Microsoft Research

Software Analytics Group

Context-Sensitive Delta Inference for Identifying Workload-Dependent Performance Bottlenecks

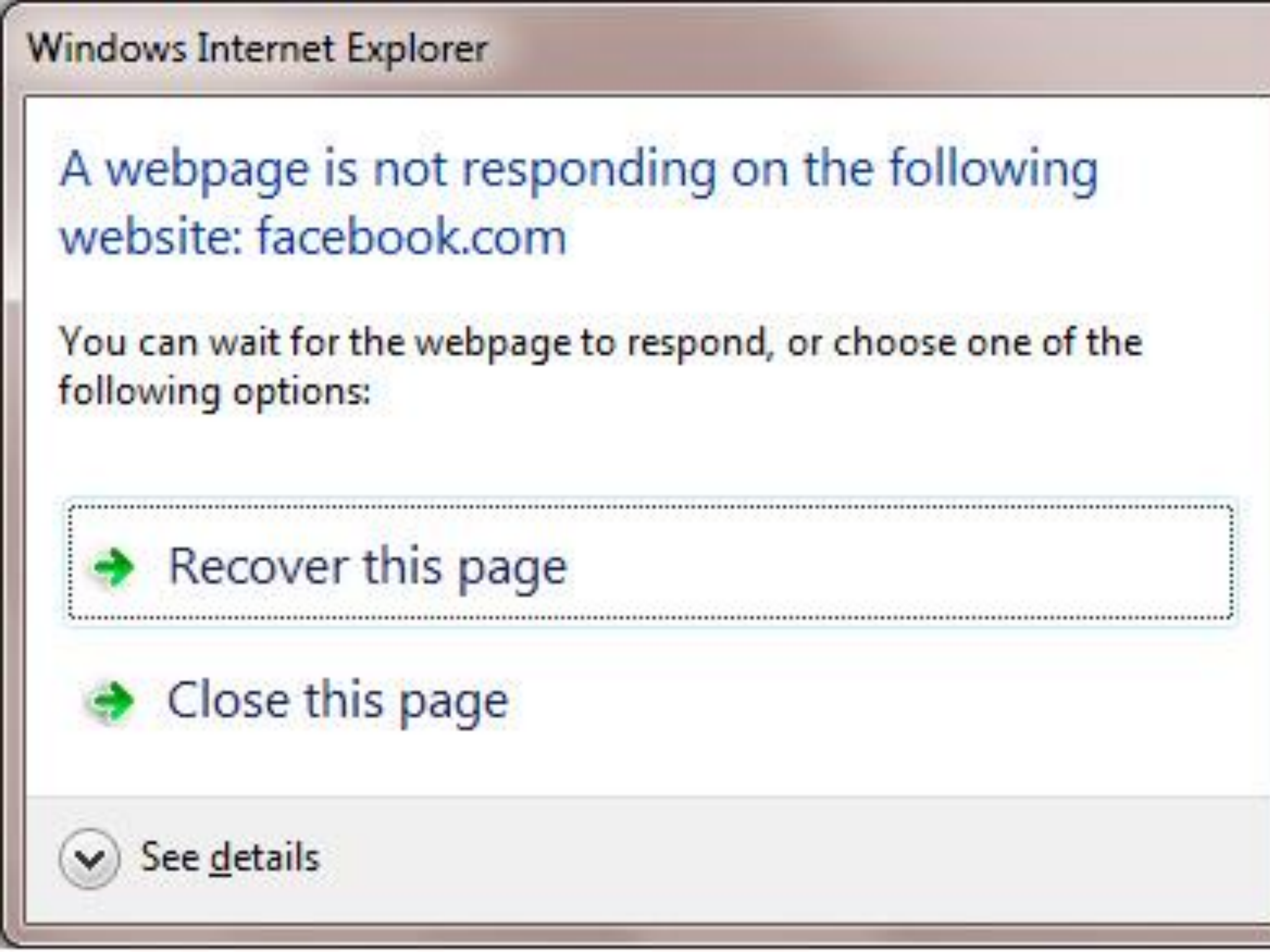
Xusheng Xiao[1], Shi Han[2], Dongmei Zhang[2], **Tao Xie**[1,3]

[1]North Carolina State University

[2]Microsoft Research Asia

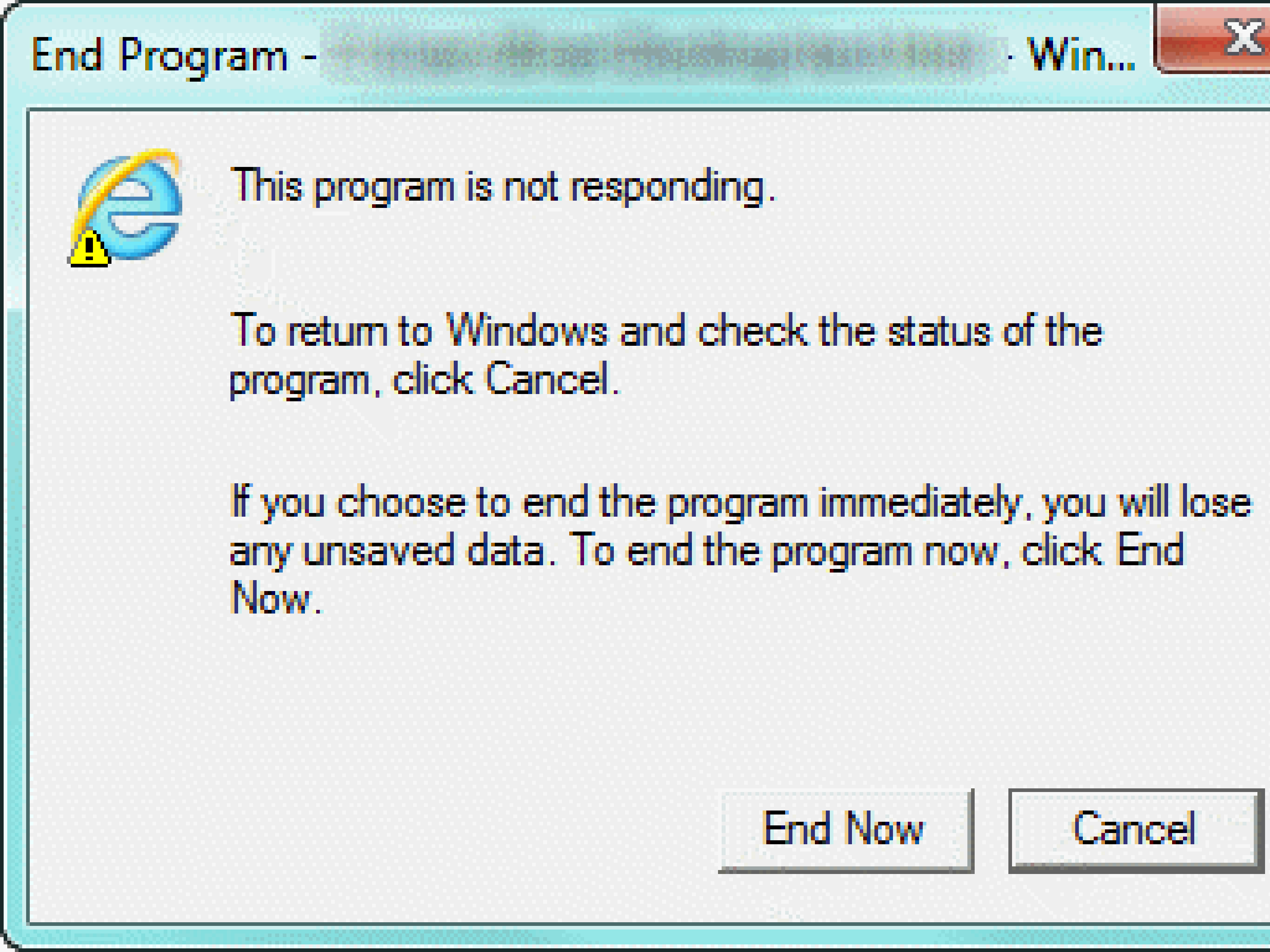
[3]University of Illinois at Urbana-Champaign

[1]xxiao2@ncsu.edu, [2]{shihan, dongmeiz}@microsoft.com, [1,3]taoxie@illinois.edu



A webpage is not responding on the following website: facebook.com

You can wait for the webpage to respond, or choose one of the following options:

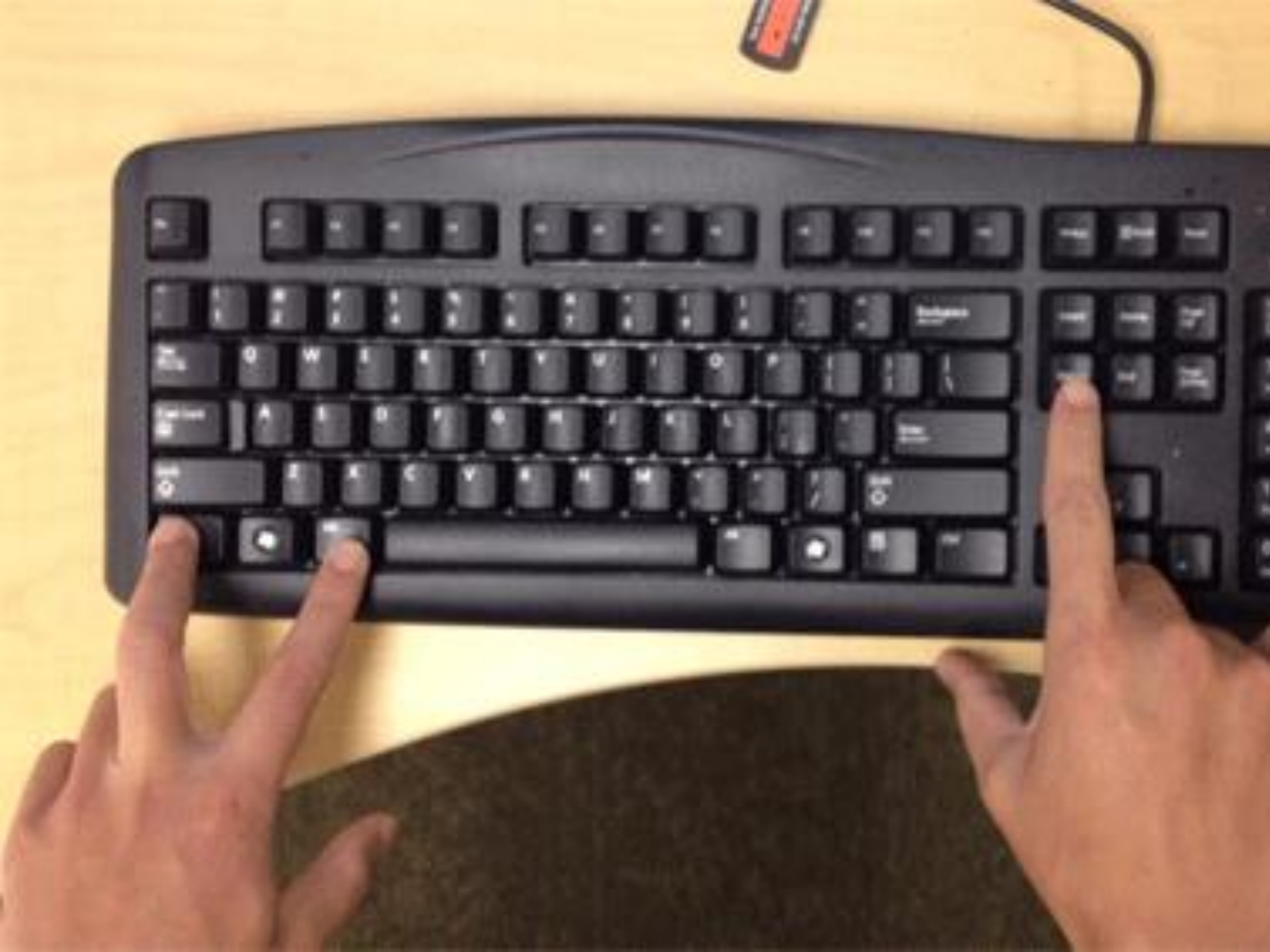


End Program - [redacted] - Win...

This program is not responding.

To return to Windows and check the status of the program, click Cancel.

If you choose to end the program immediately, you will lose any unsaved data. To end the program now, click End Now.



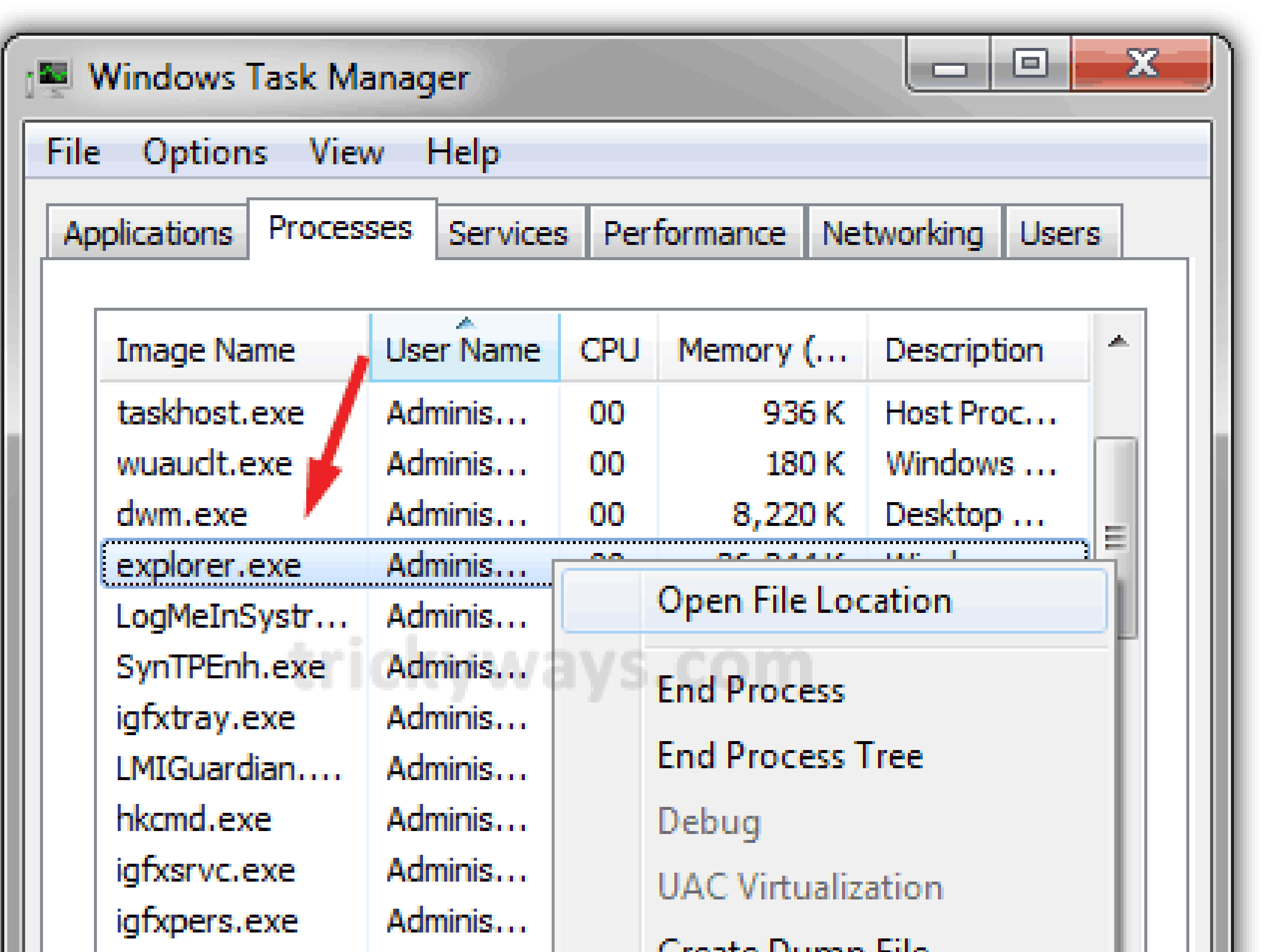
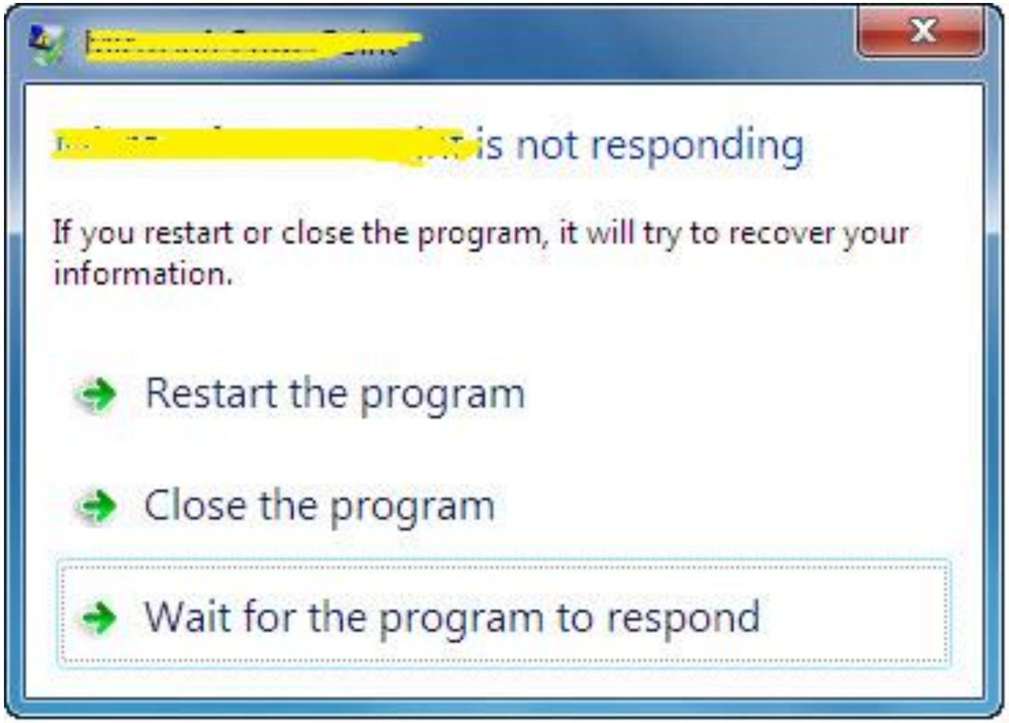


Image Name	User Name	CPU	Memory (...	Description
taskhost.exe	Adminis...	00	936 K	Host Proc...
wuauclt.exe	Adminis...	00	180 K	Windows ...
dwm.exe	Adminis...	00	8,220 K	Desktop ...
explorer.exe	Adminis...	[illegible]	[illegible]	[illegible]
LogMeInSystr...	Adminis...			
SynTPEnh.exe	Adminis...			
igfxtray.exe	Adminis...			
LMIGuardian....	Adminis...			
hkcmd.exe	Adminis...			
igfxsrvc.exe	Adminis...			
igfxpers.exe	Adminis...			

Performance Problems

- Widely exist in released software
  - Mozilla developers fixed **5–60** perf bugs **monthly** over the past 10 years [Jin et al. PLDI 12]



[redacted]

[redacted] is not responding

If you restart or close the program, it will try to recover your information.

Software Hangs in daily tools: *file managers, office tools, browsers,* …

Software Hangs

- Three major categories
  - Correctness, e.g., infinite loops
  - Blocking operations, e.g., sending files
  - Expensive operations

contribute to **27%** of 233 hang bugs

[Song et al. DSN 2010]

*With **root** of disk (C:\ ) **selected** in drop-down path selector, attempting to **enable Flat View** under the top-level View menu causes 7-Zip to hang …*

The process had to be forcibly stopped using Windows Task Manager.

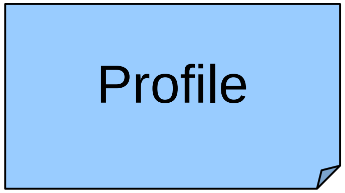
7-Zip File Manager

Workload-Dependent Performance Bottlenecks (WDPB)

- Expensive operations depending on input workloads, e.g., data processing
- **Insight**: caused often by **workload-dependent loops** with **expensive** operations, e.g.,
  - Temp-object creation/destruction
  - File I/O
  - UI updates
- Fixed by
  - Spawning new threads
  - Limiting workload/processing sizes

Target Problem:
WDPB-Loop Prediction

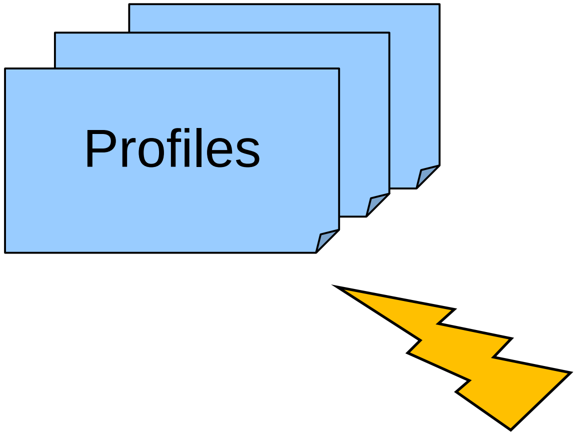
Traditional
Single-Profile Setting



Test Generation:
Hard to know
triggering workload

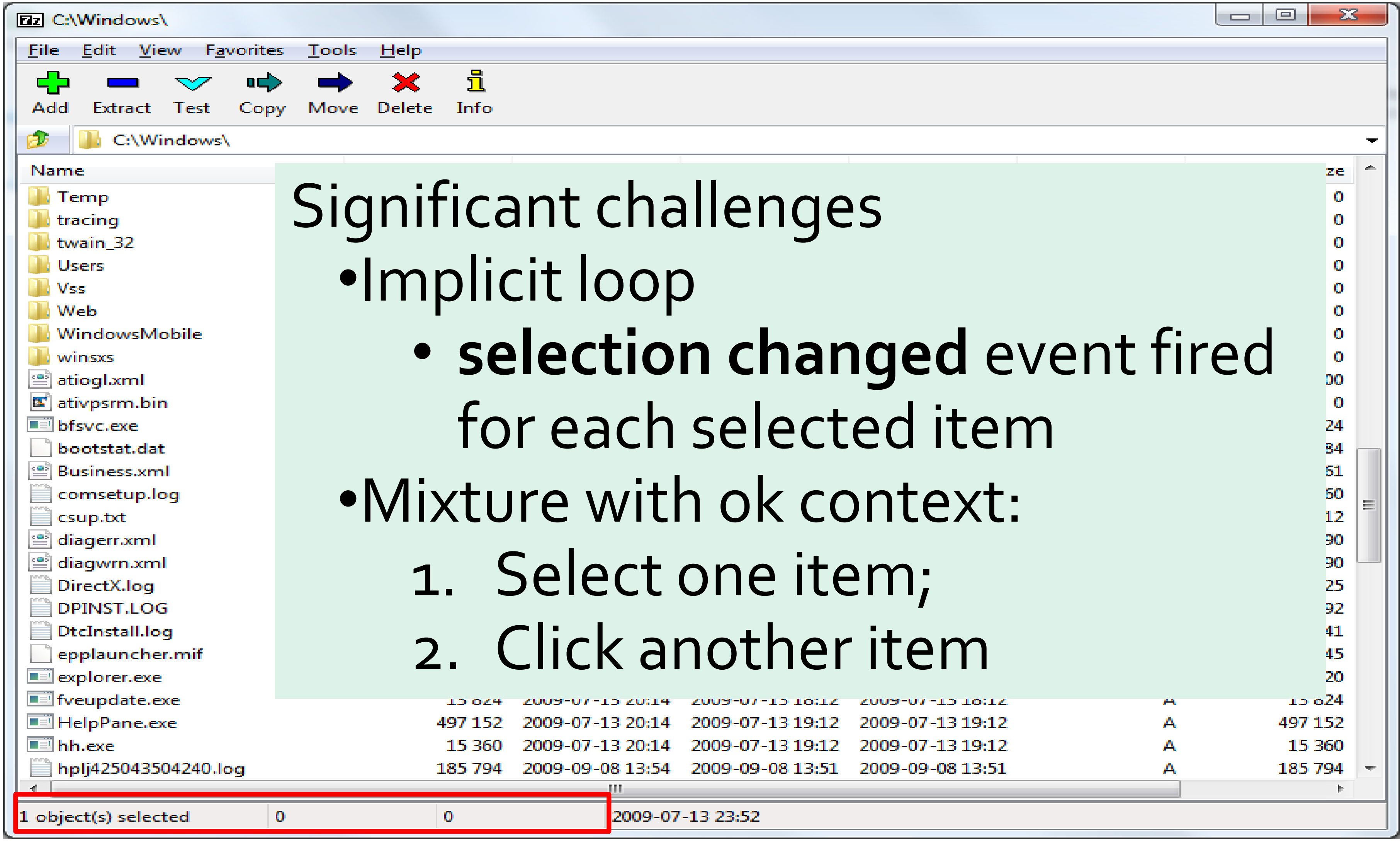
Test Oracle:
Hard to know how
slow is slow enough

Target
Multi-Profile Setting



Example WDPB: 7-Zip File Manager

WDPB Context: 1. Select all items; 2. Click any item



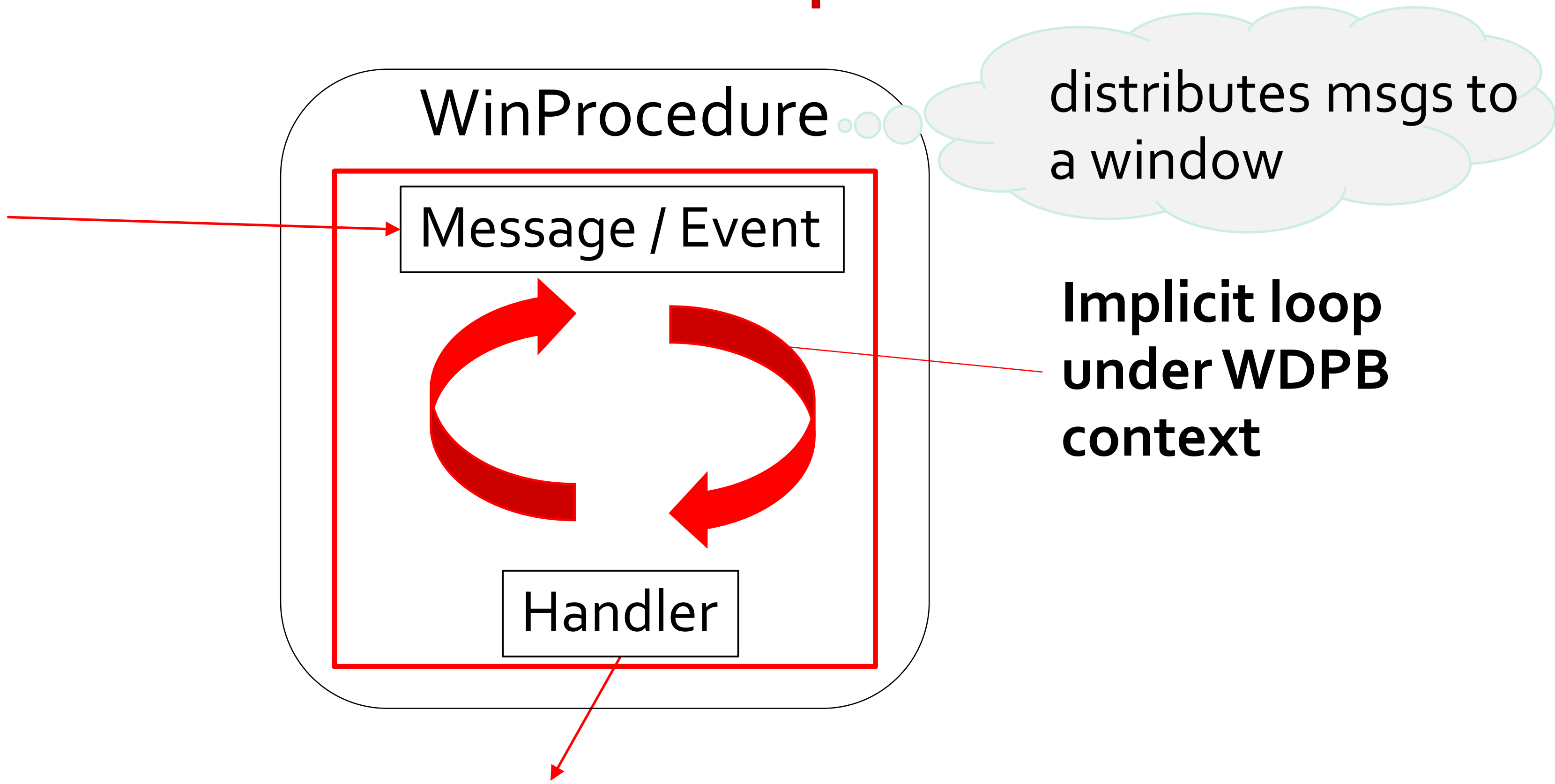
Significant challenges

- Implicit loop
  - **selection changed** event fired for each selected item
- Mixture with ok context:
  1. Select one item;
  2. Click another item

Name						ze
Temp						0
tracing						0
twain_32						0
Users						0
Vss						0
Web						0
WindowsMobile						0
winsxs						0
atiogl.xml						00
ativpsrm.bin						0
bfsvc.exe						24
bootstat.dat						34
Business.xml						51
comsetup.log						50
csup.txt						12
diagerr.xml						90
diagwrn.xml						90
DirectX.log						25
DPINST.LOG						92
DtcInstall.log						41
epplauncher.mif						45
explorer.exe						20
fveupdate.exe	13 824	2009-07-13 20:14	2009-07-13 18:12	2009-07-13 18:12	A	13 824
HelpPane.exe	497 152	2009-07-13 20:14	2009-07-13 19:12	2009-07-13 19:12	A	497 152
hh.exe	15 360	2009-07-13 20:14	2009-07-13 19:12	2009-07-13 19:12	A	15 360
hplj425043504240.log	185 794	2009-09-08 13:54	2009-09-08 13:51	2009-09-08 13:51	A	185 794

```
1 object(s) selected    0    0    2009-07-13 23:52
```

Inside the Example WDPB



distributes msgs to a window

Implicit loop under WDPB context

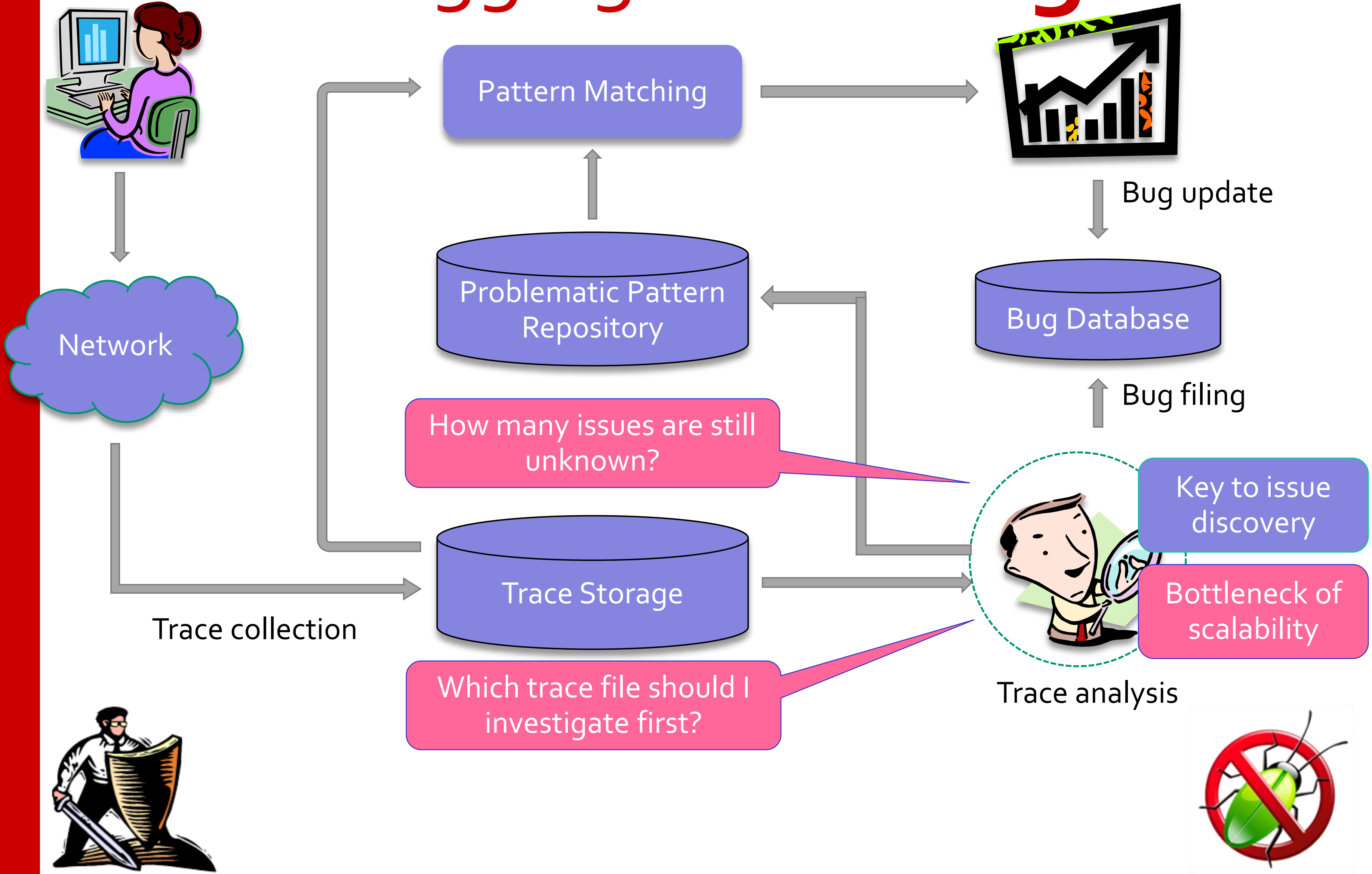
```
void CPanel::OnRefreshStatusBar() {
    ...
    GetOperatedItemIndices(indices);
    _statusBar.SetText(...);
    ...
}
```

Expensive UI updates

Background: ***StackMine***

[Han et al. **ICSE 12**]

Perf Debugging in the **Large**



Pattern Matching

Problematic Pattern Repository

Bug update

Bug Database

Bug filing

Network

How many issues are still unknown?

Key to issue discovery

Bottleneck of scalability

Trace Storage

Trace collection

Which trace file should I investigate first?

Trace analysis

Proposed Approach: DeltaInfer **Context-Sensitive** Delta Inference

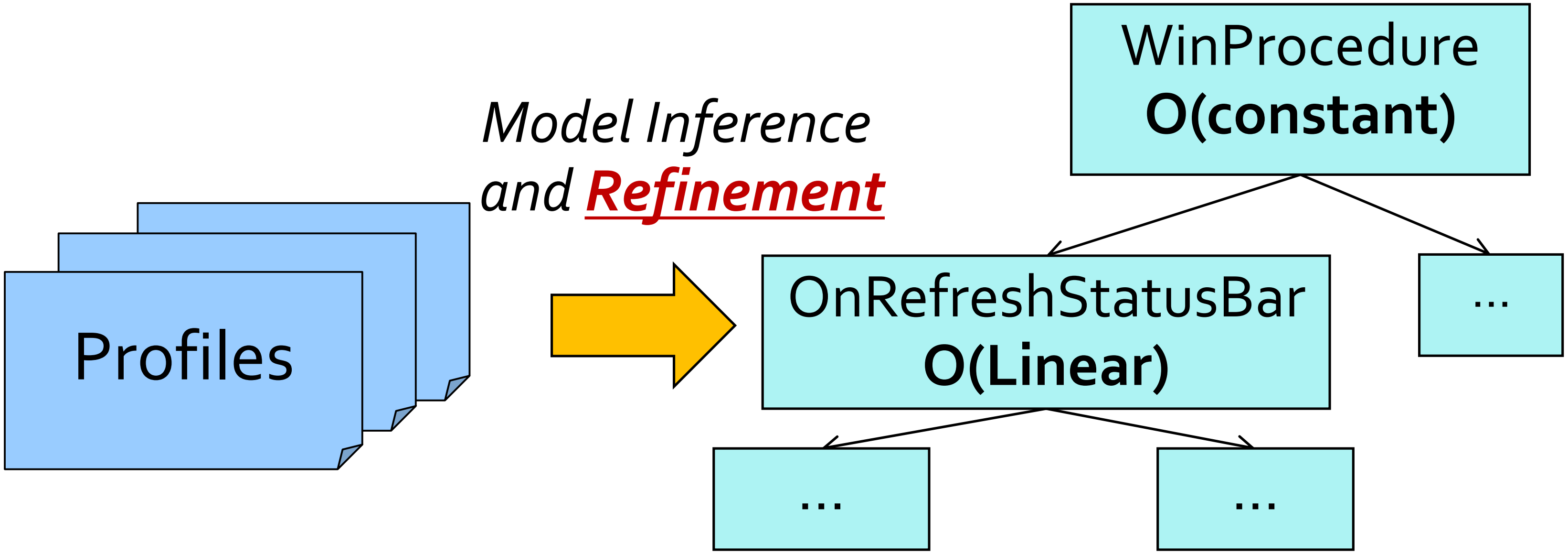
- To gain the power of prediction

***Temporal** Inference: differences between **executions** as complexity models*

- To identify WDPB loops

***Spatial** Inference: differences between **program locations** as WDPB loops*

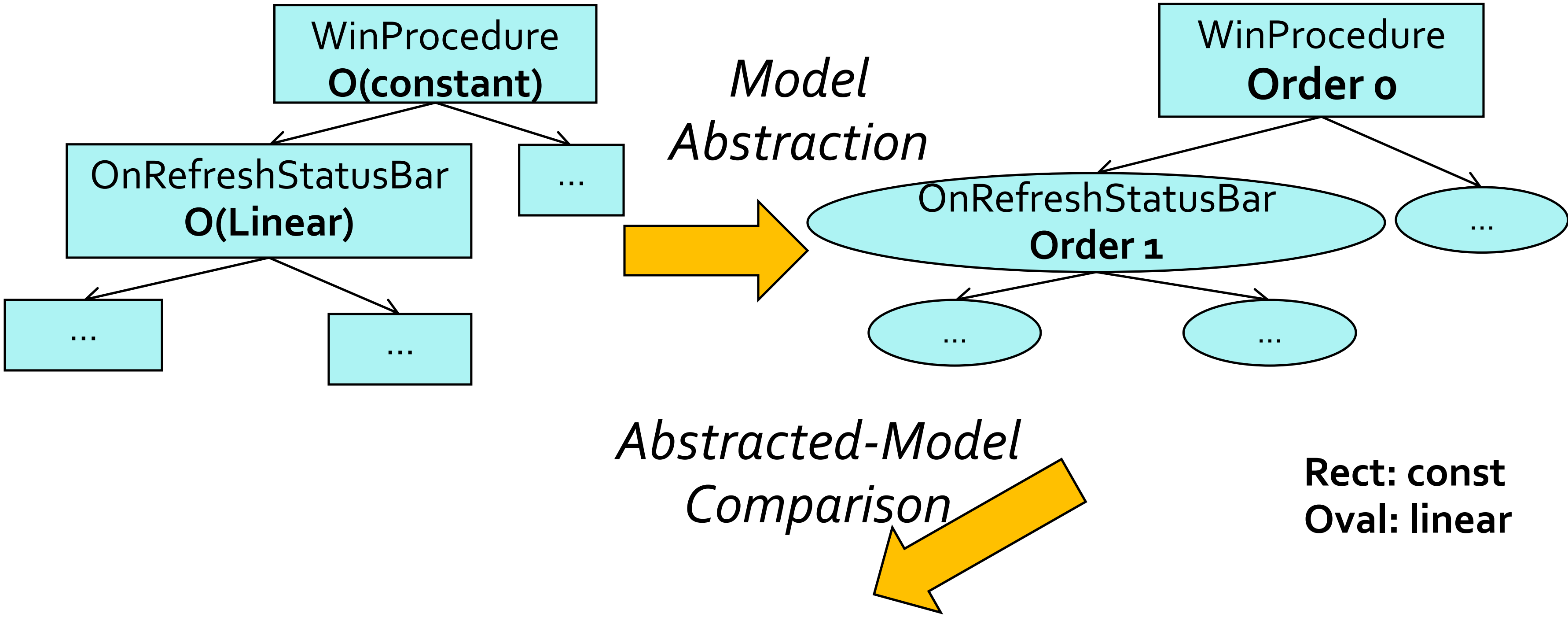
Insight: Temporal Inference



Regression learning + **refinement** to produce **complexity models** of program locations

Insight: Spatial Inference

WDPB loop **raises** complexity models of inside-loop locations to **higher order**



Complexity Transition

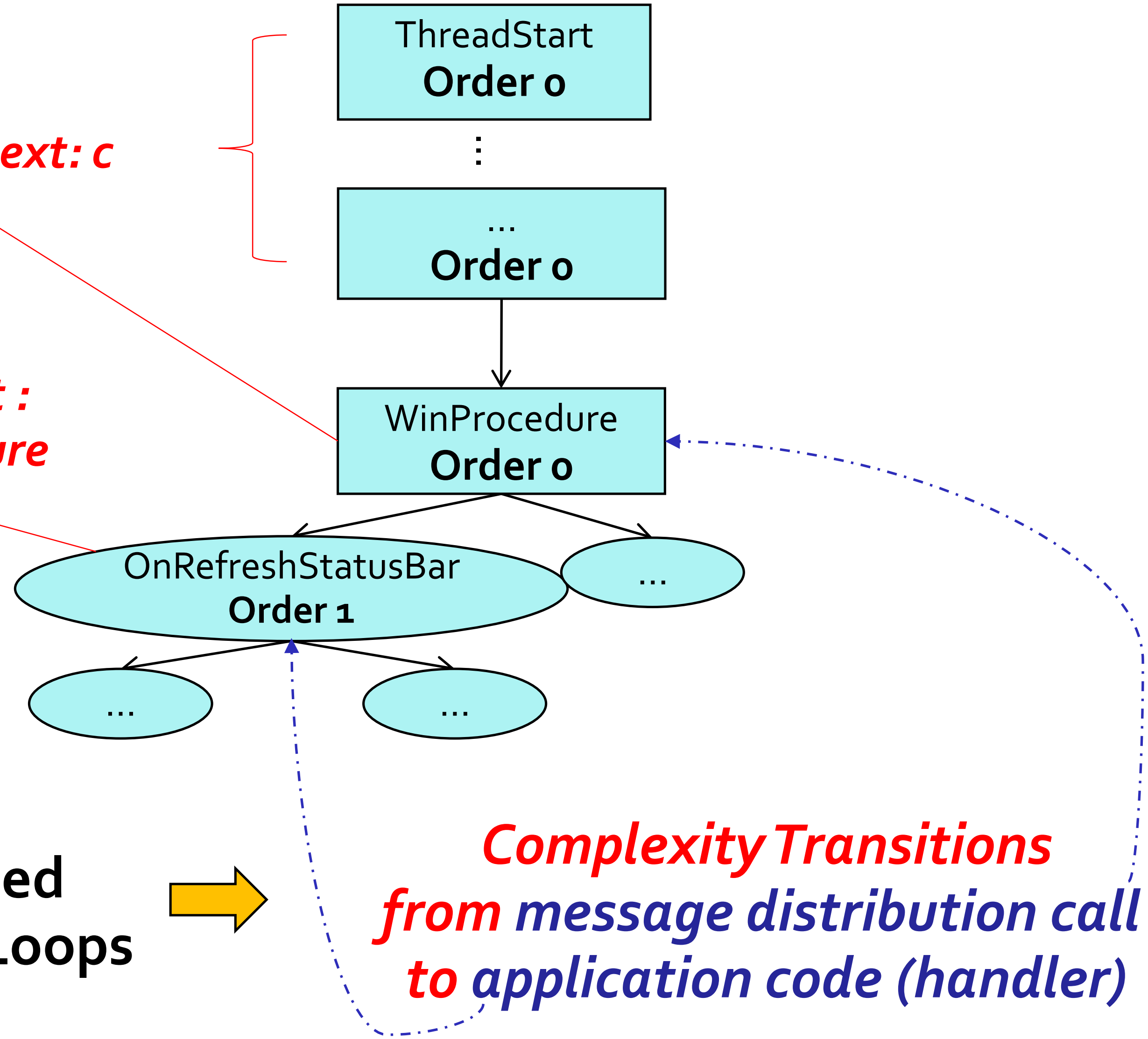
(WinProcedure, {OnRefreshStatusBar , …})

constant → linear

Insight: Context Sensitivity

Calling Context: c

Calling Context :
c+ WinProcedure



ThreadStart
Order 0

⋮

...
Order 0

WinProcedure
Order 0

OnRefreshStatusBar
Order 1

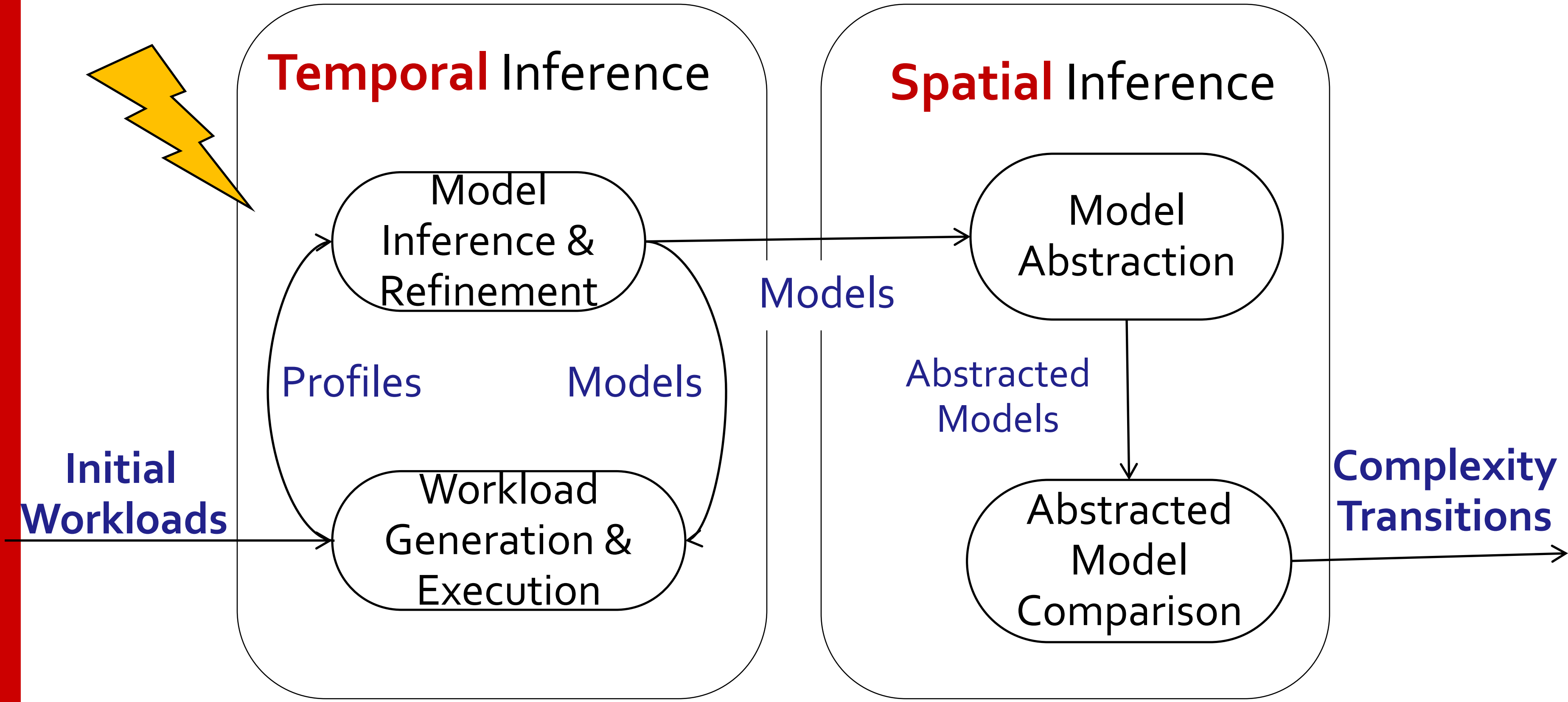
...

...

...

Detected Implicit Loops ⇨ ***Complexity Transitions from message distribution call to application code (handler)***

Overview of DeltaInfer



Workload Generation & Execution

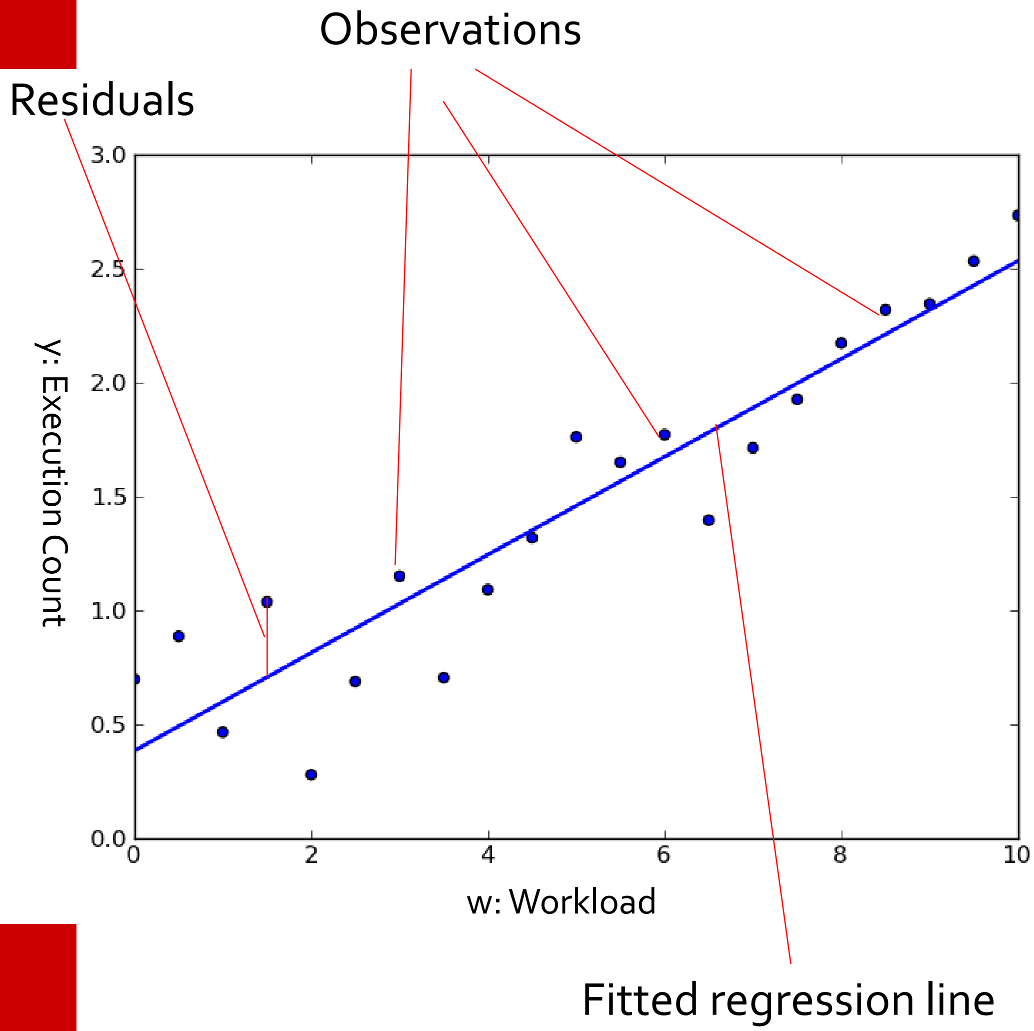
Example scenario: *open a file in text editor*

- Performance metrics
  - execution time
- Performance-relevant workload parameters
  - **# of lines** (focused parameter)
    - Rep value range (RVR): [1, 1280]
    - Initial value/variations (sorted/random inputs)
  - # of character

Example workloads

- # of lines (100, 200, … , 500)
- # of character (100 chars for a line)

Model Inference



Observations

Residuals

Fitted regression line

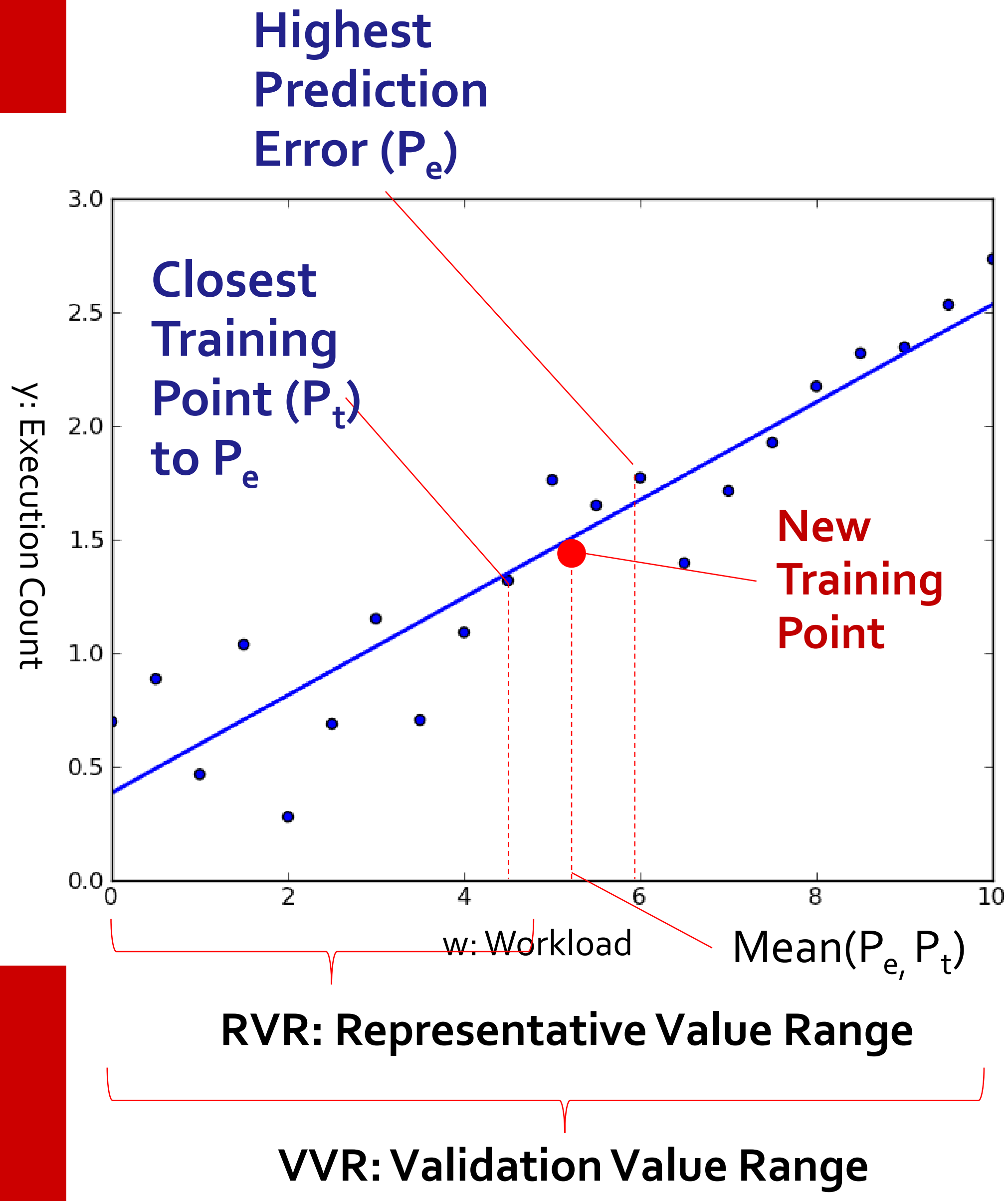
- Linear Regression
  - $y = A + Bw$
- Power-law Regression
  - $y = Aw^B$
- Quality of the model
  - Correlation coefficient R^2

Model Validation

- Model validation measures
  - relative prediction error of inferred models

- Example validation workloads: open a file in text editor
  - Validation value range (VVR): **[1,2560]**
  - Guideline: >= **2** times larger than the RVR
  - Caveat: too **large** RVR is **not** cost-effective

Iterative Refinement



Highest Prediction Error (P_e)

Closest Training Point (P_t) to P_e

New Training Point

Mean(P_e, P_t)

RVR: Representative Value Range

VVR: Validation Value Range

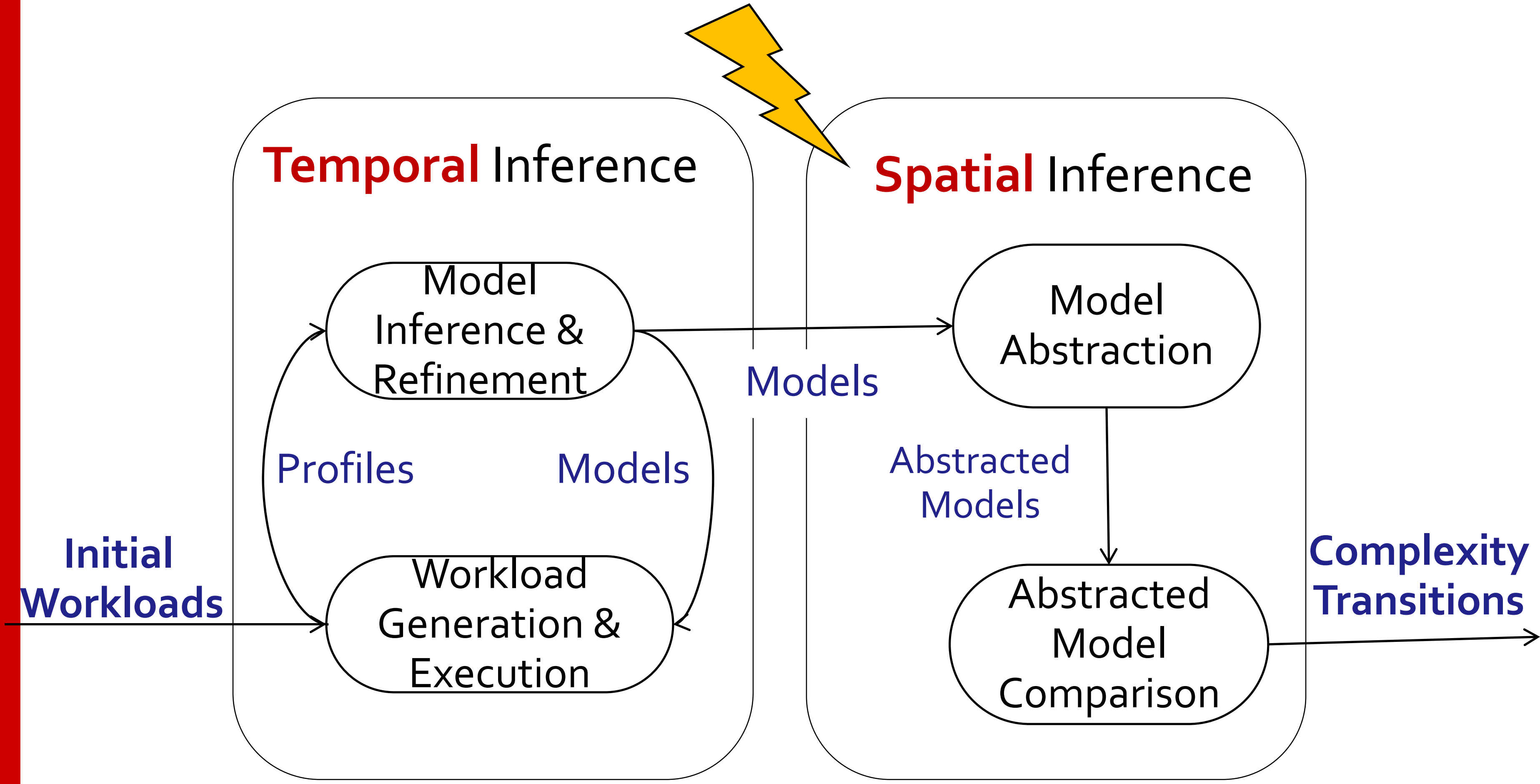
Iterate till

- Accuracy acceptable
- Improvement < threshold

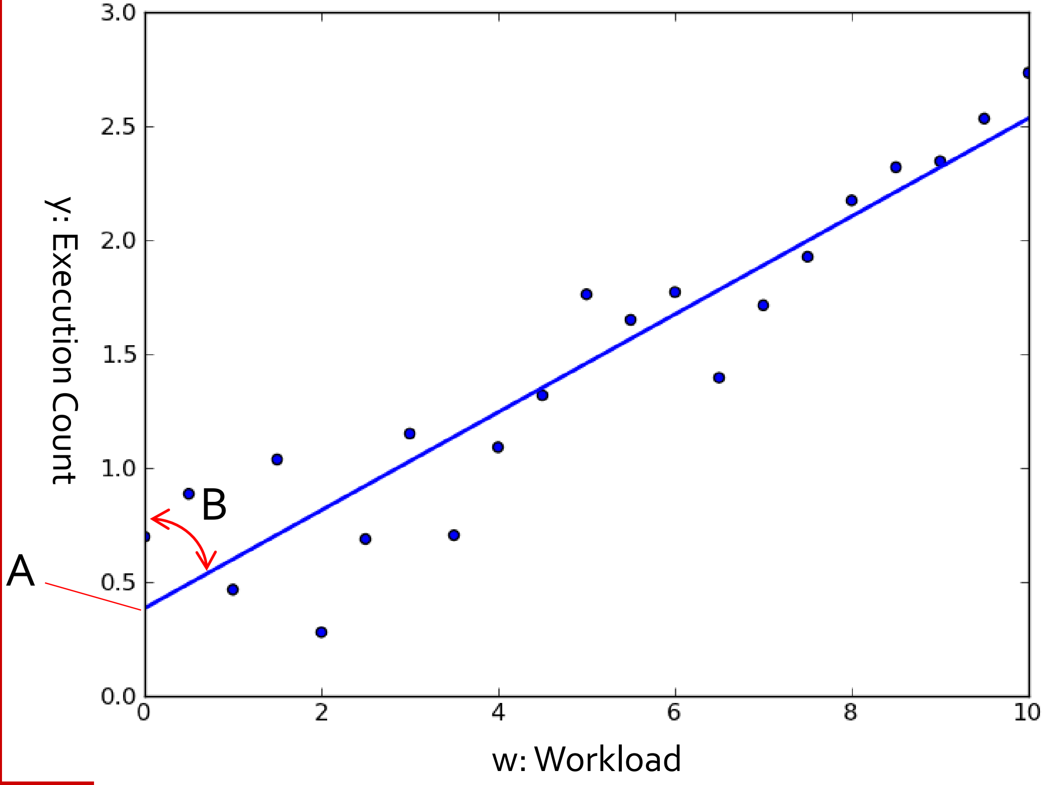
Select a new workload

- Rationale: new workload at **highest-prediction-error areas** improves most

Overview of DeltaInfer

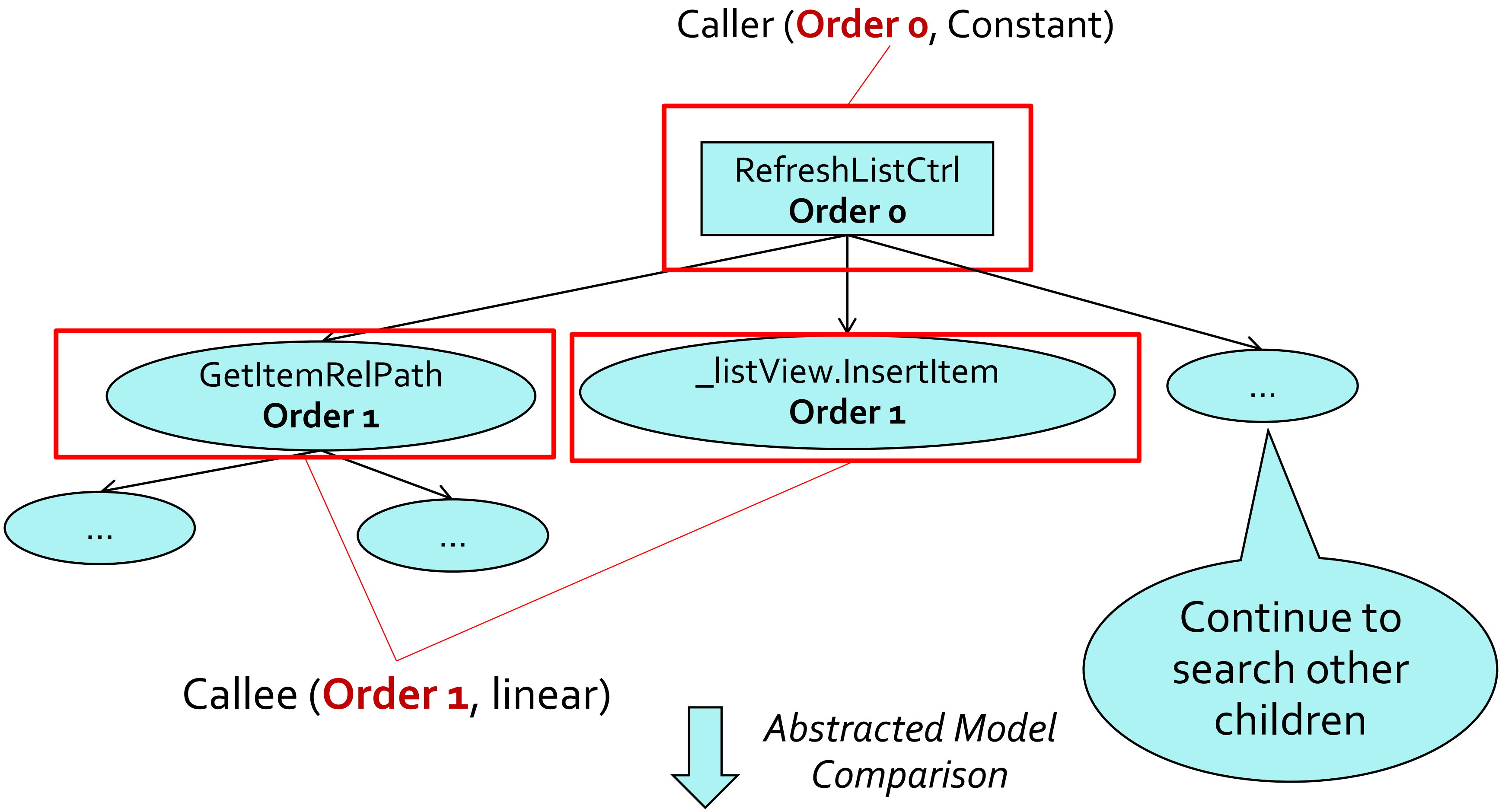


Model Abstraction: Complexity Orders



- Linear model ($y = A + Bw$)
  - **1**, if B > 0
  - **0**, otherwise
- Power-law model ($y = Aw^B$)
  - ***Round(B)***
- Model w/ R^2 below threshold_{R2} (e.g., workload-independent noise)
  - **0**

Inference of Complexity Transitions

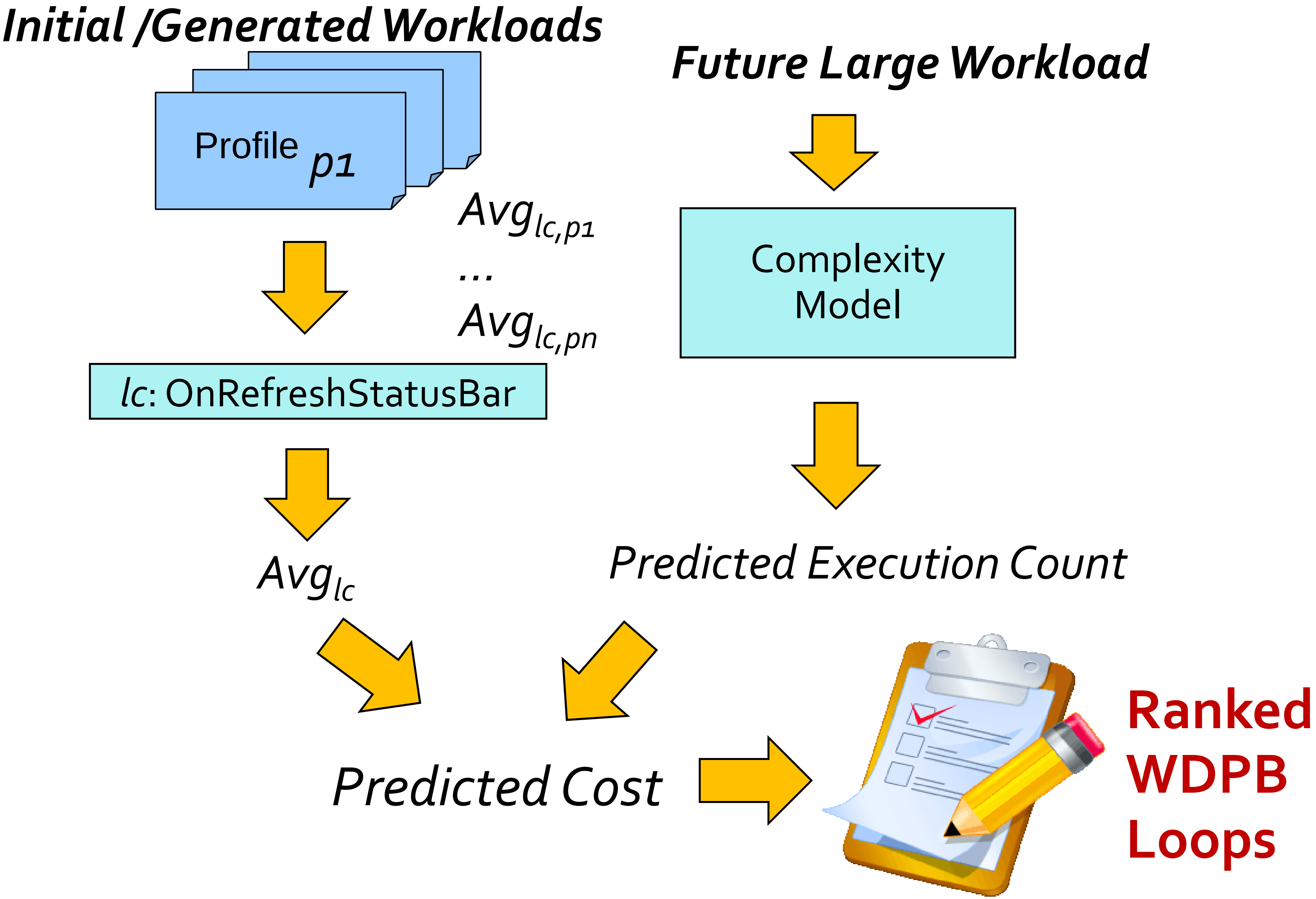

Caller (**Order 0**, Constant)
RefreshListCtrl
Order 0
GetItemRelPath
Order 1
_listView.InsertItem
Order 1
...
...
...
Continue to search other children
Callee (**Order 1**, linear)
Abstracted Model Comparison

Complexity Transition

(RefreshListCtrl, {GetItemRelPath, _listView.InsertItem, ...})

constant ➔ linear

Cost Prediction of Complexity Transitions



Initial /Generated Workloads

Profile *p1*

$Avg_{lc,p1}$

...

$Avg_{lc,pn}$

lc: OnRefreshStatusBar

Avg_{lc}

Future Large Workload

Complexity Model

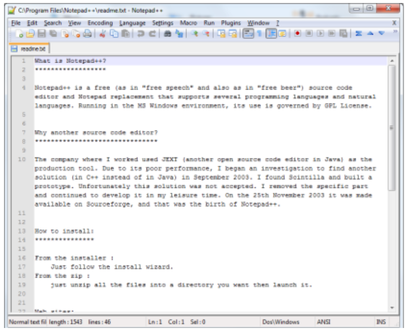
Predicted Execution Count

Predicted Cost

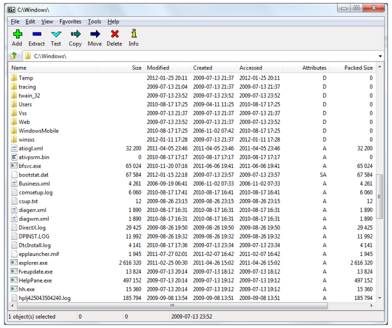
Ranked WDPB Loops

Evaluations of *DeltaInfer*

- Subjects : open-source GUI applications from SourceForge
  - 7-Zip: file manager **7,280** LOC
  - Notepad++: text editor **155,300** LOC



```
What is Notepad++?
******************

Notepad++ is a free (as in "free speech" and also as in "free beer") source code
editor and Notepad replacement that supports several programming languages and natural
languages. Running in the MS Windows environment, its use is governed by GPL License.


Why another source code editor?
*******************************

The company where I worked used JEXT (another open source code editor in Java) as the
production tool. Due to its poor performance, I began an investigation to find another
solution (in C++ instead of in Java) in September 2003. I found Scintilla and built a
prototype. Unfortunately this solution was not accepted. I removed the specific part
and continued to develop it in my leisure time. On the 25th November 2003 it was made
available on Sourceforge, and that was the birth of Notepad++.


How to install:
***************

From the installer :
    Just follow the install wizard.
From the zip :
    just unzip all the files into a directory you want then launch it.


Web sites:
```



Name	Size	Modified	Created	Accessed	Attributes	Packed Size
Temp		2012-01-25 20:11	2009-07-13 21:37	2012-01-25 20:11	D	0
tracing		2009-07-13 21:04	2009-07-13 21:37	2009-07-13 21:37	D	0
twain_32		2009-07-13 23:52	2009-07-13 23:52	2009-07-13 23:52	D	0
Users		2010-08-17 17:25	2009-04-11 11:25	2010-08-17 17:25	D	0
Vss		2009-07-13 21:37	2009-07-13 21:37	2009-07-13 21:37	D	0
Web		2009-07-13 23:52	2009-07-13 21:37	2009-07-13 23:52	D	0
WindowsMobile		2010-08-17 17:25	2006-11-02 07:42	2010-08-17 17:25	D	0
winsxs		2012-01-11 17:28	2009-07-13 21:37	2012-01-11 17:28	D	0
atiogl.xml	32 200	2011-04-05 23:46	2011-04-05 23:46	2011-04-05 23:46	A	32 200
ativpsrm.bin	0	2010-08-17 17:17	2010-08-17 17:17	2010-08-17 17:17	A	0
bfsvc.exe	65 024	2010-11-20 07:16	2011-06-06 19:41	2011-06-06 19:41	A	65 024
bootstat.dat	67 584	2012-01-15 22:18	2009-07-13 23:57	2009-07-13 23:57	SA	67 584
Business.xml	4 261	2006-09-19 06:41	2006-11-02 07:33	2006-11-02 07:33	A	4 261
comsetup.log	6 060	2010-08-17 17:41	2010-08-17 16:41	2010-08-17 16:41	A	6 060
csup.txt	12	2009-08-26 23:15	2009-08-26 23:15	2009-08-26 23:15	A	12
diagerr.xml	1 890	2010-08-17 16:31	2010-08-17 16:31	2010-08-17 16:31	A	1 890
diagwrn.xml	1 890	2010-08-17 16:31	2010-08-17 16:31	2010-08-17 16:31	A	1 890
DirectX.log	29 425	2009-08-26 19:50	2009-08-26 19:50	2009-08-26 19:50	A	29 425
DPINST.LOG	11 992	2009-08-26 19:32	2009-08-26 19:32	2009-08-26 19:32	A	11 992
DtcInstall.log	4 141	2010-08-17 17:36	2009-07-13 23:34	2009-07-13 23:34	A	4 141
epplauncher.mif	1 945	2011-07-27 02:01	2011-02-07 16:42	2011-02-07 16:42	A	1 945
explorer.exe	2 616 320	2011-02-25 00:30	2011-04-26 15:02	2011-04-26 15:02	A	2 616 320
fveupdate.exe	13 824	2009-07-13 20:14	2009-07-13 18:12	2009-07-13 18:12	A	13 824
HelpPane.exe	497 152	2009-07-13 20:14	2009-07-13 19:12	2009-07-13 19:12	A	497 152
hh.exe	15 360	2009-07-13 20:14	2009-07-13 19:12	2009-07-13 19:12	A	15 360
hplj425043504240.log	185 794	2009-09-08 13:54	2009-09-08 13:51	2009-09-08 13:51	A	185 794

1 object(s) selected 0 0 2009-07-13 23:52

- RQs
  - – RQ1: Effectiveness of WDPB Identification
  - – RQ2: Effectiveness of Model Inference/Refinement
  - – RQ3: Effectiveness of Context-Sensitive Analysis

Evaluation Setup - Scenarios

	ID	Scenario	W. Param
7-Zip	(S1)	Open a folder	# files
	(S2)	Rename a file	# files
	(S3)	Select all items and then select the first item	# files
	(S4)	Create a folder	# files
	(S5)	Delete a file	# files
Notepad++	(S6)	Open a file	# lines
	(S7)	Enter a character and save the file	# lines
	(S8)	Go to the last line	# lines
	(S9)	Find a word not present in the file	# chars
	(S10)	Cut and past the first character	# lines

Evaluation Setup – Cont.

- Workload Selection
  - Representative Value Range (RVR)
    - Representative usage **[1, 1280]**
  - Validation Value Range (VVR)
    - Two times as RVR **[1, 2560]**
  - Initial workloads
    - Initial workload groups: **{20,40,80}** , **{100,200,400}**
- Thresholds
  - Max refinement iterations: **20**
  - Threshold for R^2: **0.9**; Improvement threshold: **2%**
  - Prediction-error threshold: **5%**

RQ1: WDPB Identification

- Manually inspect top-rank **complexity transitions**
- Report identified **performance bugs** for confirmation from developers
- Measure **cost coverage** of WDPBs as the workloads increase

Example Bugs of 7-Zip

Complexity Transition (constant to linear)
(RefreshListCtrl, {GetItemRelPath, _listView.InsertItem, …}).

```
HRESULT RefreshListCtrl(...) {
...
  for (UInt32 i = 0; i < numItems; i++) {
     ...
     const UString relPath = GetItemRelPath(i);
     ...
     if (_listView.InsertItem(&item) == -1)
        return E_FAIL;
     }
     ...
  _listView.SortItems(CompareItems, (LPARAM) this);
}
```

WDPB loop

Intensive temp-obj creation/destruction

Implicit WDPB loop!

Complexity Transition (constant to power-law)
(listviewProcedure, {CompareItems, …}).

This bug is triggered in S1, S2, S3, S5

Example Bugs of Notepad++

WDPB loop

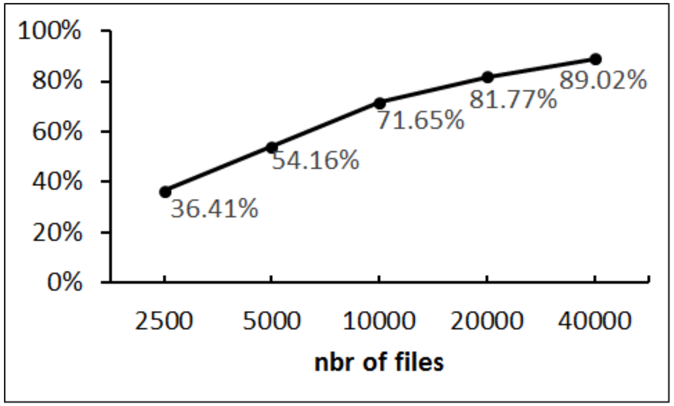
```
bool WrapLines(...) {
    ...
    while (lineToWrap < lastLineToWrap) {
        if (WrapOneLine(surface, lineToWrap)) {
            wrapOccurred = true;
        }
        lineToWrap++;
    }
    ...
}
```

Expensive computation

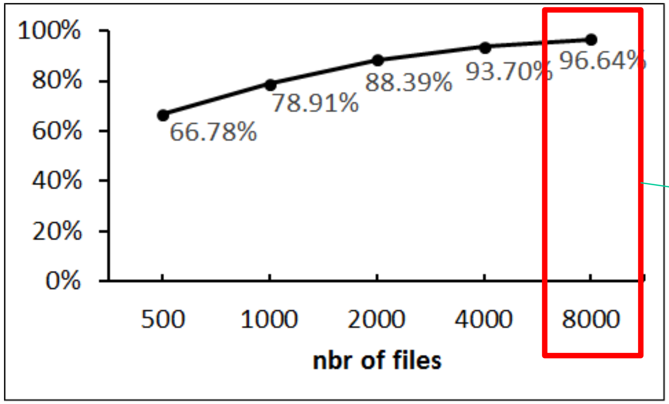
Complexity Transition (constant to linear)
(WrapLines, {WrapOneLine, ...}).

This bug is triggered in S6, S7 S10

Cost Coverage of WDPBs

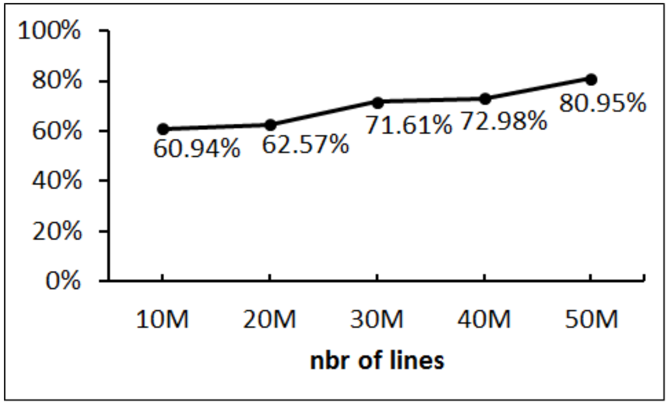


7-Zip FM (S1: open a folder)

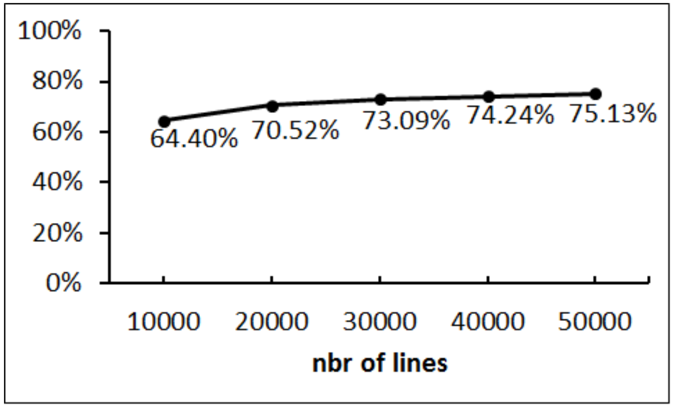


7-Zip FM (S3: select all and click any item)

Nested loop!



Notepad++ (S6: open a file)



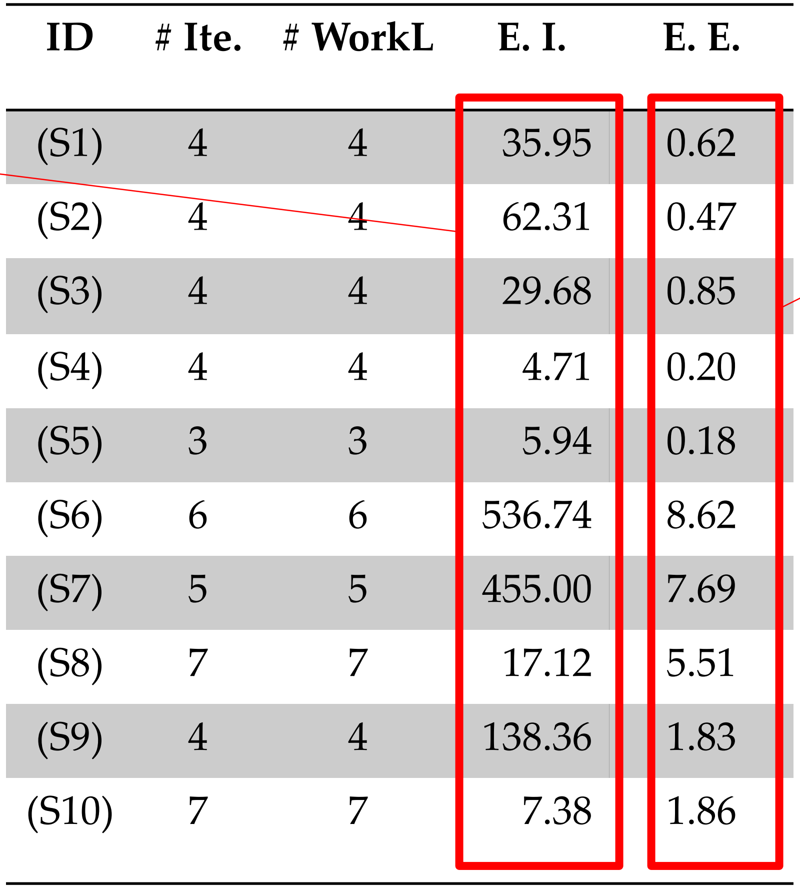
Notepad++ (S9: Find a word)

- Identified WDPBs account for **75%+** of costs ➔ low probability of missing impactful WDPBs
- Cost coverage increases very fast for **nested loops**

Bug Confirmations

- 7-Zip **(5 bugs)**
  - Bugs of **RefreshListCtrl** (S1, S2, S3, S5) confirmed with fix planned for next version
  - Bug of **RefreshStatusBar** (S4) introduced in release on Aug 05 and still remaining in latest release on Aug 11.
- Notepad++: **(5 bugs)**
  - Bug of **wraplines** (S6) found at the forum
  - Bugs caused by **wraplines** (S7,S8, S10), pending
  - New bug on **search for a not-present word** (S9), pending

RQ2: Model Inference and Refinement



ID	# Ite.	# WorkL	E. I.	E. E.
(S1)	4	4	35.95	0.62
(S2)	4	4	62.31	0.47
(S3)	4	4	29.68	0.85
(S4)	4	4	4.71	0.20
(S5)	3	3	5.94	0.18
(S6)	6	6	536.74	8.62
(S7)	5	5	455.00	7.69
(S8)	7	7	17.12	5.51
(S9)	4	4	138.36	1.83
(S10)	7	7	7.38	1.86

Initial Workloads

After Refinement

Avg relative errors of inferred complexity models

- **5 iterations (7 workloads)** to reach avg relative err 2.8%
- **Insensitive** to potential variations of initial workloads

RQ2: Prediction Error of Cost

ID	10 (%)	20 (%)	50 (%)
(S1)	3.18	4.45	6.16
(S2)	2.98	4.07	5.55
(S3)	*1.40	*1.60	*1.86
(S4)	1.65	2.29	3.08
(S5)	1.58	2.19	2.95
(Ave(7-Zip))	*2.35	*3.25	*4.44
(S6)	18.51	6	47.24
(S7)	16.84	5	36.28
(S8)	16.80	7	35.23
(S9)	11.15	4	39.09
(S10)	10.79	7	24.63
(Ave(Notepad++))	14.82	20.97	36.49

X RVR Upper Bound

Developers optimize message processing during idle time

Prediction error

– **4.4%** (7-Zip file manager): excluding S3

– **36.5%** (Notepad++): robust even under complex situations

RQ3: Context Sensitivity

ID	*DeltaInfer*	# L.	InSen.	# Missed by InSen
(S1)	11	10	521	6
(S2)	21	19	579	12
(S3)	17	16	486	10
(S4)	21	19	640	12
(S5)	22	20	546	12
(S6)	10	10	509	3
(S7)	29	14	877	6
(S8)	10	10	526	5
(S9)	20	20	131	0
(S10)	12	11	861	3

Real WDPB loops

- Context helps **reduce false positives & negatives**
  - No context: **> 90%** of identified WDPB loops being **false positives**
  - No context: **40%** of DeltaInfer-identified WDPB loops being **missed**
- Context helps achieve only 14% of identified WDPB loops being **false positives** (top/low-level sys lib calls)

Conclusion

- Predictive approach for WDPBs: ***context-sensitive delta inference***
  - **Temporal inference ➔** complexity models
    - **Deltas** of different **executions** (workloads)
  - **Spatial inference ➔** complexity transitions
    - **Order deltas** of different **locations**
- Evaluations: effectively identifies **impactful WDPBs** (for causing **10** performance bugs)

Microsoft Research

Thank You !

Supported in part by NSF grants
CCF-0845272, CCF-0915400, CNS-0958235, CNS-1160603

Conclusion

- Predictive approach for WDPBs: ***context-sensitive delta inference***
  - **Temporal inference ➔** complexity models
    - **Deltas** of different **executions** (workloads)
  - **Spatial inference ➔** complexity transitions
    - **Order deltas** of different **locations**

- Evaluations: effectively identifies **impactful WDPBs** (for causing **10** performance bugs)

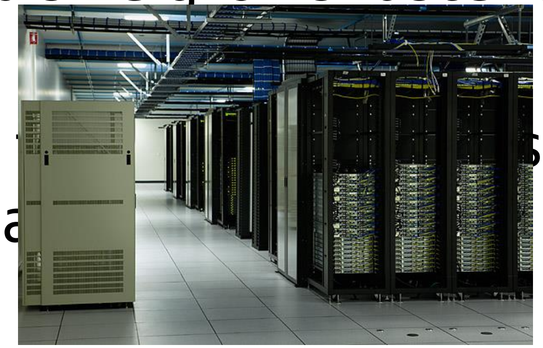
Discussion

- Generalization to Other Types of Applications
- Multiple Workload Parameters
- Value-Dependent Performance Bottlenecks
- Scalability of Scenario-Based Profiling

Software Performance

- An important quality of software
  - A kind of non-functional requirement
  - Characterized by the amount of work accomplished given time and resources
- Software performance matters
  - Software functionality and size grows faster than
  - s been done s avoid
    related mista

Tablet



Data Center

http://betanews.com/2012/05/09/software-performance-matters/

Two Significant Challenges in GUI Applications

- Complex contexts
  - GUI applications are event-driven applications
  - A program location may exhibit different complexities under different contexts
- Implicit WDPB loops
  - Event handlers can be invoked repetitively
    - E.g., selection change events for all items
  - No explicit loop statements
    - Cause challenges to manual inspection and static analysis

Limitations of Traditional Approaches

- Traditional approaches
  - Performance testing (blackbox-random testing or manual testing)
  - Profiling (call-tree profiling and callstack sampling)

41 out of 109 studied performance bugs are due to wrong assumption of workloads. [Jin et al. PLDI 2012]

- Two major issues
  - Insufficiency
    - WDPBs may not surface on given workloads
    - Workload specifications are usually missing or outdated
  - Incompleteness:
    - WDPBs may overshadow other WDPBs

Least-Squares Regression

- Linear Regression
  - Infers: $y = A + Bw$
  - Minimizes: $Q(A,B) = \sum_{i=1}^{k}(y_{l_c,i} - (A + Bw_i))^2$
- Power-law Regression
  - Infers: $y = Aw^B$
  - Minimizes: $Q(A,B) = \sum_{i=1}^{k}(y_{l_c,i} - (Aw_i{}^B))^2$
- How good does the model fit the data points?
  - Correlation coefficient
    - $R^2 = \frac{(\sum_{i=1}^{k} wy - k\overline{wy})^2}{(\sum_{i=1}^{k} w^2 - k\bar{w}^2)(\sum_{i=1}^{k} y^2 - k\bar{y}^2)}$,
    - $\bar{w}$ is the mean of w, $\bar{y}$ is the mean of y

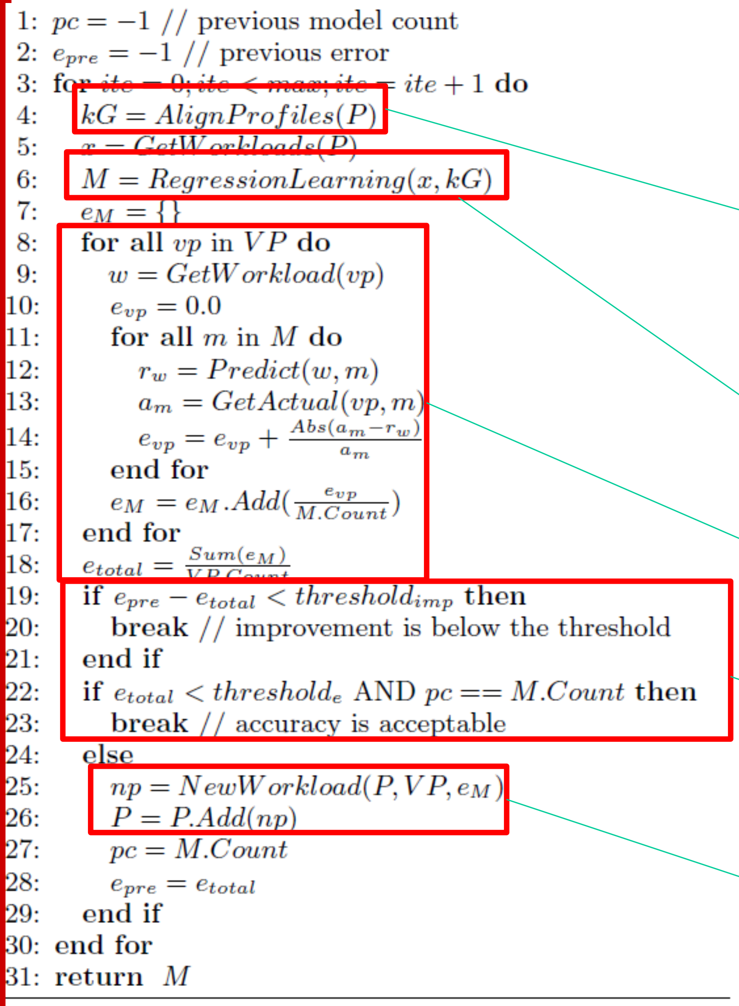
A Few Definitions

- Application *A*, location *l* and cost *y*
- Call graph G(E,V), calling context *c*, and execution profile *P*
- *k-profile Graph:* an annotated call graph, G(E, V), where a location l with its corresponding vertex is annotated with a vector of counters for l on k workloads for each of its calling context c.

Complexity Transitions

- A pair (n,M), such that:
  - 1. *n* is a vertex (method) in the k-profile graph and *M* is a subset of children vertices (callees) of *n*;
  - 2. $f_{n,c}(W)$ is the complexity model of n under the calling context *c*, and $f_{li,ci}\ (W)$ is the complexity model of the location l_i, where l_i is a location in *M* and the calling context c_i is *c* concatenated with *n*.
  - 3. $O(f_{li}, c_i\ (W))$ is at least 1 more than $O(f_{n,c}(W))$;
  - 4. $\forall l_i,\ l_j \in M$, i ≠ j, $O(f_{li,ci}\ (W)) = O(f_{lj\,,cj}\ (W))$.

Model Inference and Refinement



```
$pc = -1$ // previous model count
$e_{pre} = -1$ // previous error
for $ite = 0; ite < max; ite = ite + 1$ do
    $kG = AlignProfiles(P)$
    $x = GetWorkloads(P)$
    $M = RegressionLearning(x, kG)$
    $e_M = \{\}$
    for all $vp$ in $VP$ do
        $w = GetWorkload(vp)$
        $e_{vp} = 0.0$
        for all $m$ in $M$ do
            $r_w = Predict(w, m)$
            $a_m = GetActual(vp, m)$
            $e_{vp} = e_{vp} + \frac{Abs(a_m - r_w)}{a_m}$
        end for
        $e_M = e_M.Add(\frac{e_{vp}}{M.Count})$
    end for
    $e_{total} = \frac{Sum(e_M)}{VP.Count}$
    if $e_{pre} - e_{total} < threshold_{imp}$ then
        break // improvement is below the threshold
    end if
    if $e_{total} < threshold_e$ AND $pc == M.Count$ then
        break // accuracy is acceptable
    else
        $np = NewWorkload(P, VP, e_M)$
        $P = P.Add(np)$
        $pc = M.Count$
        $e_{pre} = e_{total}$
    end if
end for
return $M$
```

Align Profiles

- Align locations using calling contexts
- Extract execution vector for each location under each calling context

Regression Learning

Model Validation

Termination Checks

Select New Workloads

- Assumption: a new workload at the area with the highest prediction error improves most

Cost Prediction of Complexity Transitions

- Compute $avg_{lc,p}$ for each location l_c on each profile p
  - E.g., for p_1, $Cost(refreshList_c)$ = 1s, $ExeCount(refreshList_c)$ = 100, $avg_{lc,p}$ = 1/100 s
- Compute $avg_{lc} = average(avg_{lc,p})$
- Given a workload value w, $pred_{lc,w} = f_{lc}(w)$
- Get $Cost_{lc,w} = pred_{lc,w} * avg_{lc}$