

v0.01– addressed Peter's comments -- 10may06

Gelato Federation Scalability in a Box [SiaB] Focus Group April 2006

Summary:

The Gelato Scalability Focus Group hosted a 4 session track on Wednesday, 26Apr06, including 3 presentations and a wrap-up/discussion session. This track provides a forum for Gelato Federation members who work together to understand and improve Linux scalability to present their work to each other and to the wider federation membership; and to discuss areas that still need work to drive future efforts. As the name of the group implies, we focus our efforts on scaling of a single Linux image running on large scale symmetric and NUMA multiprocessor platforms, as opposed to grid and cluster computing. The latter is well covered in other Gelato tracks. Note that our interest extends to scalability in terms of high processor/core counts, large memory sizes, large amounts of attached storage—both in terms of capacity and number of paths/adapters—and network connectivity.

The presentations [covered in more detail below—DO check them out!] were:

Blktrace: An Overview	Alan Brunelle, HP
Scaling Linux to 512 Processors and Beyond	John Hawkes, SGI
Improving NFS Performance [work-in-progress]	Peter Chubb, UNSW

The first and third presentations cover some tools that we are using to investigate and improve Linux scalability in the areas of locally attached and NFS accessed storage. John Hawkes' presentation provides an excellent overview of the concepts of OS scalability and the work that has been done and still needs to be done in Linux to make it scale to impressively large systems.

As for the wrap-up session: let's just say that we [I] need to work on this. You can read more below.

The Presentations:

1. Blktrace: An Overview

The presentation slides may be found at:

http://www.gelato.org/pdf/apr2006/gelato_ICE06apr_blktrace_brunelle_hp.pdf

This presentation could just as easily have been included in the “Tools and Tuning” track. However, Alan works in the HP OSLO Scalability and Performance Team, and his involvement with blktrace arose because of his work in characterizing and improving the performance and scalability of the Linux storage subsystem. Alan co-developed the blktrace tool with Jens Axboe of SuSE [upstream maintainer of the Linux block IO subsystem] and others.

Blktrace has been accepted in the upstream Linux kernel as of 2.6.17 and, ultimately, should be available in the commercial distros. The developers have worked to make blktrace of sufficiently low overhead that the distros can leave it configured in production kernels, so that it is available to investigate storage performance anomalies at any time in

v0.01– addressed Peter's comments -- 10may06

production use. Even when enabled, blktrace incurred less than 2% application performance impact on a “fairly stressful IO situation.”

Alan discussed some of the techniques—such as per cpu buffers and data collection threads—that allow blktrace to scale to quite large [in terms of cpus and storage adapters/devices] systems. Noting that blktrace is NOT an analysis tool, Alan went on to show the results of a post-processing analysis tool that he's written called “Blktrace Time Line” [btt]. Alan had used btt to show the behavior of a mkfs [make file system—sample load] run doing buffered block IO, ultimately pushed to disk by the in-kernel “pdflush” daemon.

Alan is using blktrace, btt and other tools he has written to explore the performance and scalability behavior of the entire Linux storage IO stack, including the different IO schedulers [Linux supports several], Logical Volume and multi-path management via both the “md” [multi-disk] and newer “dm” [device mapper] layers, up through various file systems. Perhaps Alan will present the results of this on-going work at future Gelato Conferences.

2. Scaling Linux to 512 Processors and Beyond

Slides available at:

http://www.gelato.org/pdf/apr2006/gelato_ICE06apr_scaling_hawkes_sgi.pdf

John Hawkes presented the ambitious work that he and other SGI engineers have undertaken to achieve the goal indicated by the presentation title. This presentation is an excellent overview of the topic of operating system scalability, as John provided definitions of scalability and discussed the various limits to scalability that one encounters in attempting to run a “single system image” OS on systems with large processor counts [100s] and very large memory [Terabytes]—see the presentation slides.

John noted that earlier releases of the SGI Itanium-based platform were supported by the 2.4 series of Linux kernels. Much of the scalability work done at SGI was to address the rather severe scalability limitations of the the 2.4 kernel. To achieve acceptable scalability, SGI shipped a “Pro Pack” that included a kernel modified with SGI's enhancements. The current generation of SGI platforms run the “native” 2.6.x kernels supported by the standard enterprise distros. The present day scalability of the 2.6 kernels is due, in no small part, to the efforts of the SGI engineers.

One area of scalability limitation that John discussed was the effect of the platform's NUMA [Non-Uniform Memory Access] architecture. Again, the 2.4 kernel was not “NUMA-aware”, whereas much has been done to make the 2.6 kernel work well on NUMA platforms. It is worth noting that one of John's colleagues, Christoph Lameter, presented a more in-depth review of NUMA support in the Linux kernel in Tuesday's Advanced Topics track. For more information on this topic, see:

http://www.gelato.org/pdf/apr2006/gelato_ICE06apr_numa_lameter_sgi.pdf

3. Improving NFS Performance

Slides available at:

http://www.gelato.org/pdf/apr2006/gelato_ICE06apr_nfs_chubb_unsw.pdf

v0.01– addressed Peter's comments -- 10may06

Peter Chubb presented work in progress at National ICT Australia and the University of New South Wales to understand the behavior of NFS in contemporary environments to aid in the improvement of NFS performance on Linux. NFS does not enjoy a stellar reputation for performance on Linux. Peter posits that the use of SpecSFS'97 by the upstream maintainers as the benchmark for measuring NFS performance may contribute to this. To test this theory, Peter and a research assistant, Shehjar Tikoo, have undertaken an effort to provide tools to capture actual NFS traffic in various application environments to see how folks are actually using NFS today.

The tools, based on nfsdump by Daniel Ellard of Harvard, collect NFS traffic by monitoring network interfaces. A post processing step “anonymizes” the data, replacing all file names, user names and ids, and IP addresses with unique “cookies”. Peter et al can then analyze the mix of operations, transfer sizes, etc. used by the measured system. Another tool to play back the trace against a live system—e.g., to test the effects of proposed enhancements—is still under construction.

To date Peter has collected 24-hour traces from eight servers—one at the UNSW computer center—a “typical”, though “out-of-session”, university timeshare environment, and seven at the Ohio Super-Compter Center—a production, High Performance Computing environment. As Peter notes, these are not sufficient data points to draw final conclusions, and he would love to obtain traces from more commercial and scientific environments. One of the attendees, Evan Felix of PNNL, offered that he would try to obtain traces from PNNL for Peter's analysis.

One thing that Peter did note was that none of the traces collected match the distribution of operations and file sizes used by SpecSFS'97. He offered possible explanations for these differences [see the presentation linked above—Peter has included excellent annotations, so I won't repeat them here].

Going forward, Peter et al, hope to start playing the traces back against a live [instrumented?] Linux system to look for bottlenecks and other impediments to performance in NFS. Peter mentioned that, although in his estimation it is not yet “ready for prime time”, he believes that NFS v4 addresses many limitations in V3. This, too, can be tested when trace playback is working.

4. Wrap-up/Discussion Session

Unfortunately, participation in this session, was not what we'd hoped for, despite a pitch for participation in the Tuesday afternoon "In Search of Collaboration" session. Perhaps this was because Wednesday was the last day of the conference, and previous Gelato meetings have ended mid-day on Wednesday, or perhaps because the competing sessions were of more interest to the attendees. Whatever the reason, by the time we got to the wrap-up discussion session, we were down to a one or two people from each of SGI, UNSW, Intel and HP—folks who are already in fairly frequent communication and collaboration; plus one bona fide user of Itanium systems. Peter Chubb characterized this as “preaching to the converted”. However, we did have an opportunity to ask questions to get more details than we had time for in the more tightly scheduled formal presentation sessions, and to discuss related work we've each been doing.

Perhaps this is as it should be, where scalability is concerned. End users just want us platform providers and researchers to “make it so”. <heavy sigh>

v0.01– addressed Peter's comments -- 10may06