The Case of the Quadratic Killer

A Detective Tempus Stamp Mystery

A Victorian Tale of Murder, Graphs, and Algorithmic Justice

A Killer on the Cobblestones

The fog rolled thick through London… like poorly optimized code settling into production. A string of mysterious failures plagued the city. First, young Master Endpoint collapsed processing just twenty elements. Then Sir Reginald Response expired during a routine transformation. The latest victim: Lady Renderby, heiress to a processing fortune, silenced by a timeout.

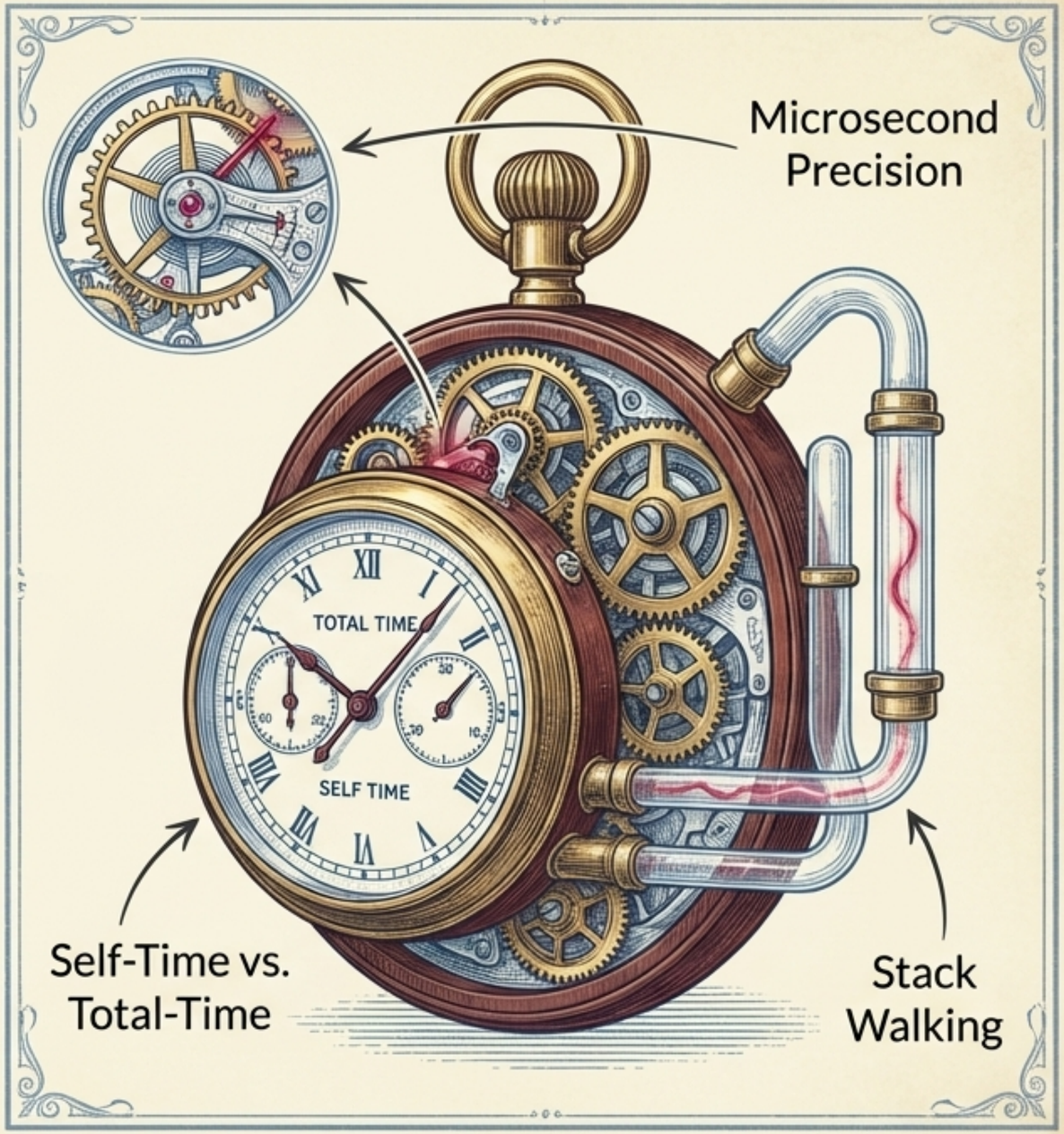
The gravest news: The Duchess of Documentsworth, 33 kilobytes of the finest HTML aristocracy, had timed out. Permanently.

Key Insight: A serial killer was at work. If the Duchess, the largest and most complex document, could be killed, no webpage was safe.

The Detective's Apparatus

The investigation centres on Attribute Manor, a sprawling estate with a curious three-node model architecture. The chaos inside suggests a deep-seated problem. To find the killer, Detective Stamp requires a revolutionary device: **The Timestamp Decorator.**

"Total time can be deceiving... Self-time reveals the truth—where the actual computation occurs." – Detective Tempus Stamp



Forensic Report: A Singular Suspect

The Timestamp Decorator was deployed across Attribute Manor. The initial results were illuminating.

Operation	Calls	Total Time (ms)	Self Time (ms)	% of Total
Register Attributes	**17**	**183.3**	**183.1**	**52.9%**
Body Setup	1	101.2	3.8	1.1%
Sir Convert From-Dict	1	345.8	1.5	0.4%
Styles Setup	1	4.6	4.6	1.3%
Head Setup	1	3.2	3.2	0.9%
... other operations	...	...	...	...
TOTAL		**345.9**		

Over half the total time in just 17 calls! Nearly 11ms per registration. This is where the killer hides. We drill deeper.

Following the Scent of Self-Time

With `Register Attributes` identified as the scene of the crime, Detective Stamp deployed more granular measurements within the operation to pinpoint the exact cause.

Operation within Register Attributes	Calls	Total Time (ms)	Self Time (ms)
add_attributes	**68**	**129.2**	**129.2**
registration logic	17	183.1	53.9

*The adding of attributes is **seven times more** expensive than the registration itself. But why?*

The Killer's Signature

The final, most granular measurements inside `add_attributes` reveal the terrible truth.

Operation within add_attributes	Calls	Total Time (ms)	Self Time (ms)	Per Call (ms)
get_or_create_value	**68**	**61.8**	**61.8**	**~9.1**
get_or_create_name	**68**	**51.2**	**51.2**	**~7.5**
other logic	68	129.2	16.2	~0.2

*They're just lookups! They should be instantaneous!

*The time grows with each subsequent call. The pattern is unmistakable...
These operations are scanning every node in the graph. **Every. Single. Time.**

The Name of the Killer is Quadratic

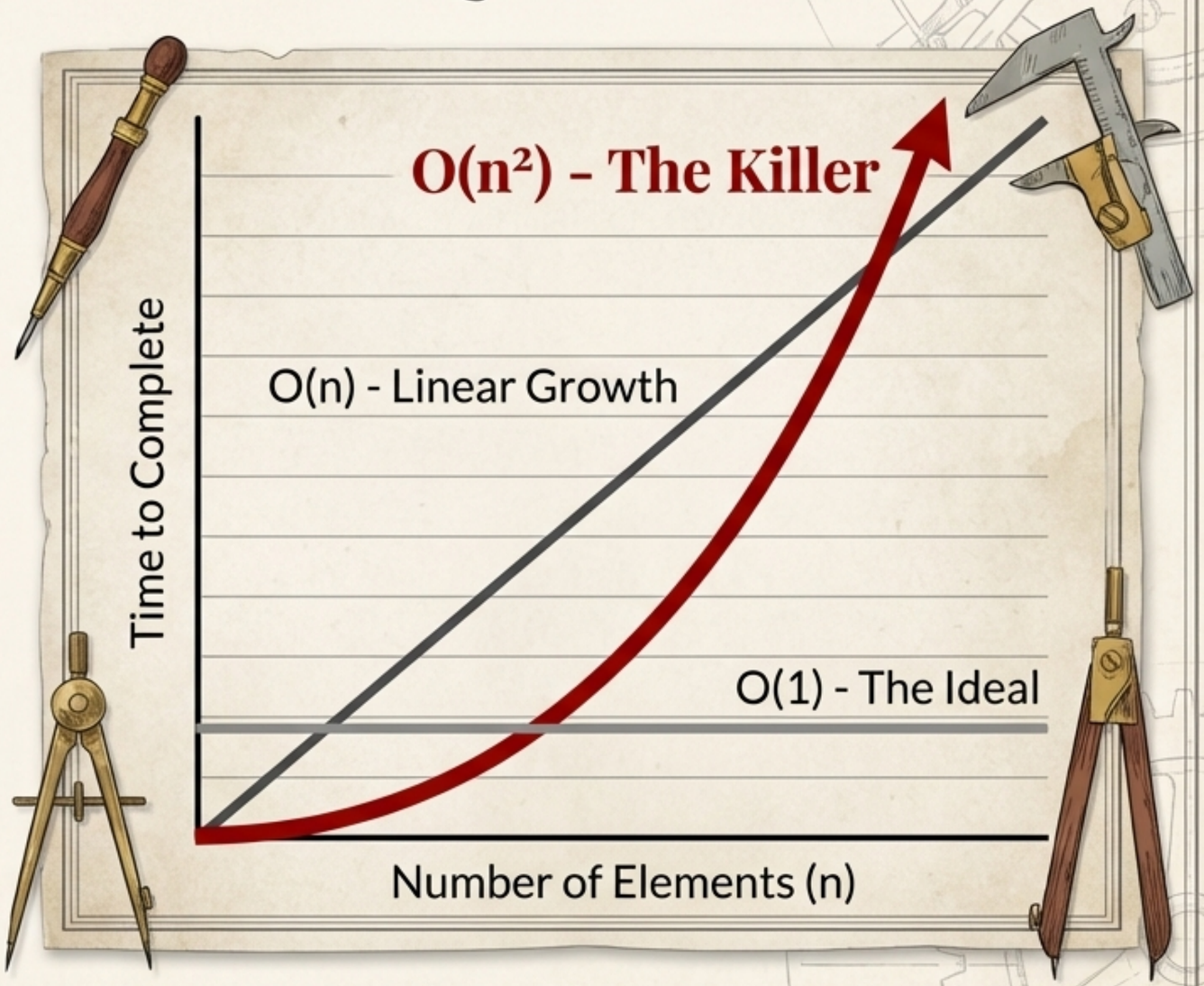
The Confession

As the graph of nodes grows, each individual lookup takes longer. When you perform thousands of lookups, the total time explodes.

The Mathematics of Murder

The complexity is not linear, O(n). It is quadratic, $O(n^2)$.

"$O(n^2)$. The signature of Baron von Quadratic." - Detective Tempus Stamp



An Instrument of Cruelty

The Baron's method was simple, readable, and utterly deadly at scale. It hides in plain sight, a choice of present convenience over future performance.

```
# Find a node by its value
def get_node_by_value(self, value):
    for node_id in self.nodes_ids():
        if self.get_node_value(node_id) == value:
            return node_id
    return None
```

CLASSIFIED EVIDENCE

Monstrous. It scans every node in existence just to find a single value. Repeatedly.

Nodes in Graph	Time per Lookup
10	1 ms
50	5 ms
100	10 ms
1000	**100 ms**

The Path to Justice: The Hash Index

The **Strategy**: Instead of scanning every node for every lookup (O(n)), we can maintain a simple dictionary mapping values to node identifiers.
The Result: Lookups become **O(1)**—constant time, regardless of graph size. **The cost is paid once during creation**, not thousands of times during lookup.

The Murder Weapon (O(n²))

```
def get_node_by_value(self, value):
  for node_id in self.nodes_ids():
     if self.get_node_value(node_id) == value:
         return node_id
  return None
```

The Hero (O(1))

```
# At creation:
self.value_to_node_id = {
    self.get_node_value(n_id): n_id
    for n_id in self.nodes_ids()
}
# The lookup:
return self.value_to_node_id.get(value)
```

"Four lines of code to undo everything you've built. Four lines to save countless future documents from your quadratic cruelty."

The Duchess is Resurrected

CONFIDENTIAL

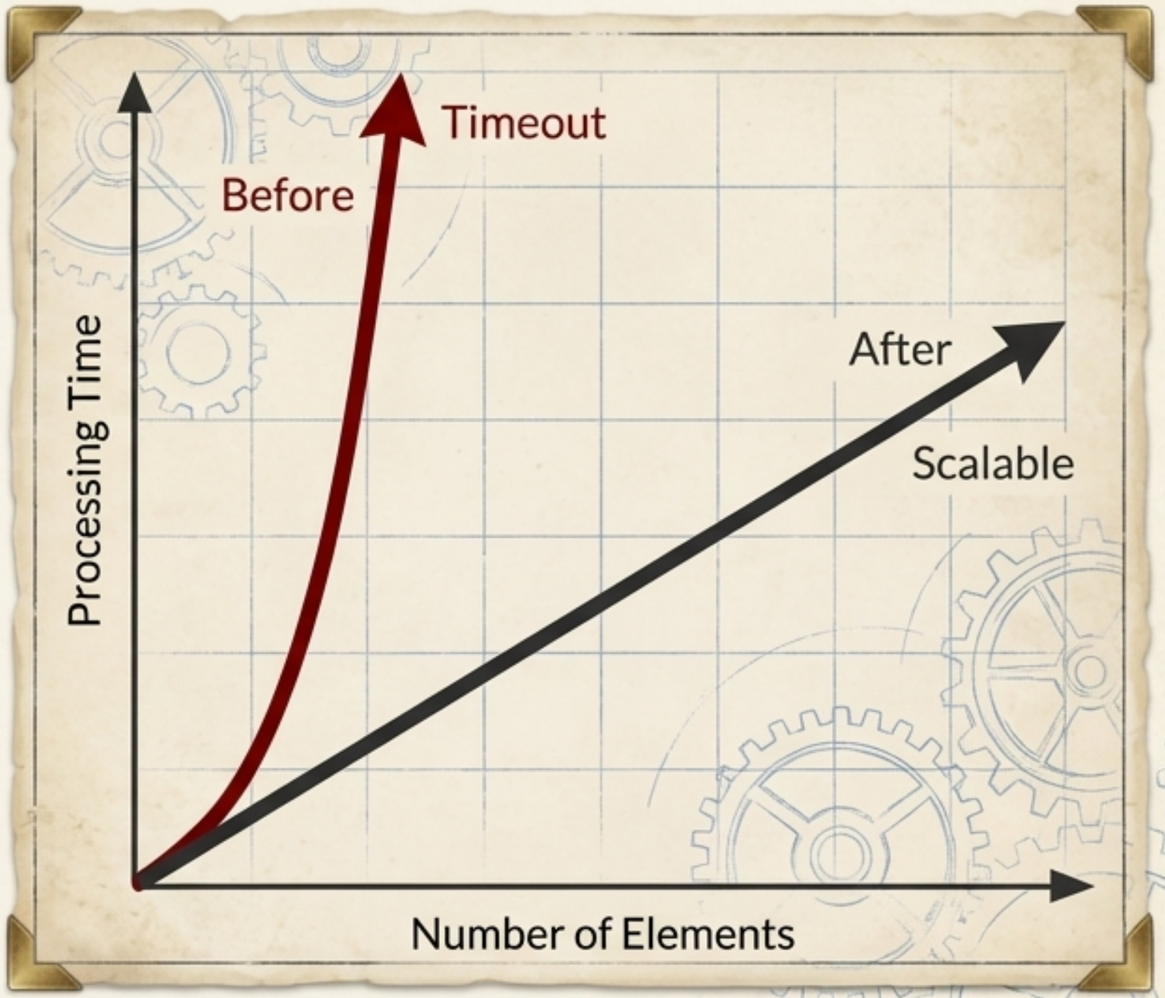
The Test: The Duchess of Documentsworth (33kb, >1000 elements) was processed again with the Hash Index in place.

The Results

Metric	Before Fix (The Baron's Method)	After Fix (The Hash Index)
Duchess (500 elements)	Timeout (Request > 30s)	**10.9 seconds**
Full Test Suite Completion	~2m 15s	**~1m 45s (22% Faster)**
Algorithmic Complexity	$O(n^2)$ - Unscalable	**O(n) - Linearly Scalable**

She lives. More importantly, she scales. The work per element now remains constant.

"Justice served. The quadratic monster is slain."



The Detective's Closing Statement

1. **Performance crimes** hide in **innocent-looking code.** A simple loop, a straightforward scan. Readable, intuitive, and deadly at scale.

2. **Vigilance and Measurement are your best defence.** Never trust an algorithm's complexity by its appearance. Always profile. Always test at scale.

3. **Focus on Self-Time.** It cannot lie. It points directly to where computational work is actually being done.

The truth is always in the timing.

"Self-time never lies."

Dramatis Personae

The Character	The Technical Reality
Baron von Quadratic	$O(n^2)$ - Quadratic Complexity
The Loop	Inefficient `for` loop scanning all nodes
The Hash Index	O(1) Dictionary / Hash Map Lookup
Detective Tempus Stamp	The Performance Engineer (You)
The Timestamp Decorator	Profiling Tool
Self-Time	Time in method (excluding children)
Timeout / Death	Request Exceeding Time Limit

*Author's Warning: *Baron von Quadratic is believed to be at large in legacy codebases worldwide. Remember:* ***Friends don't let friends*** *use linear scans for lookups."*

Final Thematic Quote: "In the grand graph of life, we are all but nodes seeking our edges. Let us ensure those connections are indexed properly." – D.T.S., 1891