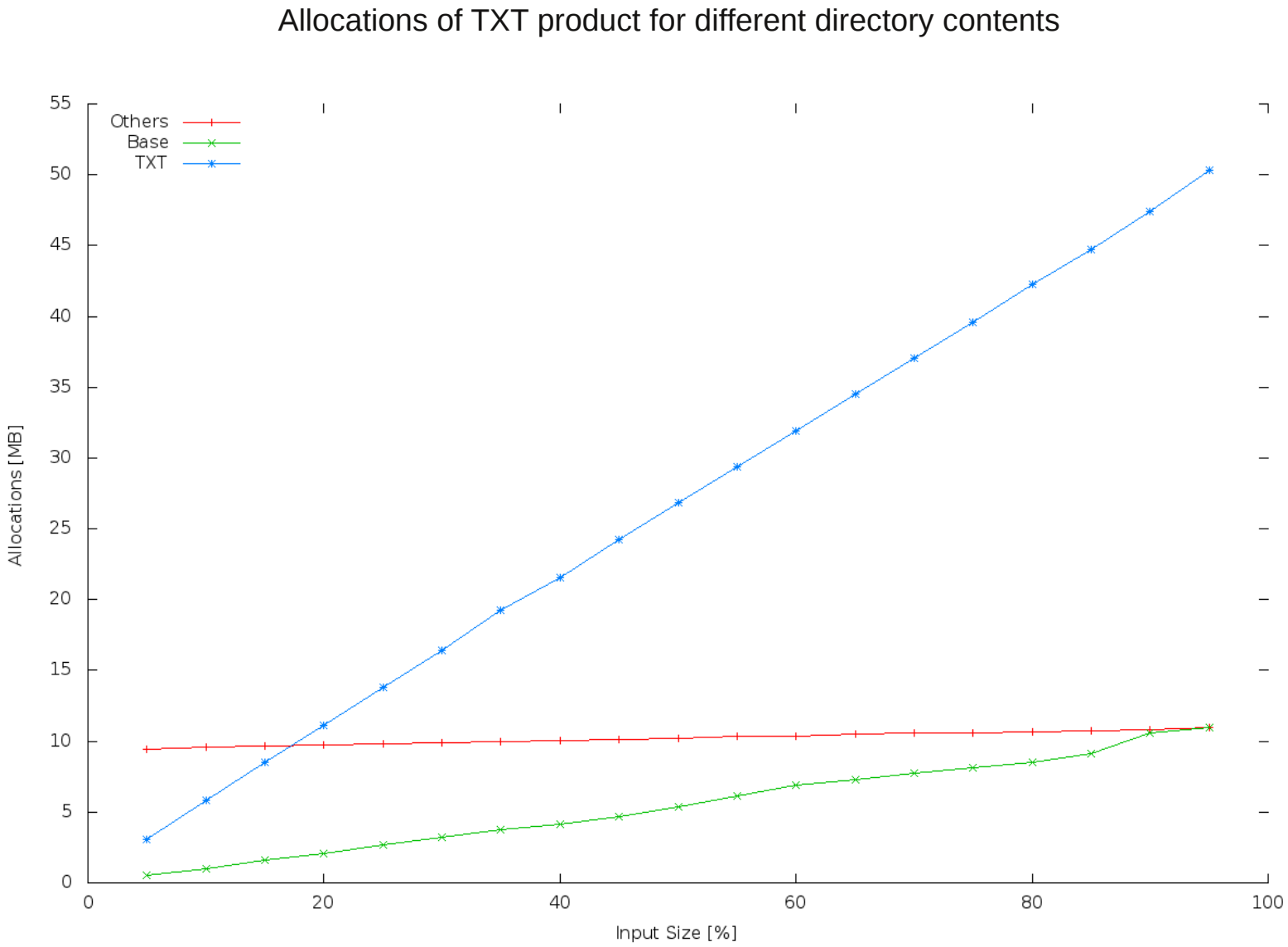


TXT Input: King James Bible,
Works of William Shakespeare
HTML Input: www.ssw.jku.at

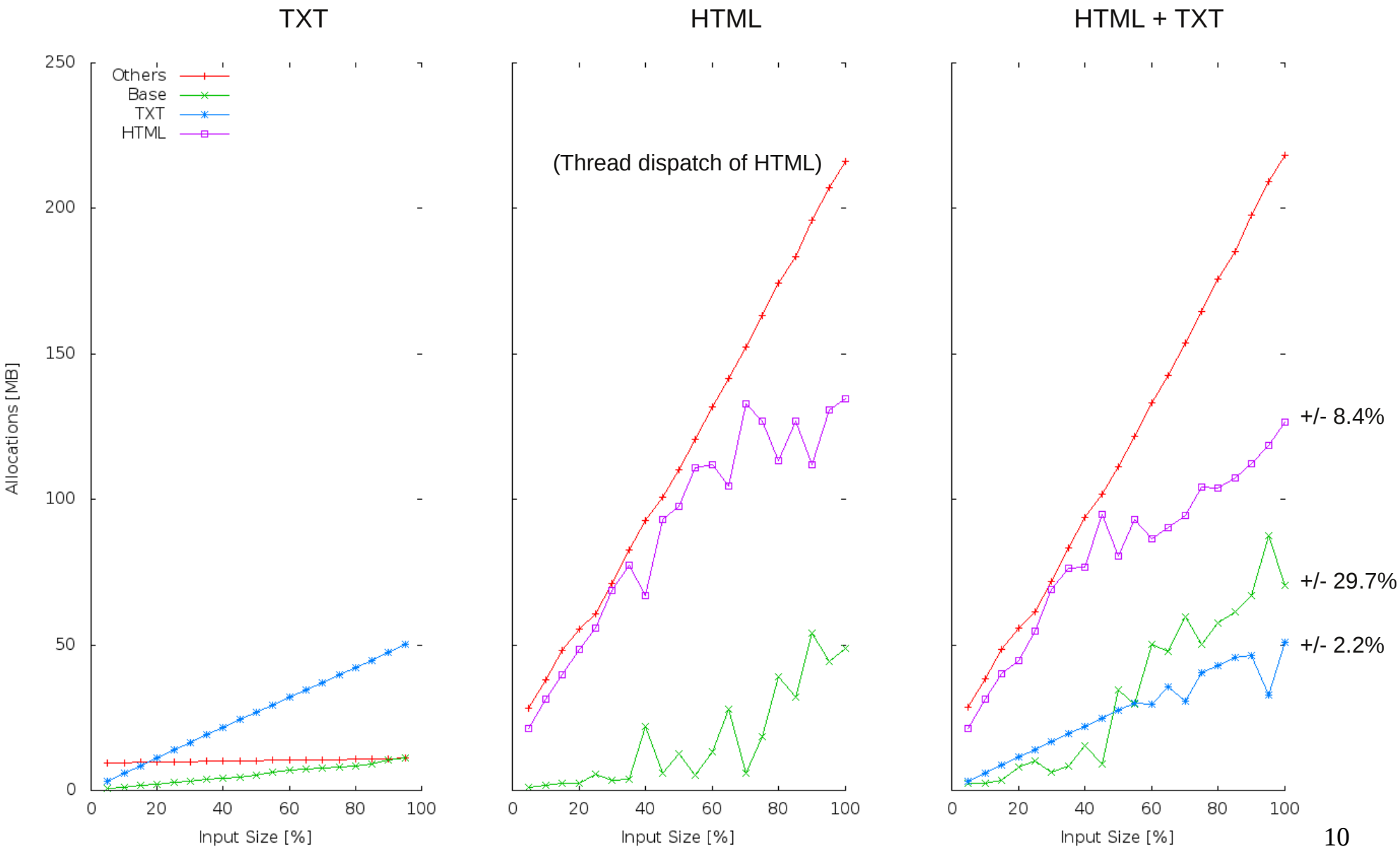
Validation (½) – Single Files



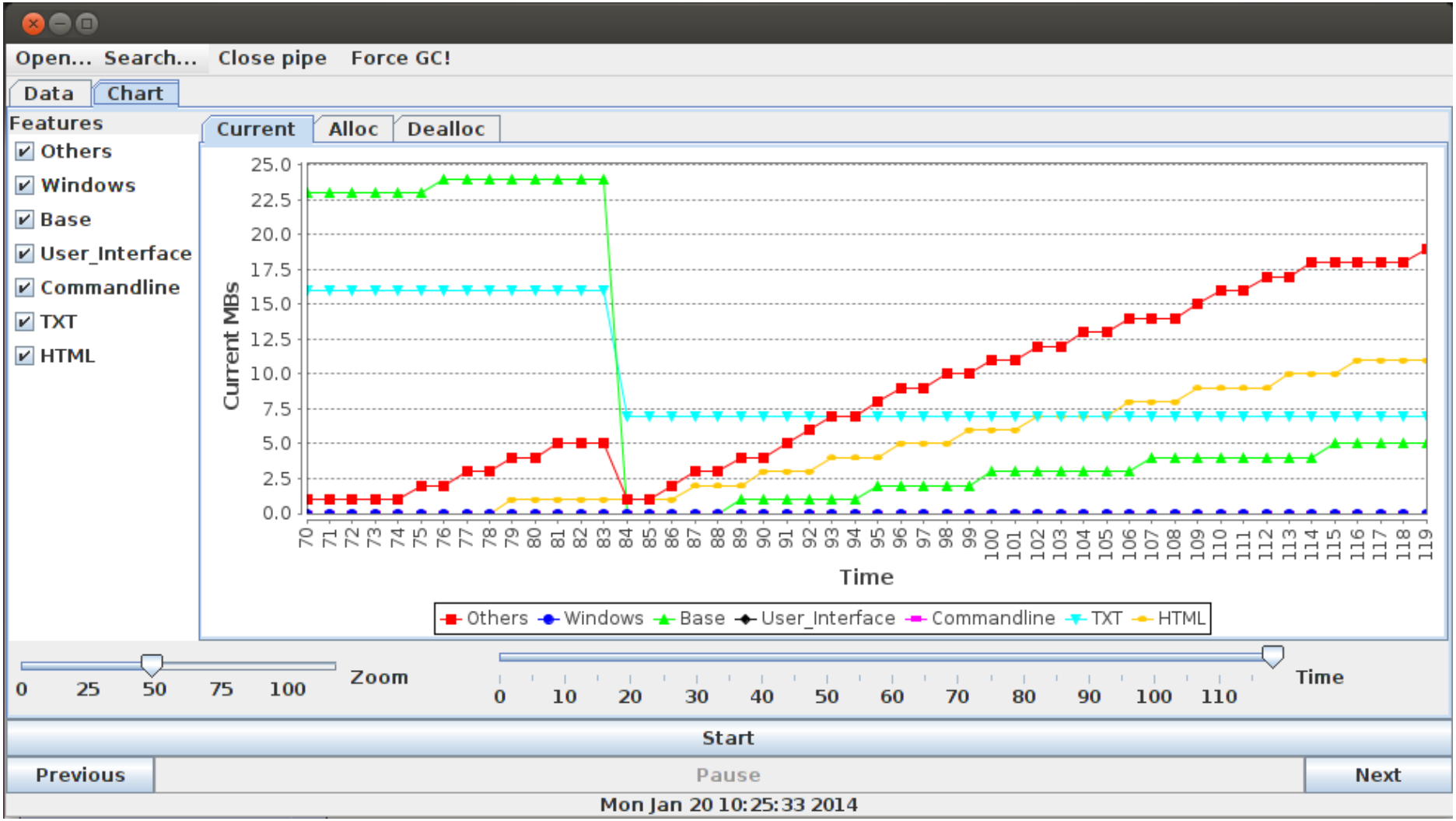
Validation (2/2) - Directories



Performance Prediction



Monitoring Tool



Questions

?