Resonse time model

Dave Abercrombie, dabercrombie@convio.com
see http://aberdave.blogspot.com
March 2011

Import and plot tab-delimited observational data. It has two columns: workload (buffer gets per millisecond) and response time (milliseconds per buffer get)

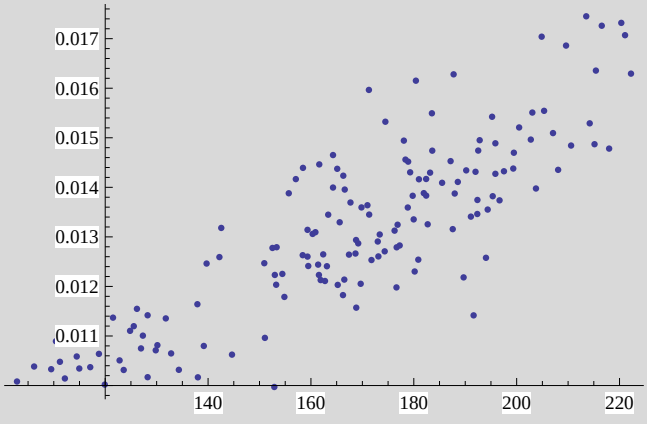
In[1]:=
```
Directory[]
```

Out[1]=
```
C:\Users\abe\Documents
```

In[2]:=
```
responseTimeObservations =
  Import["Mathematica\\response-time\\db9-response-time-model-02.txt", "Table"];
```

In[3]:=
```
p1 = ListPlot[responseTimeObservations]
```

Out[3]=



Define response time formulas. From OraPub's 2010 Performance Seminar, August 18 2010, NoCOUG Training day, Pleasanton CA, printed page 51 of 78, "Computing System Behavior Analysis" slide pg: 11ve1.

In[4]:=
```
utilization = (serviceTime * workload) / numEffectiveServers
```

Out[4]=

$$\frac{\text{serviceTime workload}}{\text{numEffectiveServers}}$$

In[5]:=
```
responseTime = serviceTime / (1 - (utilization)^numEffectiveServers)
```

Out[5]=
$$\frac{\text{serviceTime}}{1-\left(\frac{\text{serviceTime}\,\text{workload}}{\text{numEffectiveServers}}\right)^{\text{numEffectiveServers}}}$$

The observed service time was a near constant 0.0093 milliseconds per buffer get, with only a 5% standard deviation. I use this constant service time as an initial simplification.

In[6]:=
```
observedServiceTime = {serviceTime → 0.0093};
responseTimeSimplified = responseTime /. observedServiceTime
```

Out[7]=
$$\frac{0.0093}{1-0.0093^{\text{numEffectiveServers}}\left(\frac{\text{workload}}{\text{numEffectiveServers}}\right)^{\text{numEffectiveServers}}}$$

Do a non-linear least-square fitting of the observational data to the simplified response time curve. Ignore solutions with fewer than two servers.

In[8]:=
```
bestFitNumEffectiveServers = FindFit[responseTimeObservations,
  {responseTimeSimplified, numEffectiveServers > 2}, numEffectiveServers, workload]
```

Out[8]=
$$\{\text{numEffectiveServers} \to 2.66658\}$$

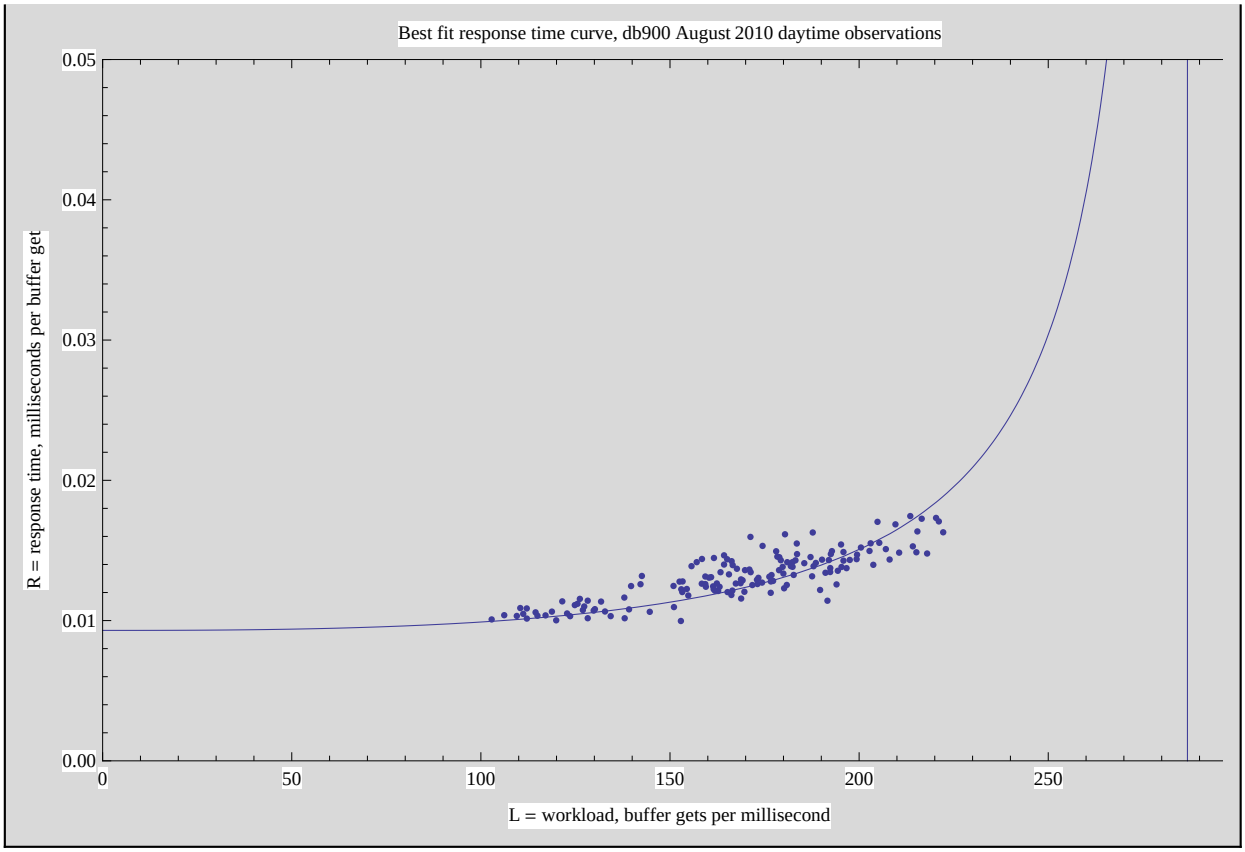
In[9]:=
```
bestFit = responseTimeSimplified /. bestFitNumEffectiveServers
```

Out[9]=
$$\frac{0.0093}{1-2.79865\times 10^{-7}\,\text{workload}^{2.66658}}$$

Compare the observational data to the best fit formula

In[10]:=
```
p2 = Plot[ bestFit, {workload, 0, 300}, PlotRange → {{0, 300}, {0, 0.050}},
    Frame → True,
    FrameLabel → {"L = workload, buffer gets per millisecond",
      "R = response time, milliseconds per buffer get",
      "Best fit response time curve, db900 August 2010 daytime observations"},
    ImageSize → 288 * 2
   ];
Show[p2, p1]
```

Out[11]=



Compare to 1) doubling the count of identical CPUs, and 2) doubling the speed of the CPUs without increasing their count. [TODO: the 2.66 is hard-coded, and the 0.0093 is copied - these should be dynamic and/or defined...]

In[12]:=
```
serviceTimeDoubleFast  = {serviceTime → 0.0093 / 2}
numEffectiveServersDoubled  = {numEffectiveServers → 2.66658 * 2}
```

Out[12]=
```
{serviceTime → 0.00465}
```

Out[13]=
```
{numEffectiveServers → 5.33316}
```

In[14]:=
```
fasterCPUs = responseTime /. serviceTimeDoubleFast /. bestFitNumEffectiveServers
moreCPUs = responseTime /. observedServiceTime /. numEffectiveServersDoubled
```

Out[14]=
$$\frac{0.00465}{1 - 4.40785 \times 10^{-8}\ \text{workload}^{2.66658}}$$

Out[15]=
$$\frac{0.0093}{1 - 1.94298 \times 10^{-15}\ \text{workload}^{5.33316}}$$

At this scale, the observational data (blue) is already bumping up against the knee in the curve. If we double the speed of the CPUs, we'd expect response time to be cut in about half. If we double the number of CPUs, we'd expect response time to stay about the same, but we could get to over 400 buffer gets per millisecond before hitting the knee again. The knee in the curve is about the same for both CPU upgrade options.

In[16]:=
```
p3 = Plot[{bestFit, fasterCPUs, moreCPUs}, {workload, 0, 600},
    Frame → True,
    FrameLabel → {"L = workload, buffer gets per millisecond",
      "R = response time, milliseconds per buffer get",
      "Compare observed db9 (blue) to predictions of faster
        CPUs (red) or more CPUs (yellow)"},
    PlotRange → {{0, 600}, {0, 0.04}},
    ImageSize → 288 * 2
  ];
Show[p3, p1]
```

Out[17]=