Different Dynamic Options for Integrating SVG

Robert Gezelter, Principal

Robert Gezelter Software Consultant

http://www.rlgsc.com

SVG – Files can be lengthy

```
<svg xmlns="http://www.w3.org/2000/svg">
  <g id="circle1">
    <circle cx="999" cy="100" r="100" style="…"/>
    …
    </g>
  …
  </svg>
```

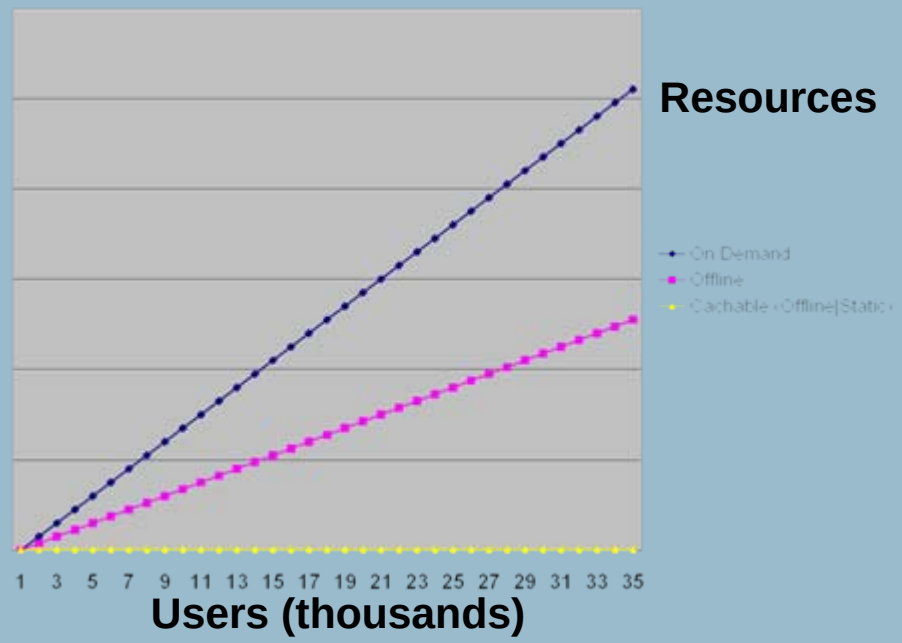
Delivering SVG images

Large image descriptions:

- Dynamic generation requires server-side resources
- Transmission requires significant bandwidth
- Successive related images require regeneration and retransmission

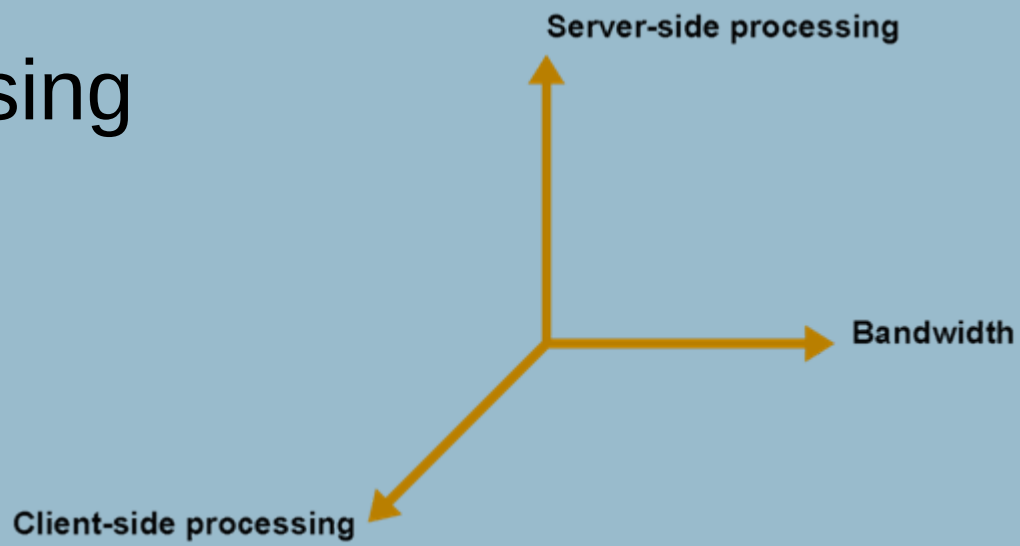
Fact: Dynamic HTML is expensive

- Non-cachable
- Expensive to generate
- Resources are finite



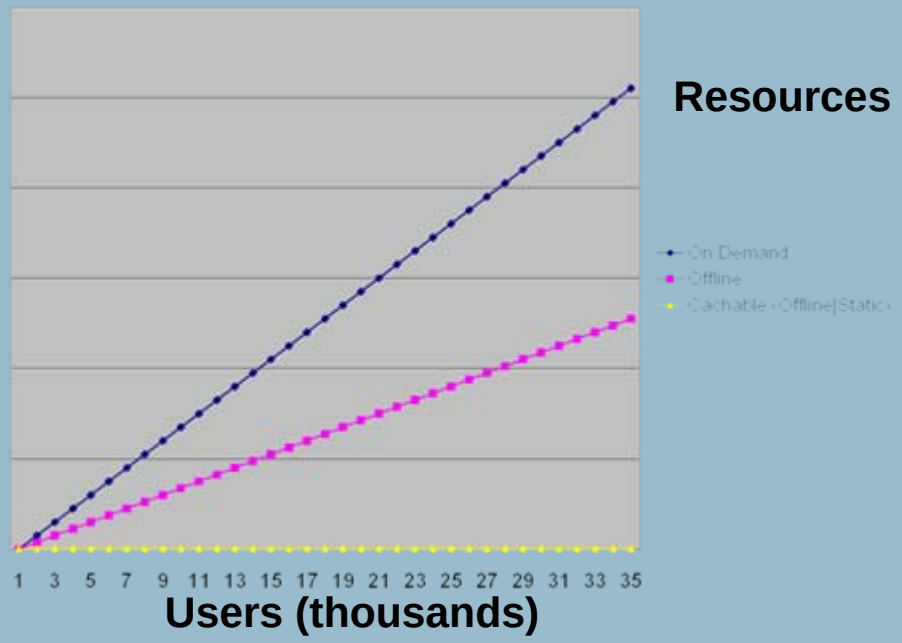
Multi-axis optimization

- Minimize overall resources
- Client-side resource scale
- Minimize processing



On demand generation:

- Cannot optimize performance
- Is not scalable
- There is a better way

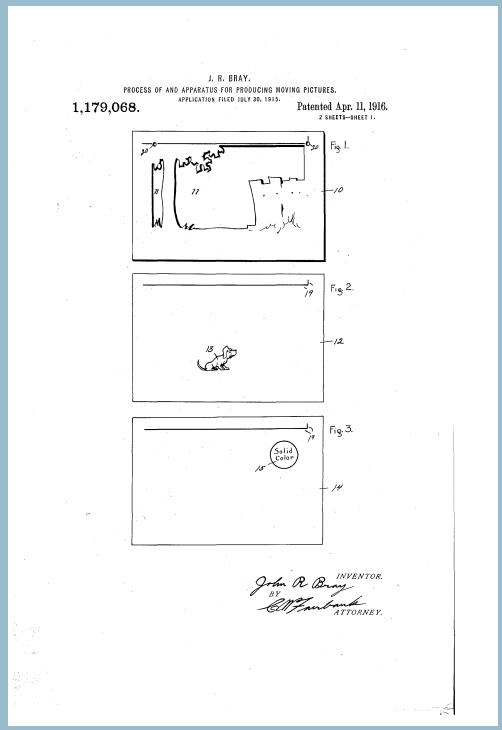


An answer: Multi-stage compositing

- Static HTML/SVG
- Event-driven non-realtime regeneration (XSLT and other tools)
- AJAX/JavaScript
- Client-side (JavaScript) annotation

Problem not without precedent

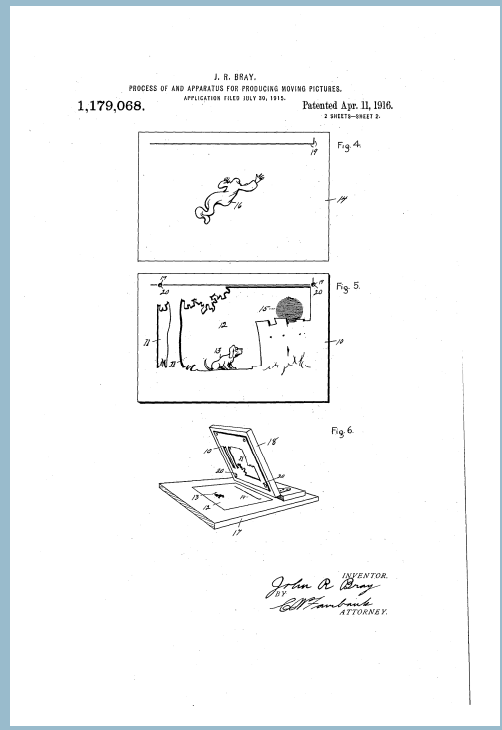
Classic cinematography patents:



J. R. BRAY.
PROCESS OF AND APPARATUS FOR PRODUCING MOVING PICTURES.
APPLICATION FILED JULY 30, 1915.
1,179,068. Patented Apr. 11, 1916.
2 SHEETS—SHEET 1.

Fig. 1.
Fig. 2.
Fig. 3.
Solid Color

INVENTOR.
ATTORNEY.



J. R. BRAY.
PROCESS OF AND APPARATUS FOR PRODUCING MOVING PICTURES.
APPLICATION FILED JULY 30, 1915.
1,179,068. Patented Apr. 11, 1916.
2 SHEETS—SHEET 2.

Fig. 4.
Fig. 5.
Fig. 6.

INVENTOR.
ATTORNEY.

"Process and Apparatus for Producing Motion Pictures", US Pat 1,179,068 (filed July 30, 1915)

Hierarchy

- Static
- Varying long scale (out of response loop)
- AJAX components
- Client-side annotation/markup (the “client cloud”)
- Dynamic server-side technologies (e.g., ASP, PHP, EJB)

Scarce resource efficiency

- Minimize realtime server response loop
- Reduce growth of server-side
- Reduce vulnerability to bandwidth issues
- Minimization particularly attractive in mobile context

Different Object Types

- Manually created
- Programmatically created triggered by external events
- On-Demand generated elements
- Client-side created elements (e.g., annotations)

IDs act as “registration”

- **`getElementById`** to identify element
- Dynamically insert subtrees retrieved using **`XMLHttpRequest`**
- Dynamically generate elements using JavaScript from **`XMLHttpRequest`** retrieved data (client-side computation)
- Dynamically generate elements using JavaScript based upon user input

Classic execution-time optimization

- This approach has a rich history in other aspects of computing
- Compiling/interpreting
- “reduction in strength”
- “include files”, object libraries
- Dynamic linking using shareable libraries

Summary

- The resources used to render a particular page varies dramatically depending upon implementation strategy and technique
- Properly used, both offline element generation and “client-cloud” processing have significant bottom-line impact.

Questions

Supplemental Materials, Slides:

http://www.rlgsc.com/svgopen/2011/options-for-dynamic-svg.html