Dot Net Perls

Search Dot Net Perls Search

Firefox 3 Memory Benchmarks and Comparison

by Sam Allen

Web browser performance is an often talked-about and flaunted thing, but many claims are not really backed up by solid evidence. I wrote software that collected **millions of data points** over 14 hours of actual browsing time, and this article reveals my findings.

Problem

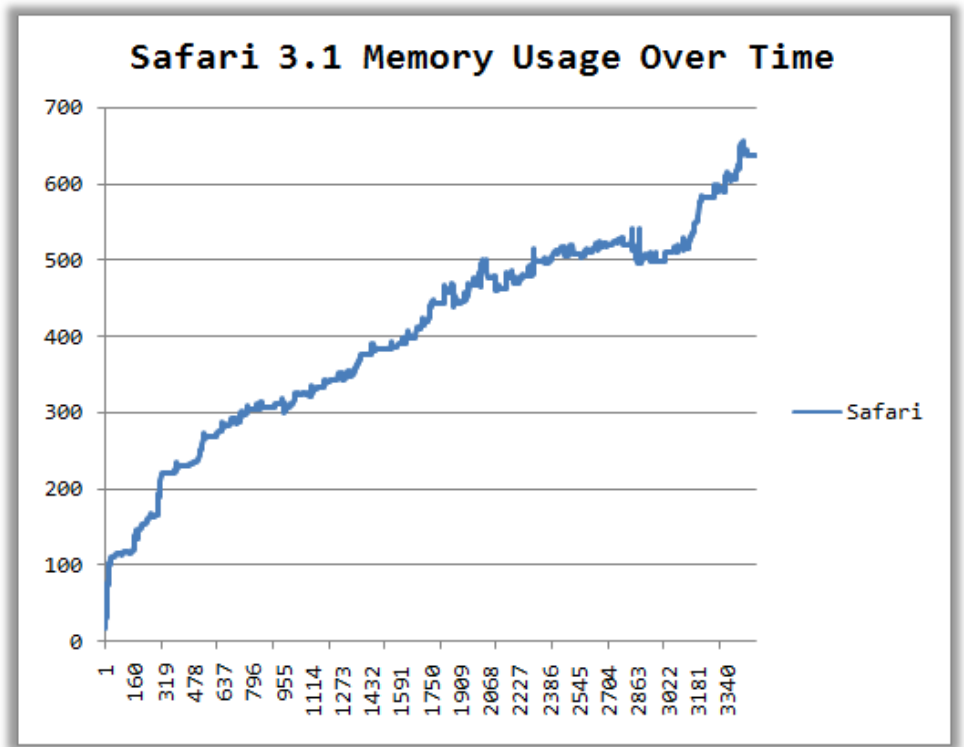
Many people load **hundreds of web pages**, sometimes at the same time, often over periods of 3+ hours. Users complain about the memory usage of Firefox, Safari, or Internet Explorer, and we need a way to identify which browsers are better at managing memory than others. Traditional benchmarks do not look at all the things you might do with a program, and we need real-world numbers over a period of hours.

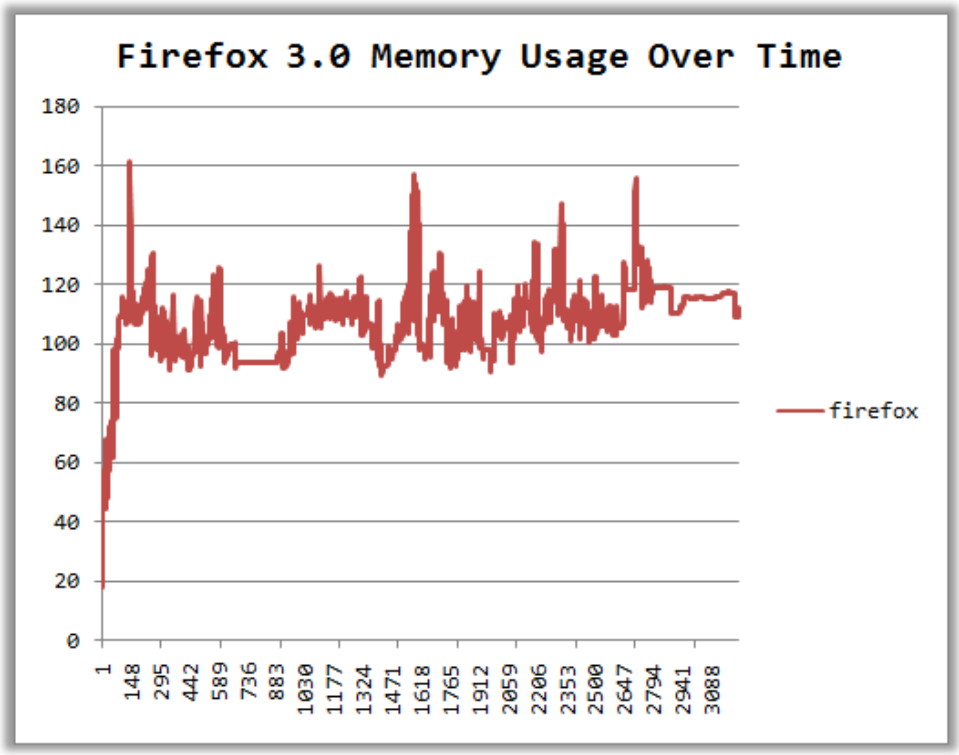
Solution

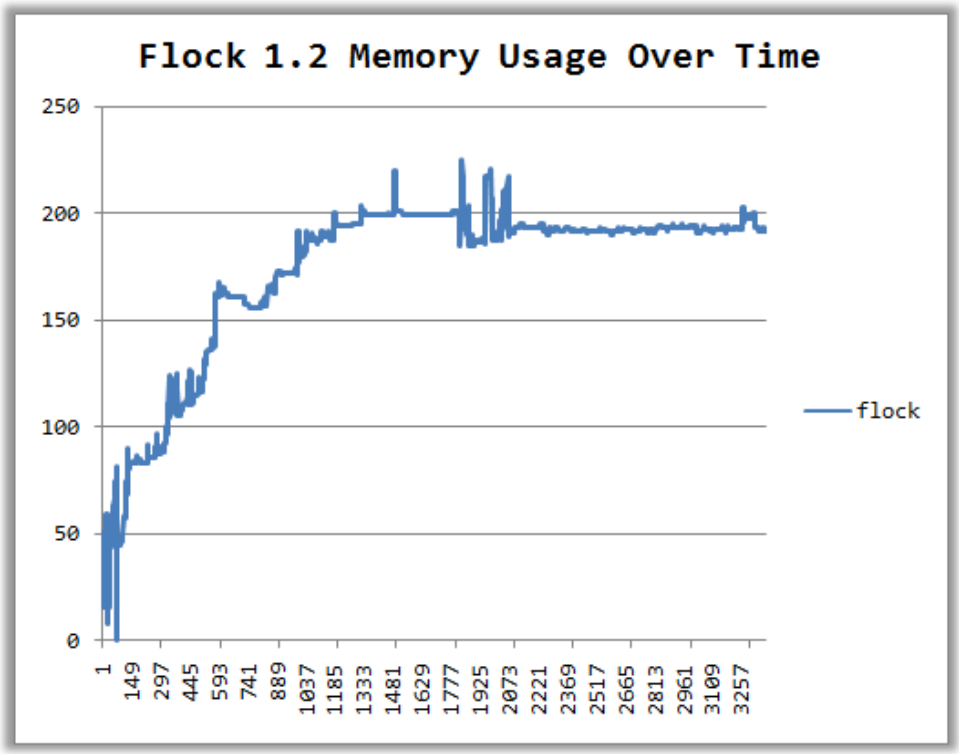
I developed a Windows Forms application in .NET called Memory Watcher that "watches" the system memory numbers. It uses a timer to poll the processes **every 3 seconds**. It then records every number and also prints them out in a grid on the screen. This allows us to keep track of each program's memory usage over time and with real-world usage.

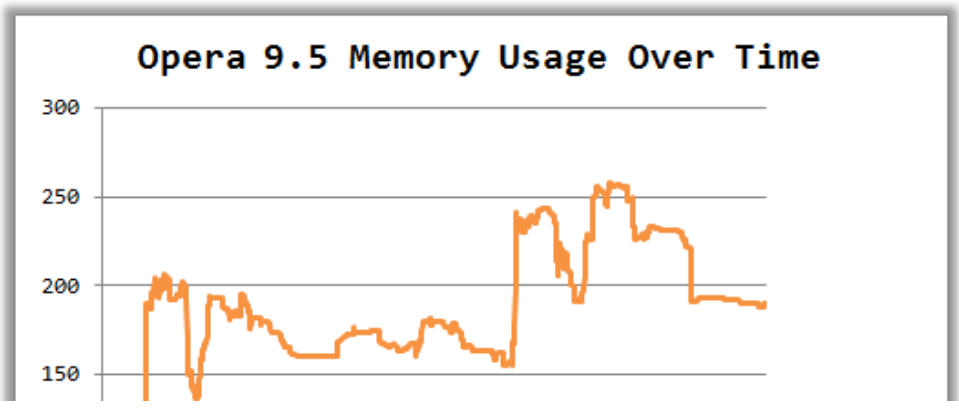
Memory Profiles

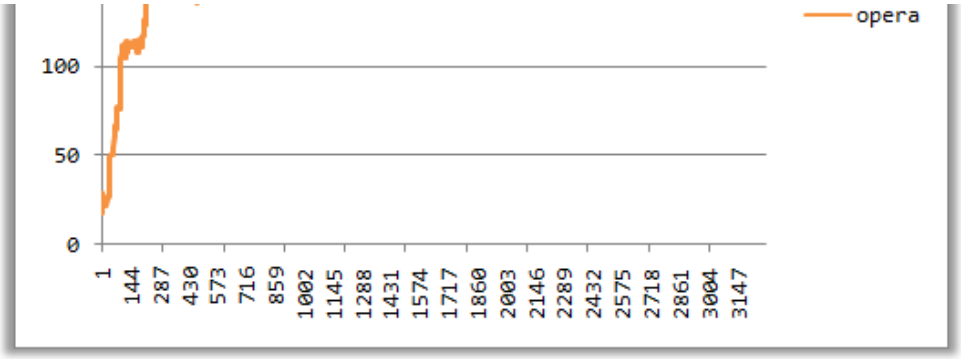
These results are from opening Memory Watcher and then using the browser between 9,000 and 11,000 seconds (close to 3 hours). Each browser is tested in a **separate session**, and there are brief periods of inactivity throughout the time period. The vertical axis is the memory used in MB, and the horizontal axis contains the memory "checkpoints" my program took (one every 3 seconds).

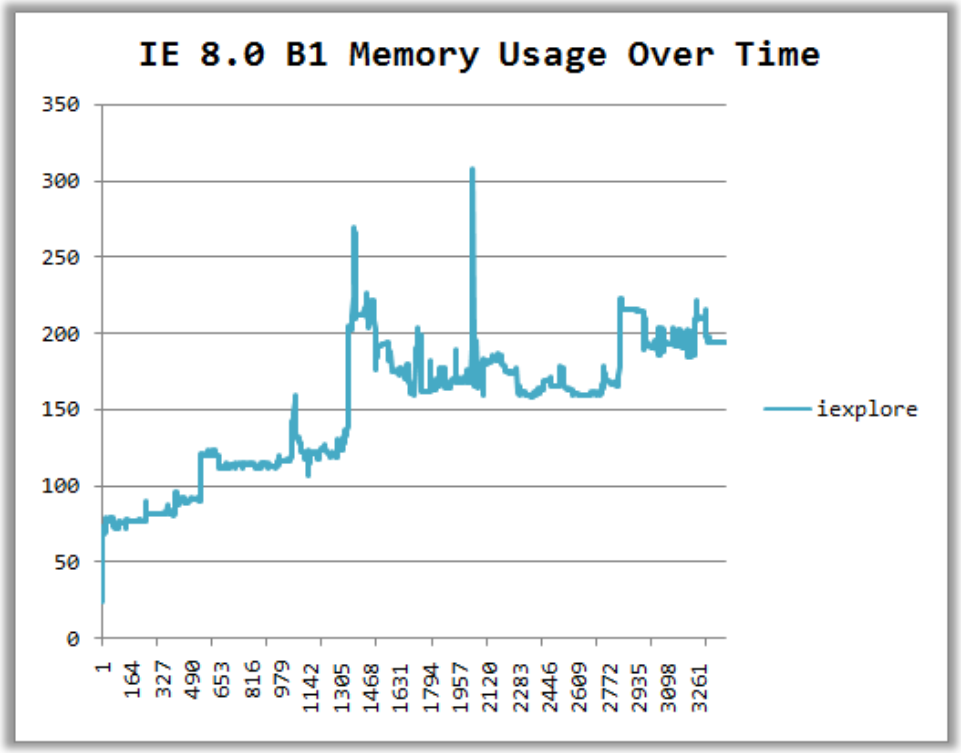



Firefox 3.0 Memory Usage Over Time
180
160
140
120
100
80
60
40
20
0
1
148
295
442
589
736
883
1030
1177
1324
1471
1618
1765
1912
2059
2206
2353
2500
2647
2794
2941
3088
firefox


Flock 1.2 Memory Usage Over Time
250
200
150
100
50
0
1
149
297
445
593
741
889
1037
1185
1333
1481
1629
1777
1925
2073
2221
2369
2517
2665
2813
2961
3109
3257
flock


Opera 9.5 Memory Usage Over Time
300
250
200
150





Benchmark Details

The above profiles are not a direct comparison in any way, but they offer a visualization of *trending* in the memory behavior of the layout engines and interfaces. This is not a diagnosis or bug report. Let me show some important metrics of the above results.

Browser name	Exact version	Time active (s) Hours	Comments
Safari	3.1.2	10,470 s 2.91 hours	Normal browsing
Firefox	3.0	9,681 s 2.69 hours	Normal browsing No extensions
Flock	1.2.2	10,146 s 2.82 hours	Flock is based on Firefox 2.0 No extensions other than the default
Opera	9.5	9,855 s 2.74 hours	No extensions Only browser was used
IE	8.0	10,236 s 2.84 hours	Used 7.0 rendering mode No extensions

The system is Windows Vista SP1, and the computer has 3.0+ GB of RAM. No plugins are disabled, but the Acrobat Reader and Java plugins were (presumably) not used. Flock is based on Firefox 2.0 but its memory usage is probably worse because it uses built-in extensions.

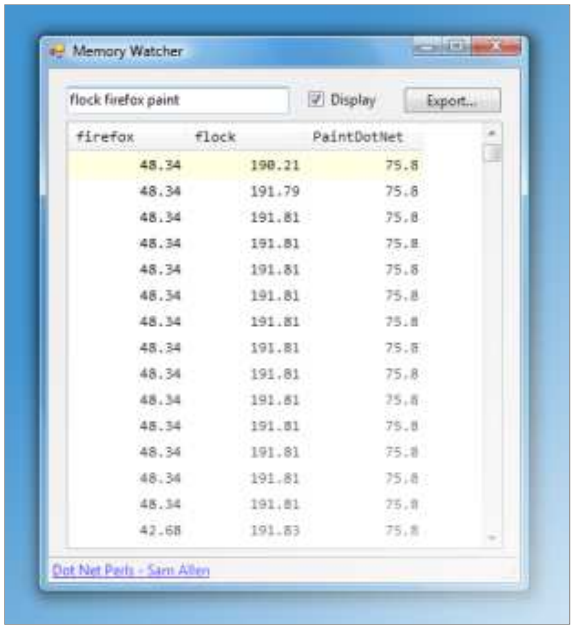
- **Just regular stuff**
  These aren't stress tests, and I probably never went over 4 windows in each browser, with at most 3 tabs in each window. I didn't look at many pages that are extremely heavy on images, and no "browser benchmark" style pages. Gmail was used on each browser.
- **Not just pages**
  It is hard for a regular benchmark to "simulate" a user actually clicking on things. Interactions with the user can greatly influence memory or performance. Having a responsive browser is probably more important than just having a "fast" one at showing pages.
- **Plugins included**
  My profiles include Flash and possibly other plugins. A browser might have memory issues with a plugin and that could cause a significant problem with the user experience. (Most Windows Vista crashes have been due to graphics cards, not Vista itself, for example.)
- **Real-life usage**
  An automation script will never give the same insight into performance over time as will this sort of profile. As developers, we want to make programs that work well for our users, and not just for tests. The tests capture the "rhythm" of software usage.

Final Memory Measurements

The data in this article are those reported by Windows Vista, but the individual numbers **should not be compared** to each other. Some browsers were tested slightly longer than others, and some different pages were loaded. That said, here are the final performance metrics.

Browser name	Ending private set in MB
Safari	636.9
Firefox 3	111.8
Flock (Firefox 2)	191.9
Opera 9.5	190.6
Internet Explorer	194.4

About the "Memory Watcher"



Memory Watcher is a small program I wrote that records the memory usage of each process on the system every three seconds. It uses the *PrivateMemorySize64* long value from the *Process* collection in .NET.

- **Simple**
  There are tools similar to this, offered on every platform, but they are not usually easy to use. Memory Watcher provides a super-easy way to monitor every process and silently work in the background.
- **Exports to spreadsheet**
  It exports the currently viewed data to a CSV file. These data are easily taken into Excel, and were used for the graphs in this article.
- **Implementation notes**
  The application uses a *DataGridView* control, and sets its *DataSource* property to a DataTable which is built from the object collection. It uses a Timer to poll the system every 3 seconds. It offers searching and filtering of processes using a TextBox.

Conclusion

These profiles are meant to provide a picture of what the memory behavior of popular browsers is over a period of time, not to provide absolute benchmark times. Firefox 3.0 shows memory usage that is significantly lower than Firefox 2, which also does very well. Here is a summary of my results.

- **Safari 3.1**
  Safari on Windows shows extremely poor memory management, and I do not know whether it ever reaches a high water mark. If this is by design, it is certainly a design that looks inefficient and seems to contradict Apple's marketing.
- **Firefox 3.0**
  This browser exhibits memory usage that is by far lower than the others. It releases memory to the system and the trend line is nearly flat.
  (This is likely due to the efforts outlined here.)
- **Flock (based on Firefox 2.0)**
  Flock did very well and this browser and Firefox 2.0 could likely be run for long periods without causing many problems. The extensions probably reduced the efficiency somewhat.
- **Opera 9.5**
  Opera's performance was about as good as Firefox 2.0 (Flock), and it could likely be used for very lengthy sessions. However, *Kestrel* is certainly not a revolutionary or even notable technology in this arena.
- **Internet Explorer 8 Beta 1**
  IE did well in the profile, although a worrying trend in the data could indicate that it would keep escalating. However, this browser could likely sustain many hours of moderate usage.

Final Thoughts

After browsing for 14 hours with these programs, and recording all the results into spreadsheets, the **most memory efficient** browser in my usage is very clear--**Firefox 3.0** not only trumps its older version, but every other popular offering on Windows. This article may help other vendors rethink their marketing campaigns, and may prompt further improvements.

Dot Net Perls is a website dedicated to sharing code and knowledge. It has

- 105 featured articles focusing on C# and .NET programming
- 7 *Tools* articles similar to this one
- A companion source code archive

© 2007-2008 Sam Allen. All rights reserved.