“ Redundancy Analysis and Elimination on Access Patterns of the Windows Applications based on I/O Log Data ”

빅데이터 관리 및 응용 연구실

이준하

Prof. 권혁윤

Contents

Contents

Contents

05

실험 결과

(Experiments Results)

Windows Registry란?

01

Windows Registry의 역할

Windows Registry 구조

Windows Registry 주요 API

1. Windows Registry란? - Windows Registry의 역할

Windows Registry (윈도우 레지스트리)

마이크로소프트 윈도우 운영 체제와 응용 프로그램(프로세스)의 동작에 필요한 가장

필수적이고 기본적인 정보를 저장하고 있는 트리(Tree) 구조의 데이터베이스

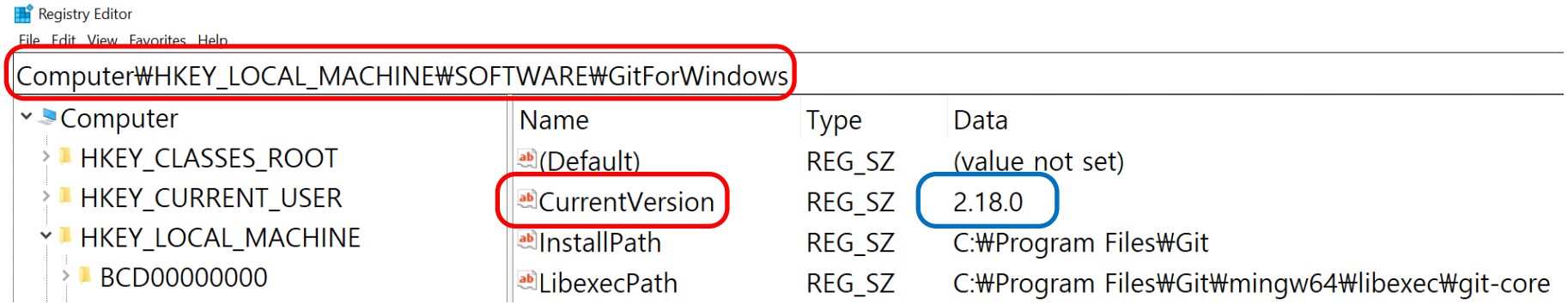
1. Windows Registry란? - Windows Registry 구조

KEY-VALUE

KEY : 폴더의 역할-KEY 안에는 Value를 포함한 또 다른 키(SubKey)를 가질 수 있음

VALUE : KEY 안에 저장되어있는 Data

KEY



VALUE

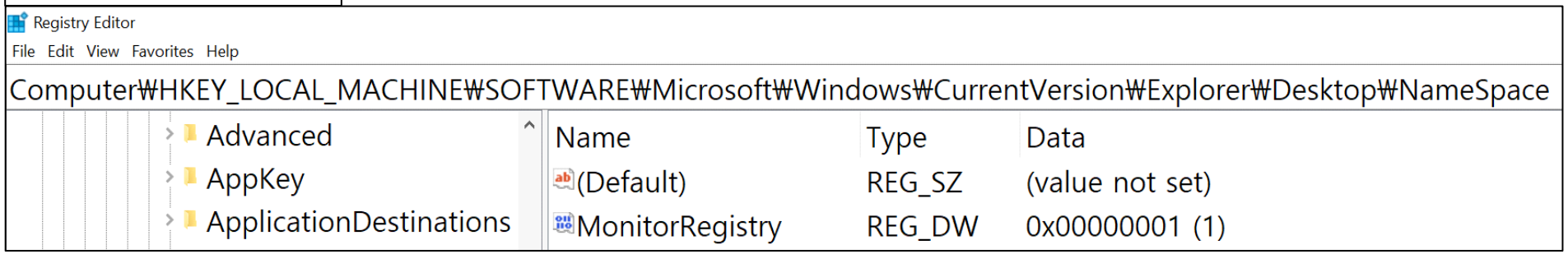
** 레지스트리 편집기(Registry Editor) : Windows Registry에 저장되어있는 데이터를 확인하고 수동으로 편집할 수 있는 프로그램*

1. Windows Registry란? - Windows Registry 주요 API

	API	Description
	RegOpenKey	특정 레지스트리 Key Open
	RegCloseKey	특정 레지스트리 Key Close
Read	RegQueryKey	특정 레지스트리 Key 검색
Read	RegQueryValue	특정 레지스트리 Key에서 주어진 Value 검색
Read	RegEnumKey	특정 레지스트리 Key가 포함하고 있는 SubKey 나열
Read	RegEnumValue	특정 레지스트리 Key에 저장된 Value 나열
Write	RegSetInfoKey	특정 레지스트리 Key에 대한 정보를 저장
Write	RegSetValue	특정 레지스트리 Key 안에 주어진 Value를 저장

1. Windows Registry란? - Windows Registry 주요 API

Registry Editor



Registry Key 및 Value

Process Monitor

Windows Registry에 접근하여 Registry API가 사용됨

** 프로세스 모니터(Process Monitor) : 마이크로소프트 윈도우 운영 체제에서 모든 파일 시스템 활동을 실시간으로 모니터링 할 수 있는 프로그램*

02

Windows Registry에 대한 접근 패턴 분석

Log Data on Windows I/O Operations

Internal Redundancy & Outer Redundancy

Binary Codes 기반 중복 접근 패턴 검증

2. Windows Registry에 대한 접근 패턴 분석

- Log Data on Windows I/O Operations

Windows Version	# of events	# of information	Collect time (seconds)	Dataset size
Windows 10	5,000,000	8	856.48	666.7MB
Windows 8.1	5,000,000	8	954.27	645.9MB
Windows 7	5,000,000	8	752.04	622.6MB
Total	**15,000,000**	**8**	**2562.79**	**1935.2MB**

Windows I/O Log Data 수집 결과

2. Windows Registry에 대한 접근 패턴 분석

- Log Data on Windows I/O Operations

Operation types	Occurrences		Duration (seconds)	
Registry	**11,463,692**	**76.42%**	**824.2535269**	**31.42%**
File system	3,426,005	22.84%	1799.217017	68.58%
Network	93,651	0.62%	0.00011	0.000004%
Process	16,652	0.11%	0.00029	0.000011%

Operation 별 Occurrences 및 Duration 결과

2. Windows Registry에 대한 접근 패턴 분석

- Log Data on Windows I/O Operations

Operations	Occurrences		Duration	
RegOpenKey	3,398,806	29.65%	460.3521	55.85%
RegQueryKey	2,957,266	25.80%	94.4239	11.46%
RegQueryValue	2,127,163	18.56%	85.5787	10.38%
RegCloseKey	1,878,538	16.39%	48.9831	5.94%
RegSetInfoKey	394,141	3.44%	8.6366	1.05%
RegEnumValue	337,098	2.94%	51.702	6.27%
RegEnumKey	229,246	2.00%	8.1201	0.99%
……	……	……	……	……

2. Windows Registry에 대한 접근 패턴 분석

- Internal Redundancy

Internal Redundancy

RegOpenKey와 RegCloseKey 사이에서 동일한 Registry API가 동일한

Registry Key 또는 Value에 대해 반복 수행되는 패턴

2. Windows Registry에 대한 접근 패턴 분석

- Internal Redundancy

Internal Redundancy Case Study #1

Process Name	Operation	Path
Explorer.EXE	RegQueryKey	HKCU₩Software₩Classes
Explorer.EXE	RegQueryKey	HKCU₩Software₩Classes
Explorer.EXE	RegQueryKey	HKCU₩Software₩Classes
Explorer.EXE	RegQueryKey	HKCU₩Software₩Classes
Explorer.EXE	RegQueryKey	HKCU₩Software₩Classes
Explorer.EXE	RegQueryKey	HKCU₩Software₩Classes

특정 Registry API(**RegQueryKey**)가 연속적으로 반복되는 경우

2. Windows Registry에 대한 접근 패턴 분석

- Internal Redundancy

Internal Redundancy Case Study #2

Process Name	Operation	Path
Explorer.EXE	RegOpenKey	HKCR₩CLSID₩{4234D49B-0245-4DF3-B780-3893943456E1}₩InProcServer32
Explorer.EXE	RegQueryKey	HKCR₩CLSID₩{4234d49b-0245-4df3-b780-3893943456e1}₩InProcServer32
Explorer.EXE	RegQueryKey	HKCR₩CLSID₩{4234d49b-0245-4df3-b780-3893943456e1}₩InProcServer32
Explorer.EXE	RegQueryValue	HKCR₩CLSID₩{4234d49b-0245-4df3-b780-3893943456e1}₩InProcServer32₩(Default)
Explorer.EXE	RegQueryKey	HKCR₩CLSID₩{4234d49b-0245-4df3-b780-3893943456e1}₩InProcServer32
Explorer.EXE	RegQueryKey	HKCR₩CLSID₩{4234d49b-0245-4df3-b780-3893943456e1}₩InProcServer32
Explorer.EXE	RegCloseKey	HKCR₩CLSID₩{4234d49b-0245-4df3-b780-3893943456e1}₩InProcServer32

특정 Registry API(**RegQueryKey**)가 비연속적으로 반복되는 경우

2. Windows Registry에 대한 접근 패턴 분석

- Outer Redundancy

Outer Redundancy

RegOpenKey와 RegCloseKey를 포함하는 동일한 Registry API의 집합이

반복 수행되는 패턴

2. Windows Registry에 대한 접근 패턴 분석

- Outer Redundancy

Outer Redundancy Case Study #1

Process Name	Operation	Path
Explorer.EXE	RegOpenKey	HKLM₩System₩CurrentControlSet₩Control₩Session Manager
Explorer.EXE	RegCloseKey	HKLM₩System₩CurrentControlSet₩Control₩Session Manager
Explorer.EXE	RegOpenKey	HKLM₩System₩CurrentControlSet₩Control₩Session Manager
Explorer.EXE	RegCloseKey	HKLM₩System₩CurrentControlSet₩Control₩Session Manager

RegOpenKey와 RegCloseKey가 동일한 패턴으로 반복 수행되는 경우

2. Windows Registry에 대한 접근 패턴 분석

- Outer Redundancy

Outer Redundancy Case Study #2

Process Name	Operation	Path
Explorer.EXE	RegOpenKey	HKCU₩Software₩Microsoft₩OneDrive₩Accounts
Explorer.EXE	RegQueryValue	HKCU₩Software₩Microsoft₩OneDrive₩Accounts₩LastUpdate
Explorer.EXE	RegCloseKey	HKCU₩Software₩Microsoft₩OneDrive₩Accounts
Explorer.EXE	RegQueryKey	HKLM
Explorer.EXE	RegOpenKey	HKCU₩Software₩Microsoft₩OneDrive₩Accounts
Explorer.EXE	RegQueryValue	HKCU₩Software₩Microsoft₩OneDrive₩Accounts₩LastUpdate
Explorer.EXE	RegCloseKey	HKCU₩Software₩Microsoft₩OneDrive₩Accounts

RegOpenKey와 RegCloseKey 사이에서 RegQueryValue가 동일한 패턴으로 반복 수행되는 경우

2. Windows Registry에 대한 접근 패턴 분석

- Outer Redundancy

Outer Redundancy Case Study #3

Process Name	Operation	Path
Explorer.EXE	RegOpenKey	HKCR₩Excel.CSV
Explorer.EXE	RegQueryKey	HKCR₩Excel.CSV
Explorer.EXE	RegQueryKey	HKCR₩Excel.CSV
Explorer.EXE	RegQueryValue	HKCR₩Excel.CSV₩IsShortcut
Explorer.EXE	RegQueryValue	HKCR₩Excel.CSV₩IsShortcut
Explorer.EXE	RegCloseKey	HKCR₩Excel.CSV
Explorer.EXE	RegOpenKey	HKCR₩Excel.CSV
Explorer.EXE	RegQueryKey	HKCR₩Excel.CSV
Explorer.EXE	RegQueryKey	HKCR₩Excel.CSV
Explorer.EXE	RegQueryValue	HKCR₩Excel.CSV₩IsShortcut
Explorer.EXE	RegQueryValue	HKCR₩Excel.CSV₩IsShortcut
Explorer.EXE	RegCloseKey	HKCR₩Excel.CSV

RegOpenKey와 RegCloseKey 사이에서 **다수의 Registry API**가 동일한 패턴으로 반복 수행되는 경우

2. Windows Registry에 대한 접근 패턴 분석

- Binary Codes 기반 중복 접근 패턴 검증

IDA Pro

RegOpenKey

↓

RegQueryValue

RegQueryValue

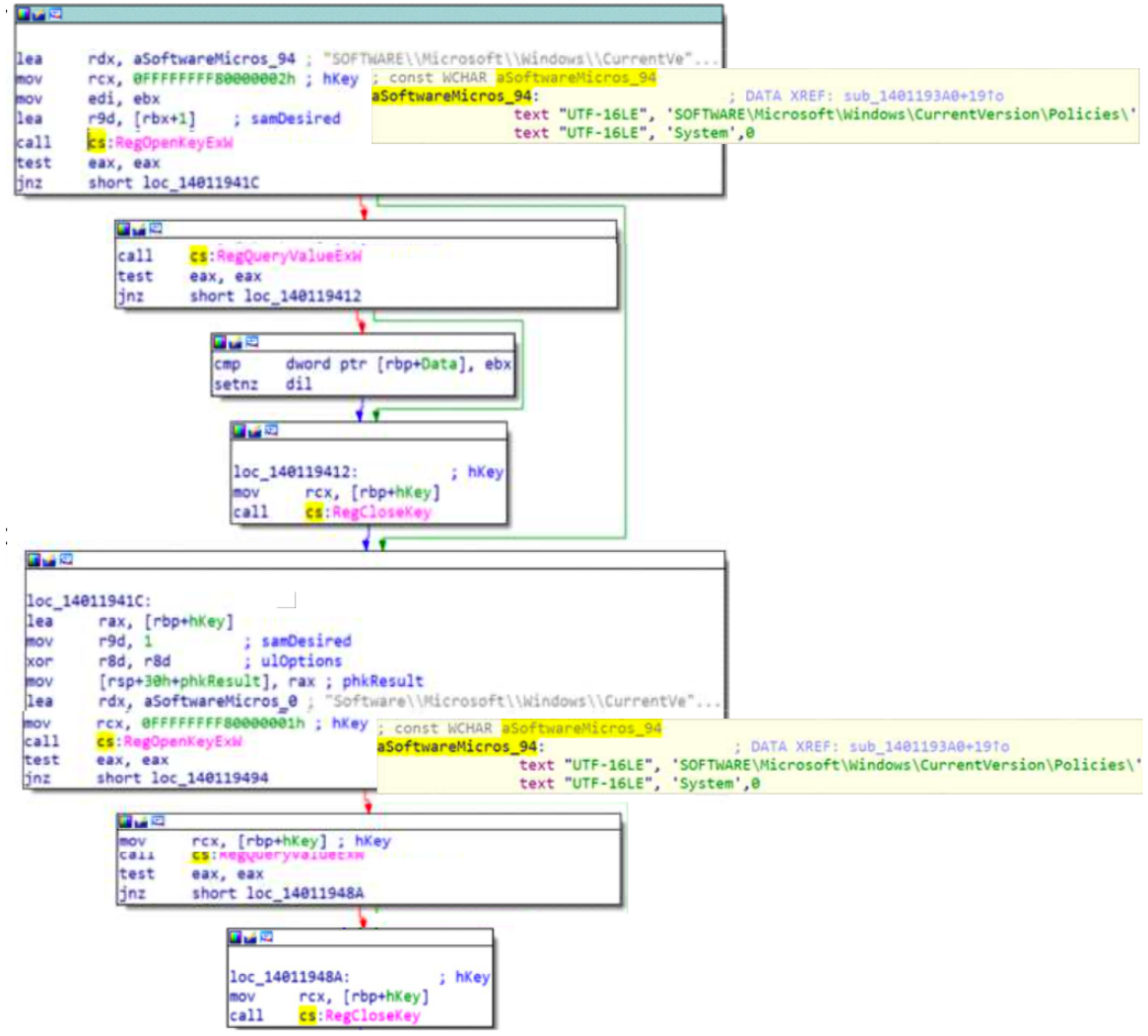
Internal Redundancy

** 아이다 프로(IDA Pro) : 컴퓨터 소프트웨어용 디스어셈블러 (기계어 코드로부터 어셈블리어 소스 코드를 생성)*

2. Windows Registry에 대한 접근 패턴 분석

- Binary Codes 기반 중복 접근 패턴 검증

IDA Pro



RegOpenKey
RegQueryValue
RegCloseKey

↓

RegOpenKey
RegQueryValue
RegCloseKey

Outer Redundancy

03

중복 접근 탐지 알고리즘

Internal Redundancy Detection Algorithm

Outer Redundancy Detection Algorithm

3. 중복 접근 탐지 알고리즘

- Internal Redundancy Detection Algorithm

```
15 def detect_internal_redundancy (process_data) :
16     internal_redundant_patterns  = {}
17     i = 0
18     while ( i < len(process_data) ) :
19         if  (process_data[i ]['Operation'] == 'RegOpenKey') :
20             j = i
21             while ( ! (process_data[j ]['Operation'] == 'RegCloseKey'
                           and process_data[j ]['Path'] == process_data[i ]['Path'] )
                           and process_data[j ]['Operation'] != 'Reg_WriteOperations') :
22                 if (process_data[i ]['Path'] in process_data[j ]['Path']):
23                     internal_pattern = (process_data[j ]['BS_ID'], process_data[j ]['Operation'],
                                           process_data[j ]['Path'])
24                     if  (process_data[j ] not in internal_redundant_patterns.keys()) :
25                         internal_redundant_patterns.append(internal_pattern, 1)
26                     else :
27                         (internal_pattern, count) = internal_redundant_patterns.search(internal_pattern)
28                         internal_redundant_patterns.update(internal_pattern, count+1)
29                 j = j + 1
30         i = i + 1
31     return internal_redundant_patterns
```

조건 1.

RegOpenKey - … … - RegCloseKey

조건 2.

동일한 경로에서 수행된,

동일한 Registry API일 때

조건 3.

Key 또는 Value에 대한 데이터를 읽어오는

Registry API일 때

↓

Internal Redundant Pattern에 추가

3. 중복 접근 탐지 알고리즘

- Internal Redundancy Detection Algorithm

Patterns	Total	Consecutive	Inconsecutive	Total duplication	Portion of duplication
RegOpenKey	3,398,806	83,922	1,813,408	1,897,330	55.82%
RegQueryKey	2,957,266	669,249	1,666,708	2,335,957	78.99%
RegQueryValue	2,127,163	332,372	764,171	1,096,543	51.55%
RegCloseKey	1,878,538	4,950	937,221	942,171	50.15%
RegEnumValue	337,098	269,906	45,535	315,441	93.58%
RegEnumKey	229,246	84,882	113,461	198,343	86.52%
RegQueryKeySecurity	2,906	1,189	774	1,963	67.55%
RegLoadKey	705	12	192	204	28.94%
RegQueryMultipleValueKey	383	94	134	228	59.53%
Others	5,31,581	0	0	0	0.00%
Total	11,463,692	1,446,576	5,341,604	6,788,180	59.21%

11,463,692개의 Registry API 중 6,788,180개 (약 59.21%)가 Internal Redundancy

3. 중복 접근 탐지 알고리즘

- Outer Redundancy Detection Algorithm

```
1  def detect_outer_redundancy  (process_data) :
2      outer_redundant_patterns = []
3      i = 0
4      while ( i < len(process_data) ) :
5          if (process_data['Operation][i ] == 'RegOpenKey') :
6              outer_pattern = []
7              j = i
8              while ( ! (process_data['Operation'][j ] == 'RegCloseKey'
                          and process_data['Path][j ] == process_data['Path][i ]) ) :
9                  if (process_data['Path'][i ] in process_data['Path'][j ]):
10                     outer_pattern.append((process_data['BS_ID'][j ],
                           process_data['Operation'][j ], process_data['Path'][j ]))
11                 j = j + 1
12             if (outer_pattern not in outer_redundant_patterns.keys()):
13                 outer_redundant_patterns.append(outer_pattern, 1)
14             else :
15                 (outer_pattern, count) = outer_redundant_patterns.search(outer_patter
16                 outer_redundant_patterns.update(outer_pattern, count+1)
17         i = i + 1
18     return outer_redundant_patterns
```

조건 1.

RegOpenKey - … … - RegCloseKey

조건 2.

동일한 경로에서 수행된 Registry API일 때

↓

Outer Redundant Pattern에 추가

3. 중복 접근 탐지 알고리즘

- Outer Redundancy Detection Algorithm

Pattern No.	Patterns	Total	Portion	Duplication	Portion of duplication
Outer1	RegOpenKey-RegSetInfoKey-RegQueryValue-RegCloseKey	181,253	15.08%	105,425	58.16%
Outer2	RegOpenKey-RegCloseKey	181,178	15.07%	125,353	69.19%
Outer3	RegOpenKey-RegQueryValue-RegCloseKey	160,261	13.33%	69,600	43.43%
Outer4	RegOpenKey-RegQueryKey-RegQueryKey-RegQueryValue-RegCloseKey	107,535	8.95%	82,468	76.69%
Outer5	RegOpenKey-RegOpenKey-RegCloseKey	85,109	7.08%	64,090	75.30%
Outer6	RegOpenKey-RegQueryKey-RegQueryKey-RegQueryValue-RegQueryValue-RegCloseKey	66,781	5.56%	51,379	76.94%
Outer7	RegOpenKey-RegQueryKey-RegOpenKey-RegCloseKey	58,197	4.84%	48,241	82.89%
Outer8	RegOpenKey-RegQueryValue-RegQueryValue-RegCloseKey	56,020	4.66%	2,565	4.58%
Outer9	RegOpenKey-RegSetInfoKey-RegCloseKey	19,943	1.66%	14,610	73.26%
Outer10	RegOpenKey-RegQueryKey-RegQueryKey-RegCloseKey	18,820	1.57%	6,326	33.61%
Others		267,052	22.21%	121,158	45.37%
Total		1,202,149		691,215	57.50%

1,202,149개의 Registry API 패턴 중 691,215개 (약 57.50%)가 Outer Redundancy

04

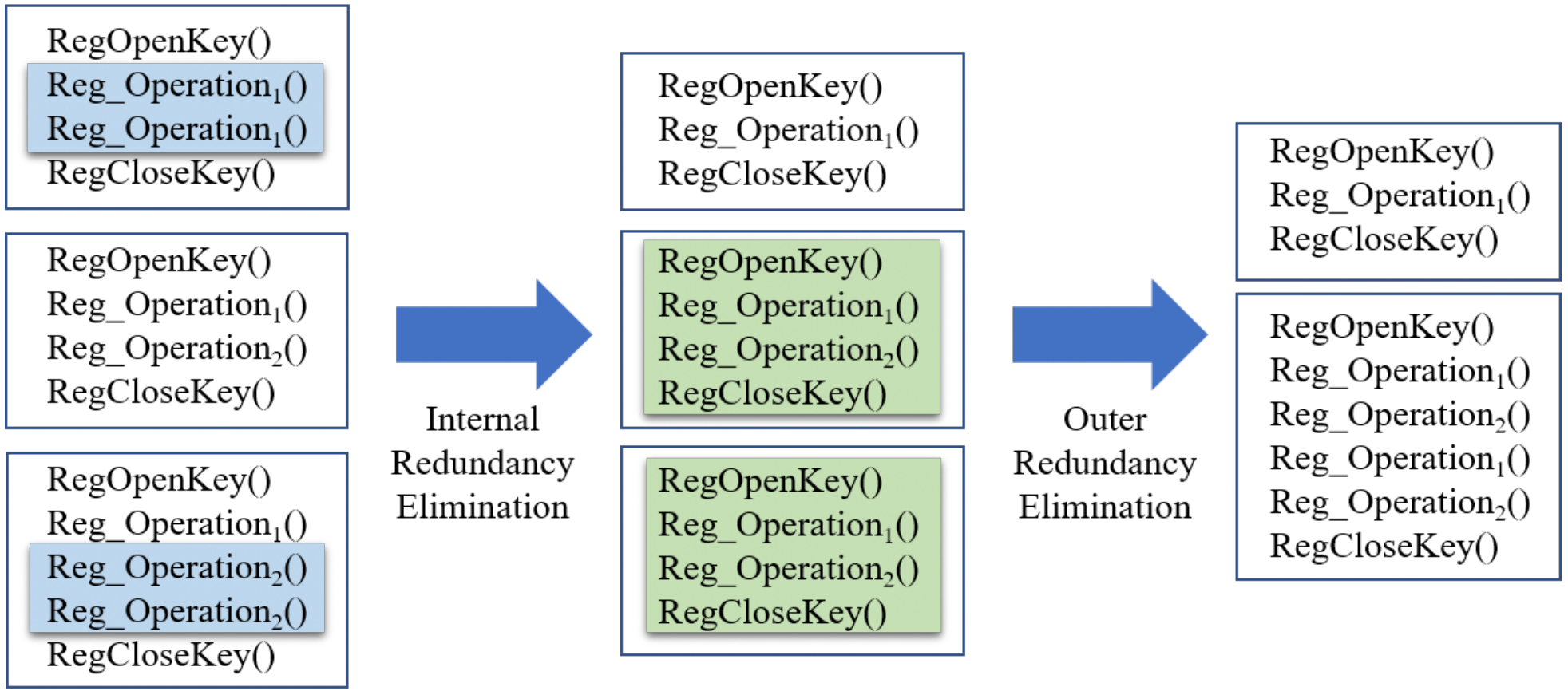
Windows Registry에 대한 접근 패턴 제거

Two-Level Redundancy Elimination Method

Event-Driven Method

4. Windows Registry에 대한 중복 접근 패턴 제거

- Two-Level Redundancy Elimination Method



Internal Redundancy Elimination ➔ Outer Redundancy Elimination

4. Windows Registry에 대한 중복 접근 패턴 제거

- Two-Level Redundancy Elimination Method

Process Name	Operation	Path
Explorer.EXE	RegOpenKey	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001
Explorer.EXE	RegQueryValue	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001₩Name
Explorer.EXE	RegQueryValue	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001₩Name
Explorer.EXE	RegQueryValue	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001₩Name
Explorer.EXE	RegQueryValue	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001₩Name
Explorer.EXE	RegCloseKey	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001
Explorer.EXE	RegOpenKey	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001
Explorer.EXE	RegQueryValue	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001₩Name
Explorer.EXE	RegCloseKey	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001

Before Internal Redundancy Elimination

4. Windows Registry에 대한 중복 접근 패턴 제거

- Two-Level Redundancy Elimination Method

Internal Redundancy Elimination

Original pattern

```
1   // Perform both registry operations and
2   // main operation for n times
3   RegOpenKey()
4
5   for i in n:
6       Reg_Operations()
7       MainOperations()
8
9   RegCloseKey()
10
```

Internal redundancy elimination

```
1   // Perform registry operations once
2   RegOpenKey()
3   Reg_Operations()
4
5   // Perform only main operations for n times
6   for i in n:
7       MainOperations()
8
9   RegCloseKey()
10
```

4. Windows Registry에 대한 중복 접근 패턴 제거

- Two-Level Redundancy Elimination Method

Process Name	Operation	Path
Explorer.EXE	RegOpenKey	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001
Explorer.EXE	RegQueryValue	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001₩Name
Explorer.EXE	RegCloseKey	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001
Explorer.EXE	RegOpenKey	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001
Explorer.EXE	RegQueryValue	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001₩Name
Explorer.EXE	RegCloseKey	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩Defaults₩Provider Types₩Type 001

After Internal Redundancy Elimination

4. Windows Registry에 대한 중복 접근 패턴 제거

- Two-Level Redundancy Elimination Method

Outer Redundancy Elimination

Original pattern

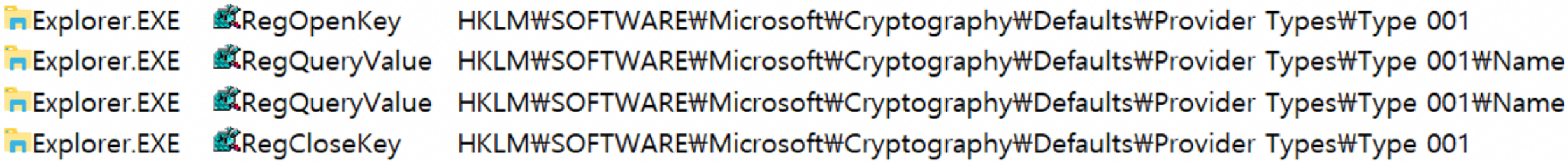
```
1   // Perform both registry operations and
2   // main operation for n times
3
4   for i in n:
5       RegOpenKey()
6       Reg_Operations()
7       MainOperations()
8       RegCloseKey()
9
10
11
```

Outer redundancy elimination

```
1   // Perform RegOpenKey once
2   RegOpenKey()
3
4   // Perform Reg_Operations and
5   // main operations for n times
6   for i in n :
7       Reg_Operations()
8       MainOperations()
9
10  // Perform RegCloseKey once
11  RegCloseKey()
```

4. Windows Registry에 대한 중복 접근 패턴 제거

- Two-Level Redundancy Elimination Method



After Outer Redundancy Elimination

4. Windows Registry에 대한 중복 접근 패턴 제거

- Two-Level Redundancy Elimination Method

Pattern No.	Original	Outer deduplicated result	Internal-then-Outer redundancy elimination	Improved effect of applying outer deduplication after internal deduplication
Outer1	181253	75,828	13,880	81.70%
Outer2	181178	55,825	26,738	52.10%
Outer3	160261	90,661	50,371	44.44%
Outer4	107535	25,067	1,504	94.00%
Outer5	85109	21,019	14,154	32.66%
Outer6	66781	15,402	19	99.88%
Outer7	58197	9,956	3,759	62.24%
Outer8	56020	53,455	8,167	84.72%
Outer9	19943	5,333	3,757	29.55%
Outer10	18820	12,494	1,167	90.66%
Others	267052	145,894	110,795	24.06%
Total	1202149	510,934	234,311	54.14%

4. Windows Registry에 대한 중복 접근 패턴 제거

- Two-Level Redundancy Elimination Method

Operations	Original	Internal Deduplication	Outer Deduplication	Effect of two-level deduplication
RegQueryKey	2957266	621309	621309	78.99%
RegOpenKey	3398806	1501476	940219	72.34%
RegQueryValue	2127163	1030620	1030620	51.55%
RegClosekey	1878538	936367	384443	79.53%
RegEnumValue	337098	21657	21657	93.58%
RegEnumKey	229246	30903	30903	86.52%
RegQueryKeySecurity	2906	943	943	67.55%
RegLoadKey	705	501	501	28.94%
RegQueryMultipleValueKey	383	155	155	59.53%
Others(Write Operations)	531581	531581	531581	0.00%
Total	11463692	4675512	3562331	68.93%

4. Windows Registry에 대한 중복 접근 패턴 제거

- Event-Driven Method

Event-Driven Method

특정 Event에 의해 프로그램의 제어 흐름이 결정되게 하는 방법

Event-Driven Method in Windows Registry

특정 Registry KEY 또는 VALUE에 대한 정보가 수정되었을 때를 Event로 설정하여,

그 이후에 특정 Registry API가 수행되게 하는 방법 → **Side Effect 제거**

4. Windows Registry에 대한 중복 접근 패턴 제거

- Event-Driven Method

Main Thread

```
1   // Perform registry operations once
2   Reg_Operations()
3   // Call event handler in the other thread
4   Call_EventHandler()
5
6   // Perform only main operations for n times
7   for i in n :
8       MainOperations()
9
10
11
```

Event Handler Thread

```
1   // Register event to catch the updates
2   // for a target registry
3   RegNotifyChangeKeyValue()
4
5   // Work infinitely to catch the registered event
6   while (1) :
7       // Wati until the registered event is caught
8       WaitForSingleObject()
9
10  // Read updated registry values
11  Reg_Operations()
```

실험 결과

05

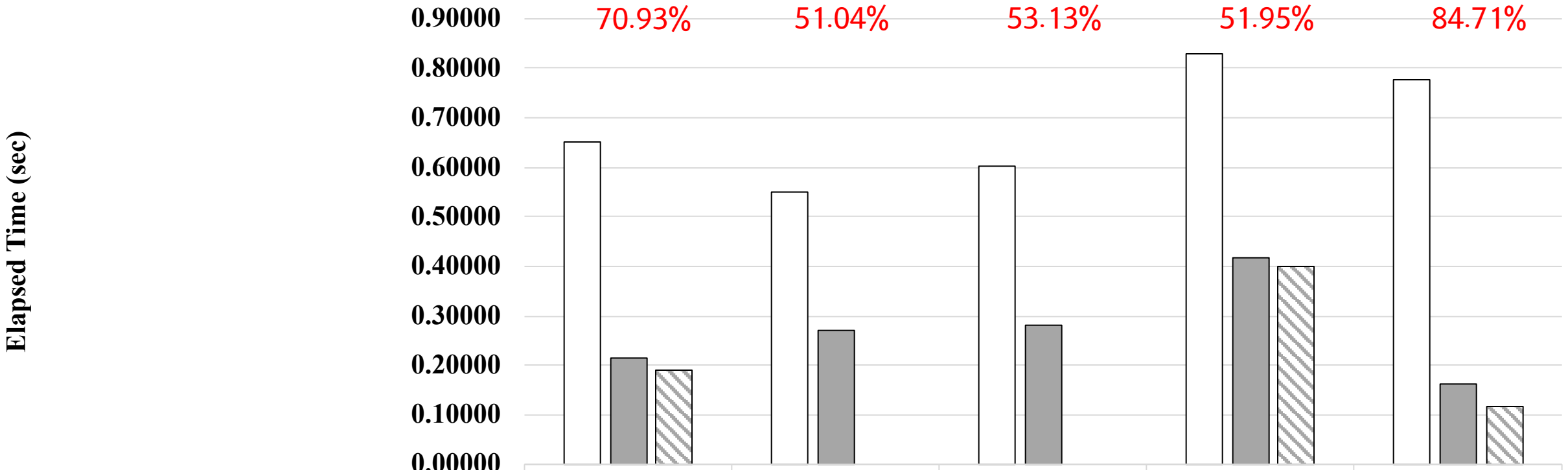
Internal Redundancy Elimination

Outer Redundancy Elimination

Two-Level Method Elimination

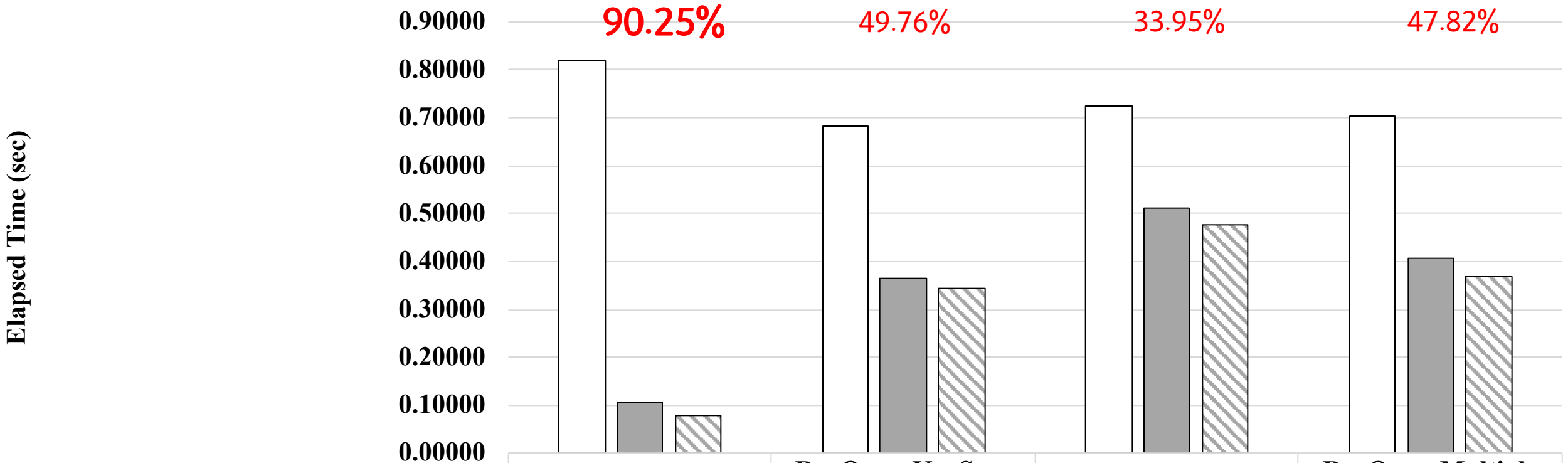
Multiple Programs having Combined Access Patterns

4. 실험 결과 - Internal Redundancy Elimination



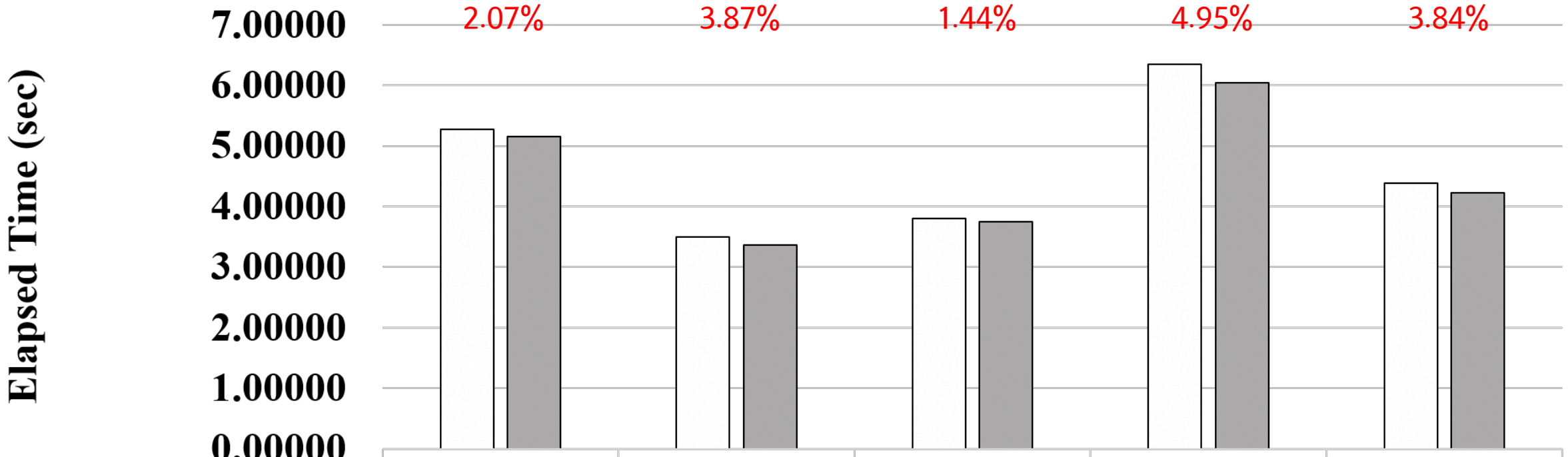
	RegQueryKey	RegCloseKey	RegOpenKey	RegQueryValue	RegEnumKey
Original	0.64985	0.55005	0.60083	0.83021	0.77533
Internal redundancy elimination with event-driven	0.21503	0.26930	0.28162	0.41642	0.16254
Internal redundancy elimination without event-driven	0.18891	0.00000	0.00000	0.39894	0.11855

4. 실험 결과 - Internal Redundancy Elimination



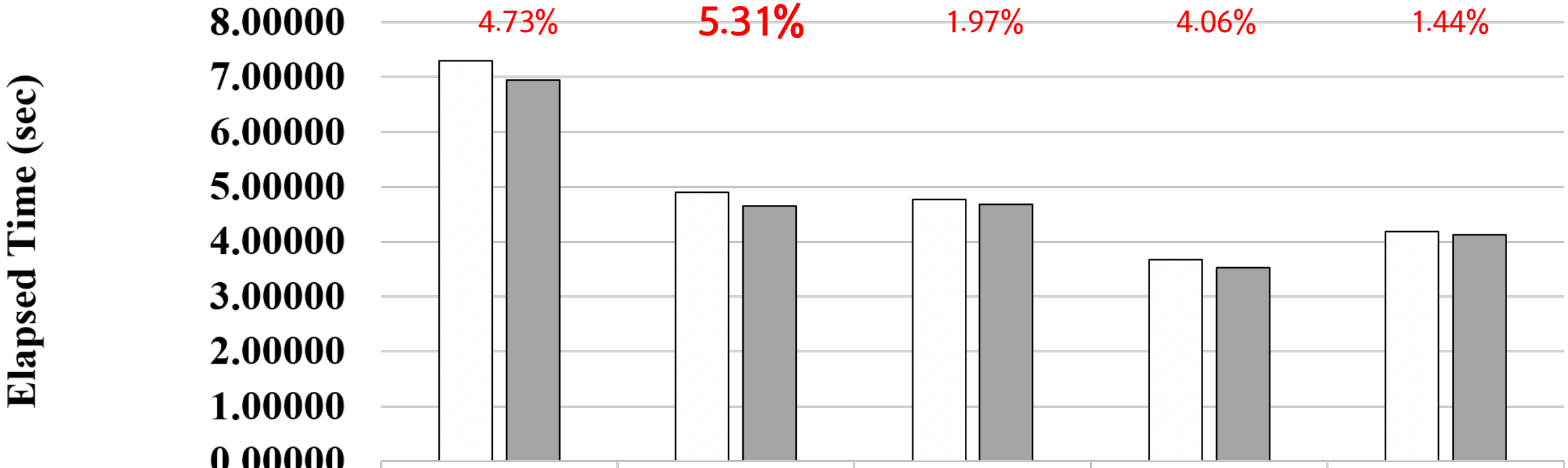
	RegEnumValue	RegQueryKeySecurity	RegLoadKey	RegQueryMultipleValueKey
Original	0.81791	0.68314	0.72340	0.70419
Internal redundancy elimination with event-driven	0.10626	0.36582	0.51106	0.40538
Internal redundancy elimination without event-driven	0.07972	0.34321	0.47781	0.36745

4. 실험 결과 - Outer Redundancy Elimination



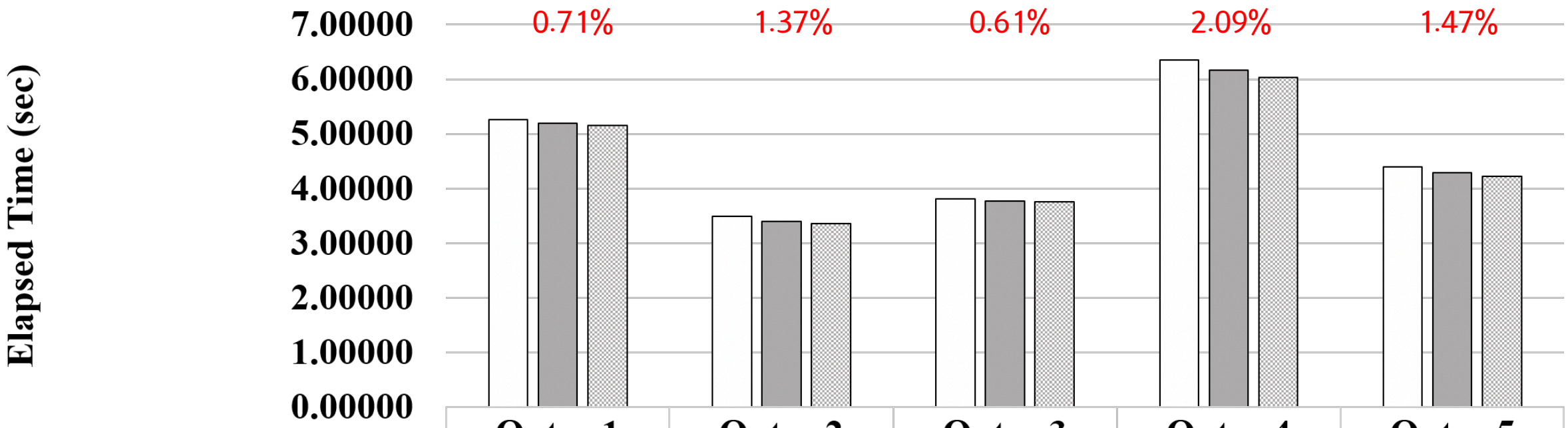
	Outer 1	Outer 2	Outer 3	Outer 4	Outer 5
□ Original	5.26669	3.49272	3.80702	6.35060	4.39017
■ Outer redundancy elimination	5.15763	3.35742	3.75210	6.03618	4.22162

4. 실험 결과 - Outer Redundancy Elimination



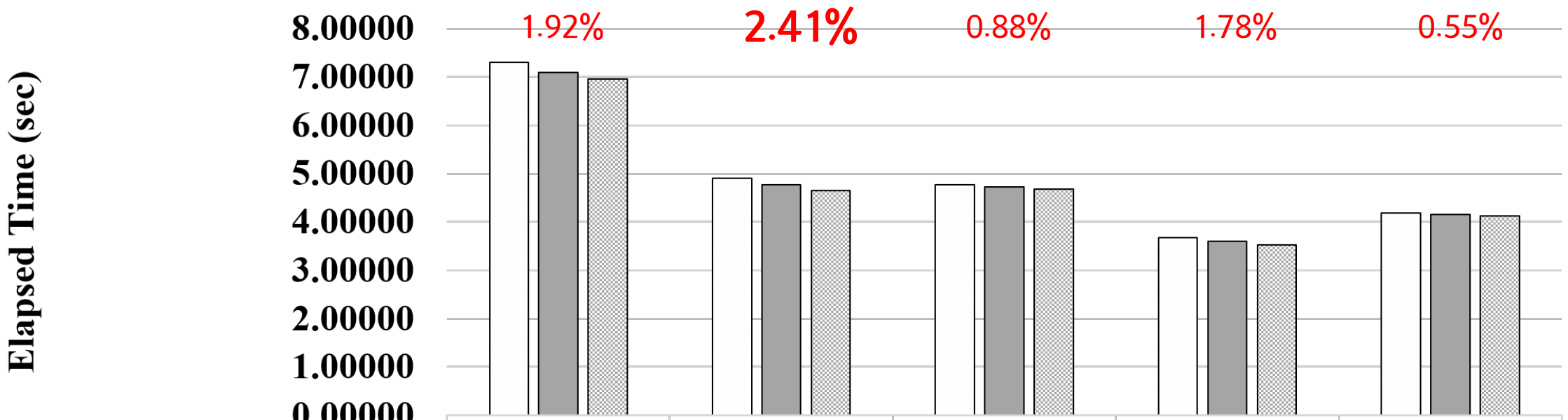
	Outer 6	Outer 7	Outer 8	Outer 9	Outer 10
□ Original	7.29953	4.90860	4.77734	3.67925	4.18946
■ Outer redundancy elimination	6.95449	4.64795	4.68333	3.52994	4.12932

4. 실험 결과 - Two-Level Method Elimination



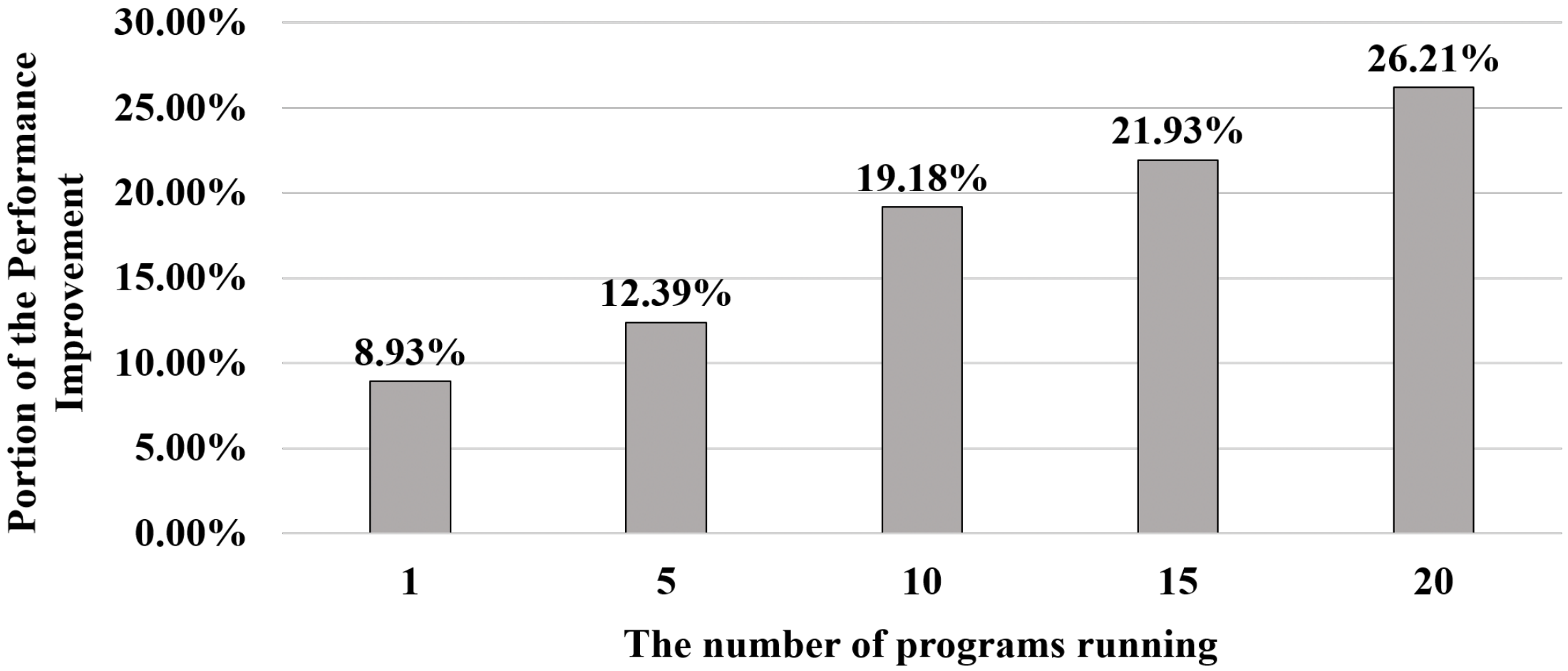
	Outer 1	Outer 2	Outer 3	Outer 4	Outer 5
Original	5.26669	3.49272	3.80702	6.35060	4.39017
Direct outer redundancy elimination	5.19468	3.40394	3.77526	6.16503	4.28455
Internal-then-outer redundancy elimination	5.15763	3.35742	3.75210	6.03618	4.22162

4. 실험 결과 - Two-Level Method Elimination



	Outer 6	Outer 7	Outer 8	Outer 9	Outer 10
Original	7.29953	4.90860	4.77734	3.67925	4.18946
Direct outer redundancy elimination	7.09059	4.76262	4.72488	3.59396	4.15225
Internal-then-outer redundancy elimination	6.95449	4.64795	4.68333	3.52994	4.12932

4. 실험 결과 - Multiple Programs Having Combined Access Patterns



References

- https://docs.microsoft.com/en-us/windows/win32/sysinfo/registry-functions
- https://searchenterprisedesktop.techtarget.com/definition/Windows-Registry-Editor
- https://docs.microsoft.com/en-us/sysinternals/downloads/procmon
- Vivek Gupta, Ethan Jackson, Shaz Qadeer and Sriram Rajamani. "P: Safe Asynchronous Event-Driven Programming". Retrieved 20 February 2017.
- https://www.hex-rays.com/products/ida/support/download_freeware.shtml

감사합니다