fxguide burn benchmarks report

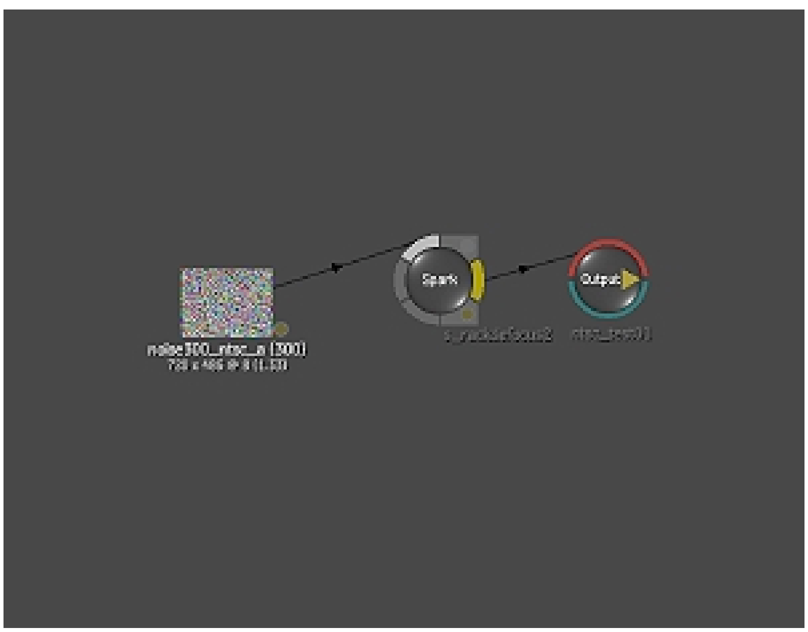
burn version 1.0 release

The fxguide crew did various rendering speed tests utilizing the release of inferno 5.3 (onyx2) and burn 1.0 for Linux. The inferno system is an Onyx2 IR3 system with 4x400Mhz IP27 processors. The burn render farm consists of four IBM X-335 rack mount PC systems with dual 2.80Ghz Xeon Processors. The systems were connected using 100baseT through a switch (Discreet recommends gigabit Ethernet for network connections, but we also figure 100baseT would be a common network infrastructure at facilities). More complete details regarding the systems are located at the end of this document.

Regarding the benchmarks, they admittedly weren't fully controlled in a laboratory manner. However, each render was run twice to ensure that there weren't any extenuating circumstances which effected the results. Work and rendering was continued on the main inferno workstation as the burn renders were in progress.

Render time results were based upon the Backburner Monitor log (for burn) and the text log which is generated by batch (for inferno). This means that the render times for burn don't contain the time it takes for batch to output the setup files and clips in preparation for burn to render (which can be 5 to 20 seconds). But it does accurately compare actual render times for the setups.

Various tests were used to try and discover where burn was most appropriate to be used and when the rendering results were less than ideal. Comments (and theories) regarding the results are listed after each test.



Test 1A

NTSC 300 frame color noise clip feeds Sapphire RackDefocus (default settings)

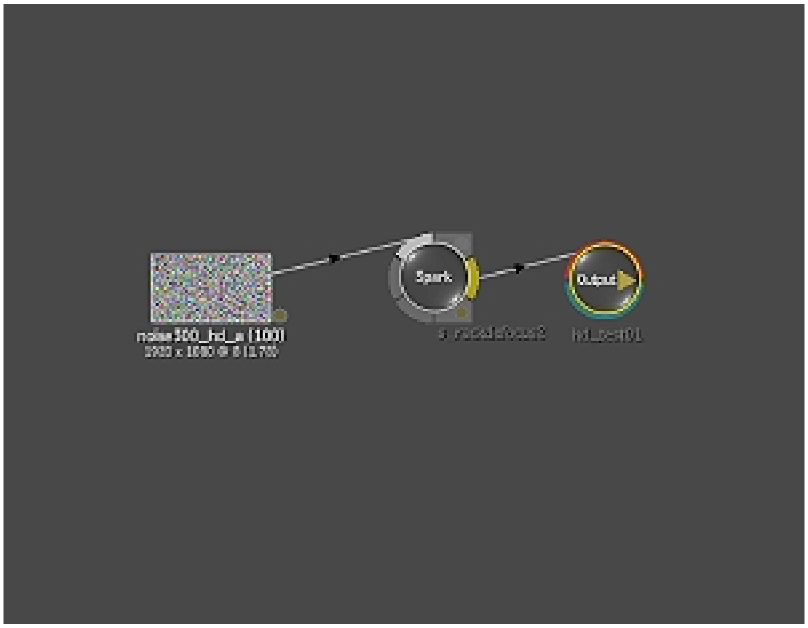
Batch render :58
Burn render 1:13

Test 1B

NTSC 300 frame color noise clip feeds Sapphire RackDefocus (default settings with subpixel set to "YES")

Batch render 3:12
Burn render 1:44

What is interesting about this test is the effect which turning subpixel rendering on has to render times. Right off the bat it becomes apparent that processor-based tasks such as sparks show the most benefit to being rendered using burn.



Test 2A

HD1920x1080 100 frame color noise clip feeds Sapphire RackDefocus (default settings)

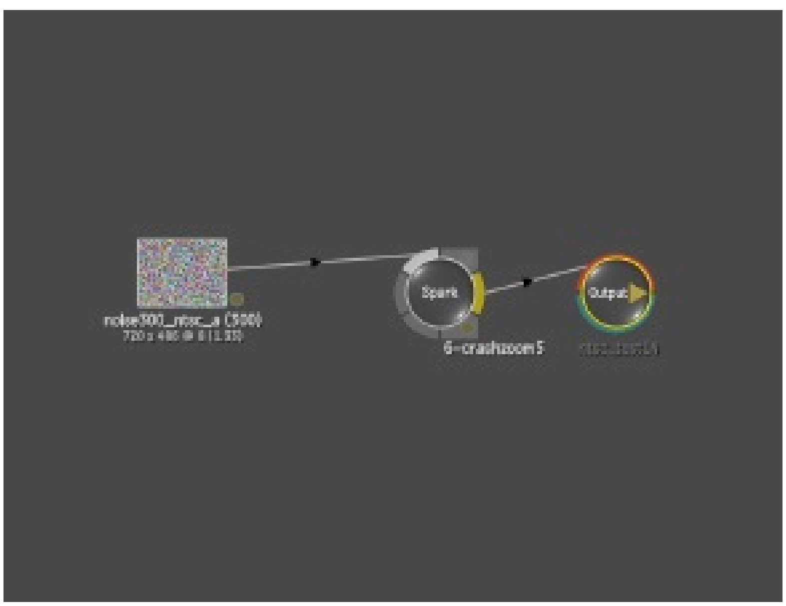
Batch render	2:34
Burn render	2:27

Test 2B

HD1920x1080 100 frame color noise clip feeds Sapphire RackDefocus (default settings with subpixel set to "YES")

Batch render	13:19
Burn render	5:16

For this test, we took the same setup as above, but used HD images instead of NTSC images (and also reduced the length of the clip for shorter testing times). This test also dramatically shows the difference of using a processor-demanding effect on render times. The burn render at HD resolution using subpixel rendering ends up being more than twice as fast as the same render on inferno. What is also interesting is that even though the size of the files being transferred over 100BaseT are six times larger than the NTSC images, the burn render is completed sooner due to the speed of the processing of the combined burn systems.

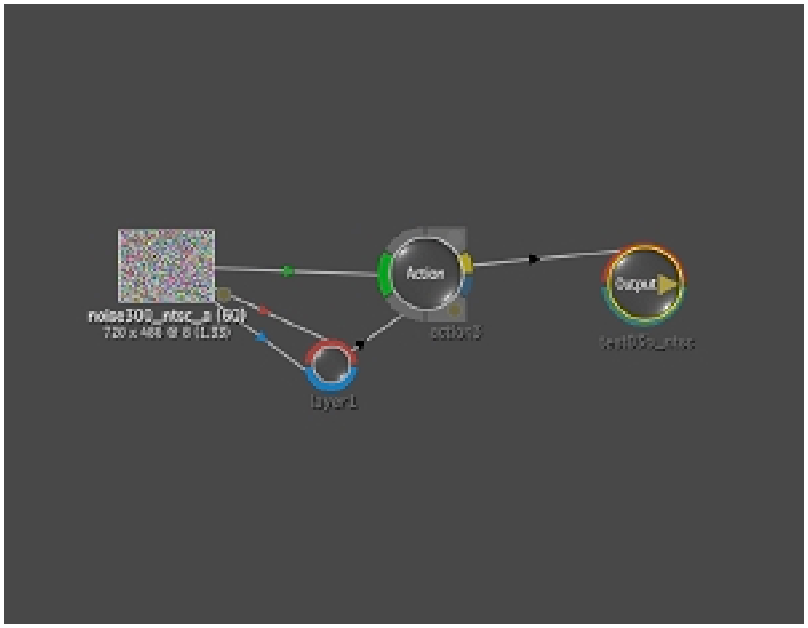


Test 3

100 Frame Render, SpeedSix crashzoom, default settings (+ 4 samples)

Batch render (NTSC)	:35
Burn render (NTSC)	:53
Batch render (HD)	6:03
Burn render (HD)	2:40

Next, we tried one of the SpeedsSix Monster sparks, crashzoom. The burn render beat the batch render, but not by as big a margin as the sapphire sparks.



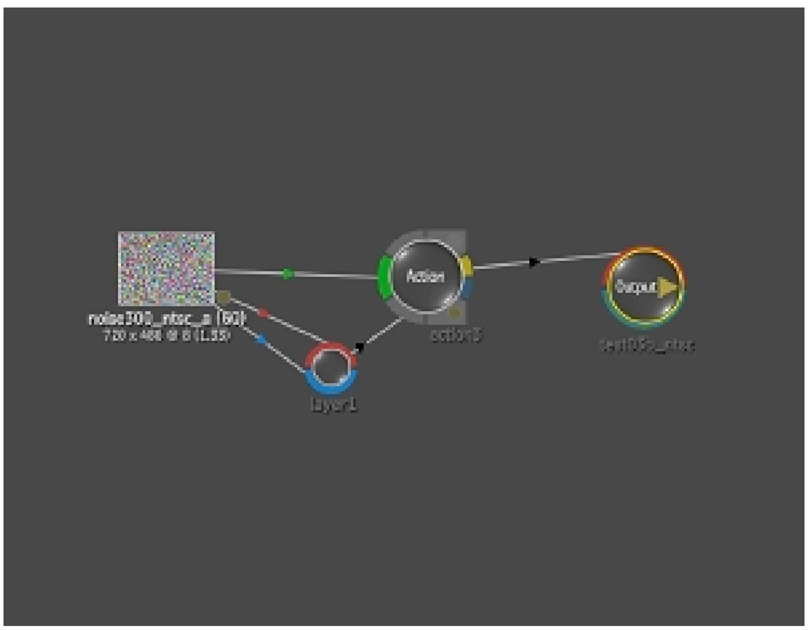
Test 4A

60 frame action render, with 60 frame source clip fed from batch schematic.

***Action node**: 1 layer/image. 4 Samples.*
***Layer Key**: Standard LUM key, Matte Shrink –6, Suppress Fgd Green –25*
***Layer Blur**: Gaussian Front 10 (blur of 26 for HD render)*

Batch render (NTSC)	:29
Burn render (NTSC)	1:01
Batch render (HD)	3:45
Burn render (HD)	3:40

This once again shows the benefits that burn has when it renders setups using higher resolution clips. This is even considering the 100BT network infrastructure.



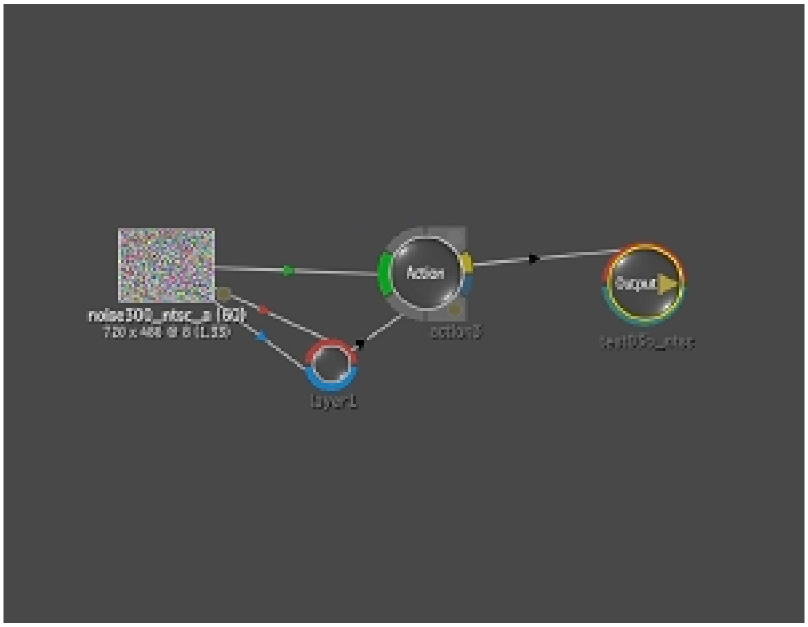
Test 4B

60 frame action render, with 60 frame source clips fed from batch schematic.

***Action node**: 1 layer/image. 16 Samples.*
***Layer Key**: Standard LUM key, Matte Shrink –6, Suppress Fgd Green –25*
***Layer Blur**: Gaussian Front 10 (blur of 26 for HD render)*

Batch render (NTSC)	:33
Burn render (NTSC)	1:26
Batch render (HD)	3:58
Burn render (HD)	5:40

This test uses essentially the same setup as Test 4A, but with the Action samples cranked up to 16 Samples. Burn becomes slower with such a high anti-aliasing setting.



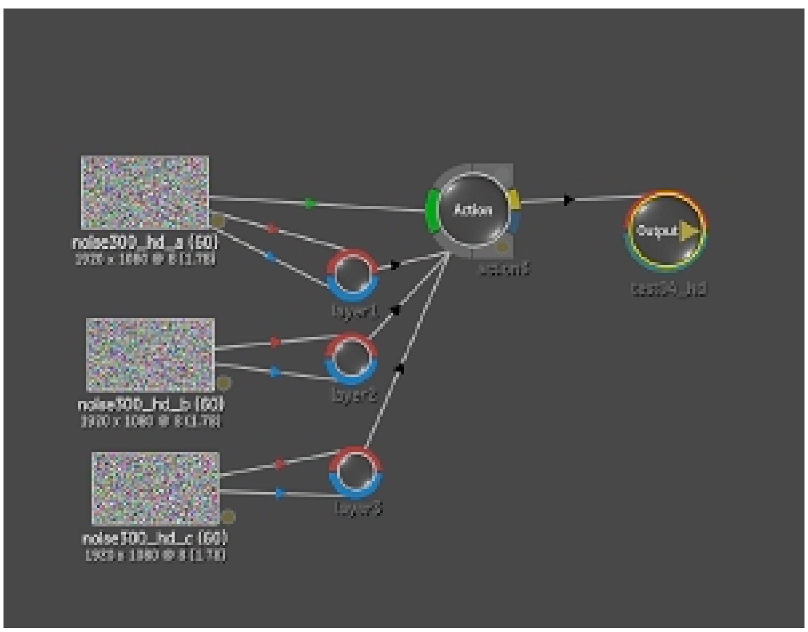
Test 4C

60 frame action render, with 60 frame source clips fed from batch schematic.

***Action node**: 1 layer/image. 4 Samples, Texture Off.*
***Layer Key**: Standard LUM key, Matte Shrink –6, Suppress Fgd Green –25*
***Layer Blur**: Gaussian Front 10 (blur of 26 for HD render)*

Batch render (NTSC)	:53
Burn render (NTSC)	1:31
Batch render (HD)	6:14
Burn render (HD)	6:32

This test uses essentially the same setup as Test 4A, but with Texture in Action turned off. There was an increase in render times for both the batch and burn renders. However, in checking the quality of the render using Difference (see below), the two burn renders (4A and 4C) showed virtually no variation in image quality. So for a render similar to this one, it makes sense to leave Texture On if you are utilizing burn.



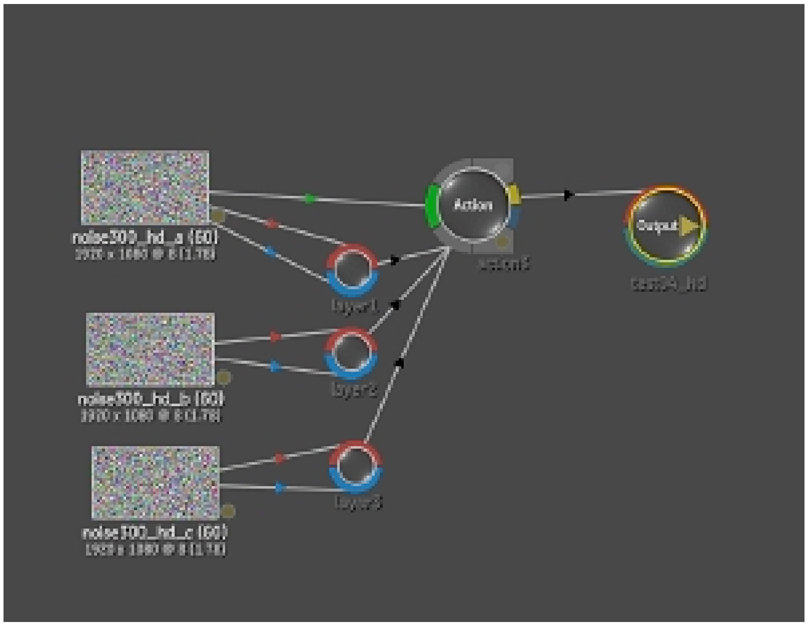
Test 5A

60 frame action render, with 60 frame source clips fed from batch schematic.

***Action node**: 3 layers/images. 4 Samples.*
***Layer Key**: Standard LUM key, Matte Shrink –6, Suppress Fgd Green –25*
***Layer Blur**: Gaussian Front 10 (blur of 26 for HD render)*

Batch render (NTSC)	1:39
Burn render (NTSC)	1:35
Batch render (HD)	9:59
Burn render (HD)	5:28

What happens when more layers are added to the mix? Initially, I would have thought that the network would start to get in the way of things, but it isn't the case. The NTSC render for burn is actually faster now than the batch render. The results are even more impressive for the burn render, which is approaching twice as fast as the batch render.



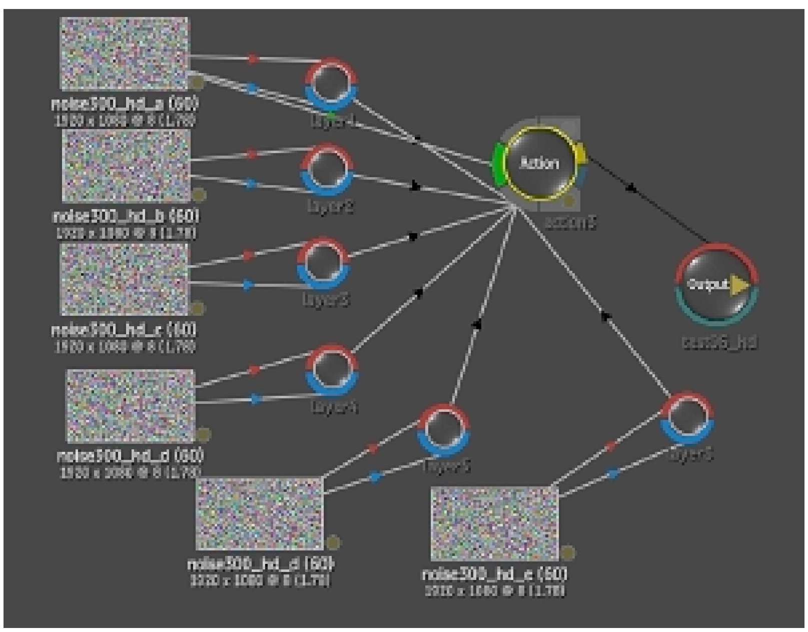
Test 5B

60 frame action render, with 60 frame source clips fed from batch schematic.

***Action node**: 3 layers/images. 4 Samples, Motion Blur Samples 10*
***Layer Key**: Standard LUM key, Matte Shrink –6, Suppress Fgd Green –25*
***Layer Blur**: Gaussian Front 10*

Batch render (NTSC)	1:45
Burn render (NTSC)	4:35

This test utilizes essentially the same setup as Test 4, but with Motion Blur turned on and samples set to 10. Examining just the NTSC results, motion blur settings have a dramatic impact upon burn render times.



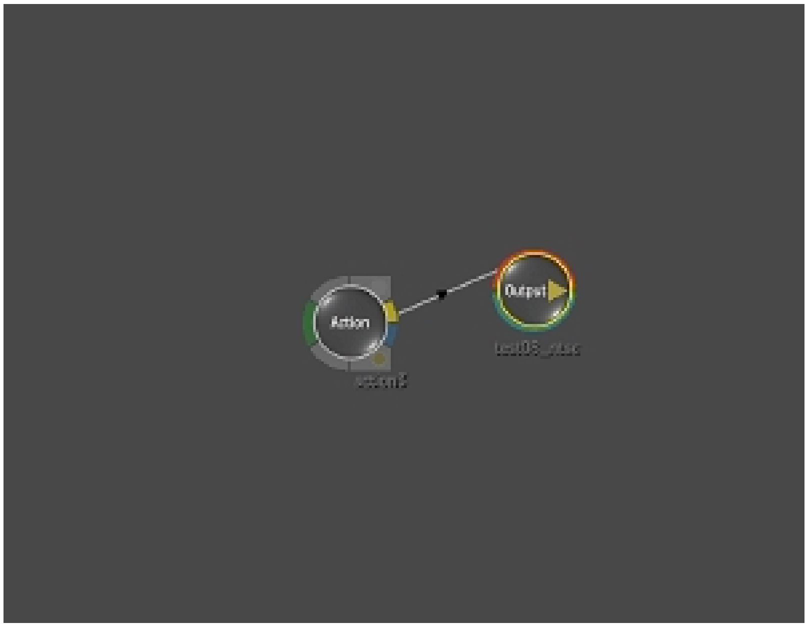
Test 6

60 frame action render, with 60 frame source clips fed from batch schematic.

***Action node**: 6 layers/images. 4 Samples, Motion Blur Samples 10*
***Layer Key**: Standard LUM key, Matte Shrink –6, Suppress Fgd Green –25*
***Layer Blur**: Gaussian Front 10 10 (blur of 26 for HD render)*

Batch render (NTSC)	3:06
Burn render (NTSC)	2:41
Batch render (HD)	19:54
Burn render (HD)	11:22

This test utilizes essentially the same setup as Test 4A, but with an additional three layers to see if there is a noticeable impact upon rendering times. Burn keeps up quite nicely.



Test 7

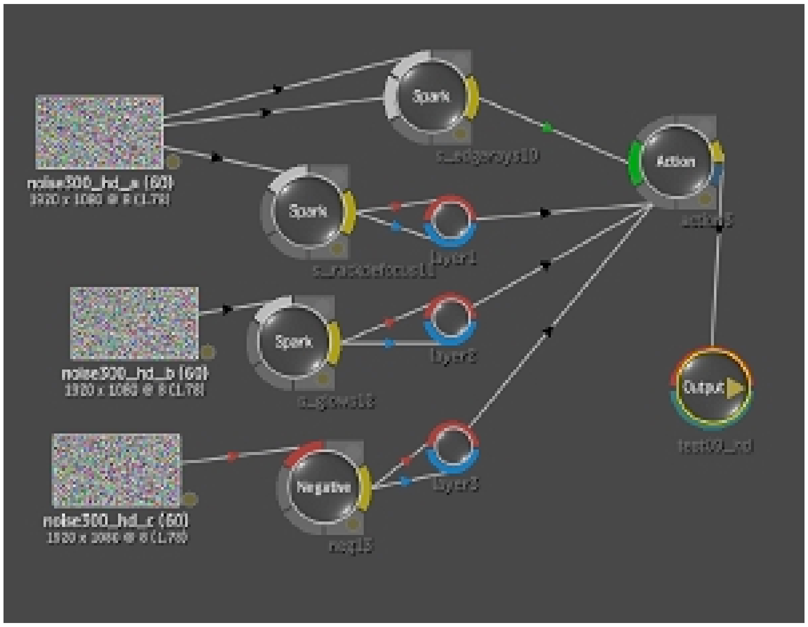
60 frame action render, with single frame source clips fed directly in action and not from the batch schematic.

***Action node**: 6 layers/images. 4 Samples, Motion Blur Samples 10*
***Layer Key**: Standard LUM key, Matte Shrink –6, Suppress Fgd Green –25*
***Layer Blur**: Gaussian Front 10*

Batch render (NTSC) :12
Burn render (NTSC) 2:45

For this test (which is very similar to Test 5), I was curious to see how direct action layers were handled by burn. A single frame used as an action layer fed from the desktop does not need to be updated and reprocessed (if there is no change in the keyer, color corrector, blur, etc) every frame. This means that render times can be considerably less if you use action layers fed from the desktop instead of batch-fed action layers in certain applications.

How does burn handle this? It treats these action layers the same as a batch fed-layer, so the render time difference is huge in this specific instance. When batch creates a burn setup file, it explodes all the desktop-fed layers into batch-fed layers before sending it off to burn. Something to be aware of….and can be especially critical when using higher resolution images. Having a lot of single frame layers directly in action can tilt render times in favor of batch. A higher speed network certainly would improve things, but some of the difference is still due to the way batch handles direct and indirect layers.



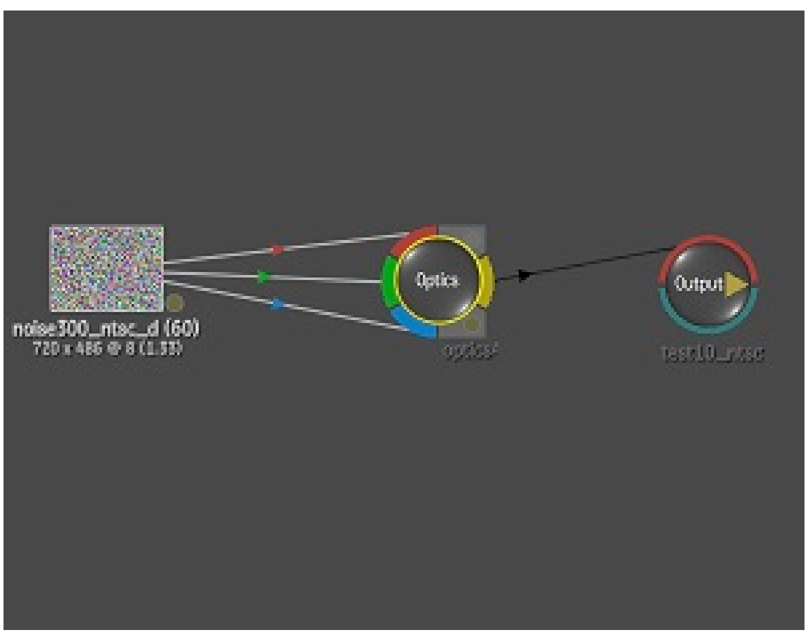
Test 8

30 frame action render, with 60 frame source clips fed from batch schematic.

***Action node**: 3 layers/images. 4 Samples, Motion Blur Samples 10*
***Layer Key (all)**: Standard LUM key, Matte Shrink –6, Suppress Fgd Green –25*
***Layer Blur (all)**: Gaussian Front 10 (blur of 26 for HD render)*
***Bknd Layer**: Sapphire EdgeRays, Default Settings +Subpixel = YES*
***Layer 1**: Sapphire RackDefocus, Default Settings*
***Layer 2**: Sapphire Glows, Default Settings*
***Layer 3**: Negative Node*

Batch render (NTSC)	1:20
Burn render (NTSC)	1:27
Batch render (HD)	10:15
Burn render (HD)	5:11

Burn holds up quite well with this mix of action effects and spark effects feeding the action layers.



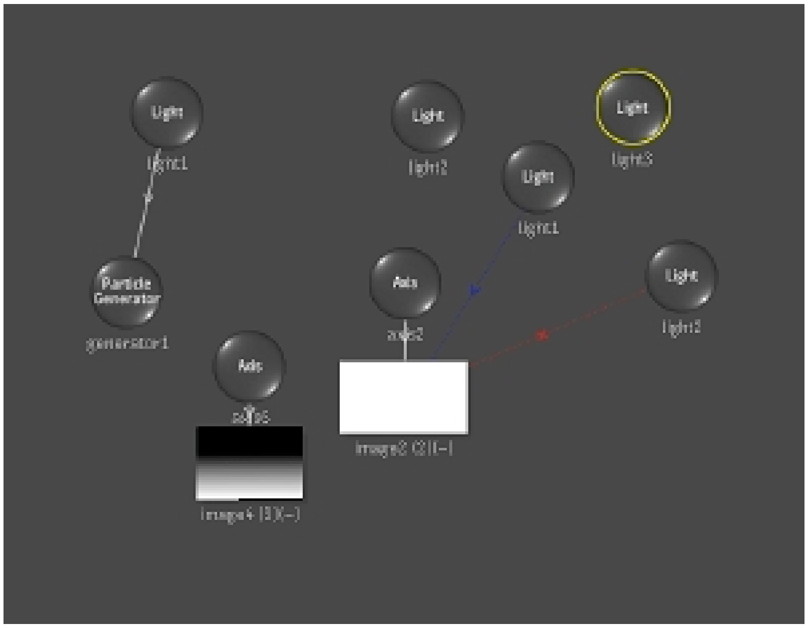
Test 9

60 frame render.

***Optics node**: Default Settings*

Batch render (NTSC)	1:20
Burn render (NTSC)	9:29
Batch render (HD)	1:40
Burn render (HD)	59:51

What more can we say other than....don’t use optics when you’re using burn. A heavily-reliant OpenGL effect which has not been optimized for burn at this point in time.



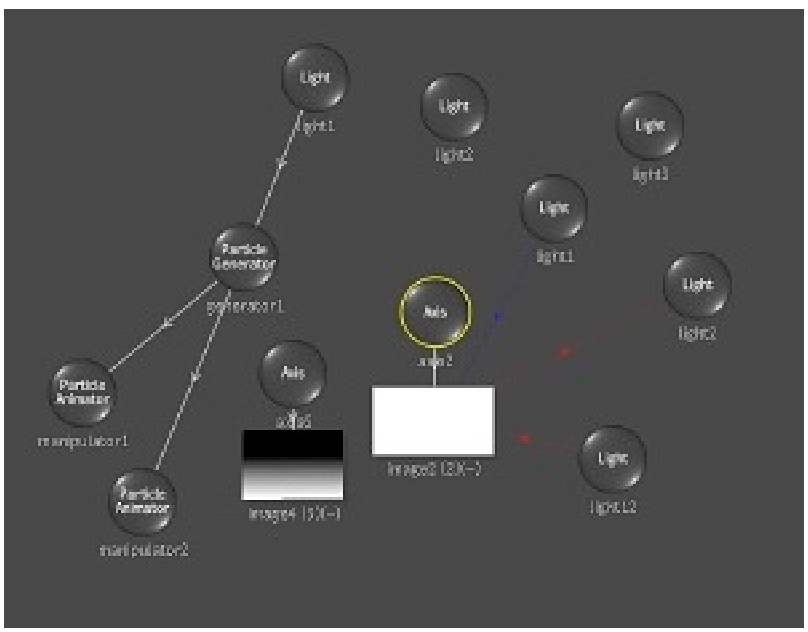
Test 10

90 frame action particle render. All clips single frame, fed directly in action.

***Action node**: Shading On, 4 Lights, 1 Sample*
***Particle Generator**: Spheres, Number 300 (Constant), Time Steps 2, Lifetime 150*

Batch render (NTSC)	3:52
Burn render (NTSC)	3:19
Batch render (HD)	6:58
Burn render (HD)	5:10

Burn does a nice job of keeping up with particle renders. We did run into a bug where the "frozen in time" particles effect (dropping the lifetime to 0) does not work when rendering over the network to burn. Just a heads up.....



Test 11

90 frame action particle render. All clips single frame, fed directly in action.

***Action node**: Shading On, 4 Lights*
***Particle Generator**: Spheres, Number 300 (Constant), Time Steps 2, Lifetime 150*
***Particle Manipulator**: transparency = lifetimel*
***Particle Manipulator**: speed = speed*0.95+turbulence3(pos*0.01,1)*

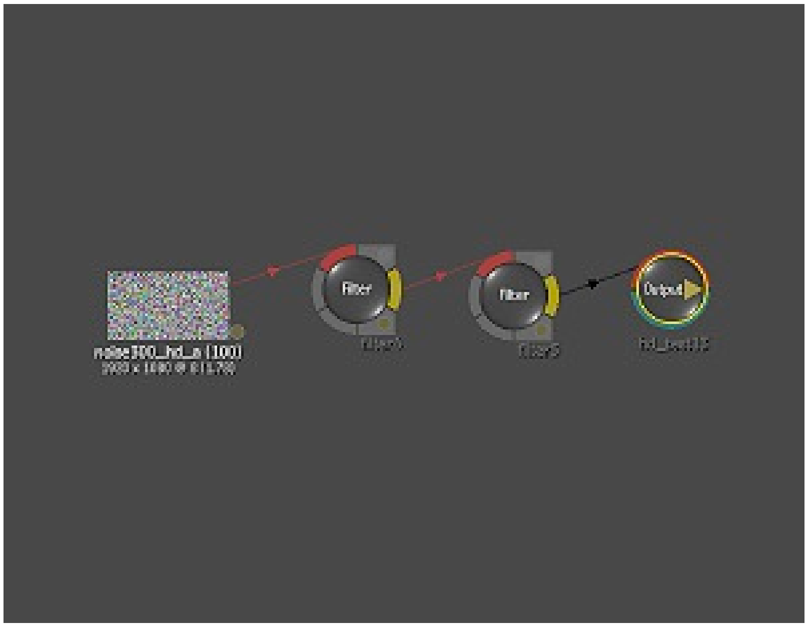
Action Samples: 4

Batch render (HD)	8:47
Burn render (HD)	6:46

Action Samples: 4

Batch render (HD)	27:04
Burn render (HD)	22:23

Adding some particle manipulators into the mix and increasing the samples up to 4 didn't hurt the burn rendering times.



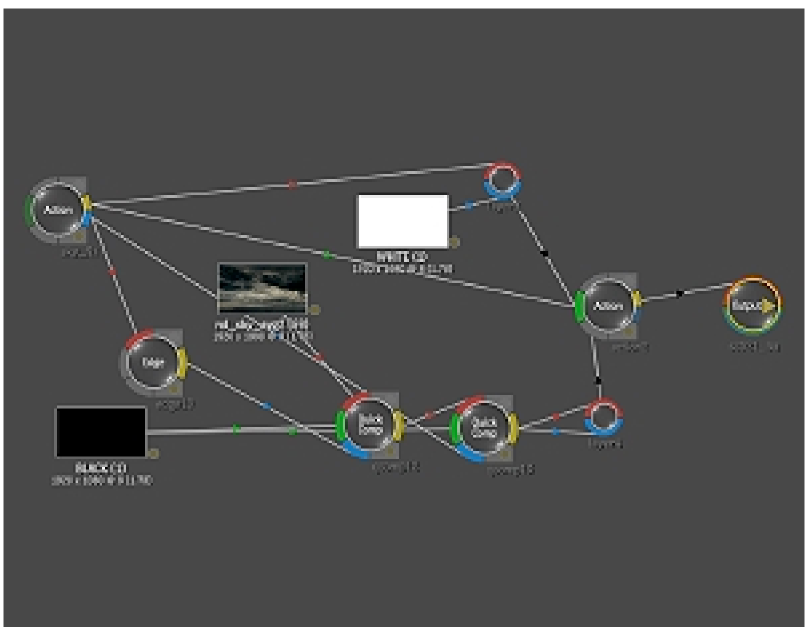
Test 12

Filter processing. 100 frames.

Filter 1: *Soften_heavy*
Filter 2: *Emboss_heavy_soft*

Batch render (NTSC)	1:50
Burn render (NTSC)	1:01
Batch render (HD)	5:10
Burn render (HD)	3:08

We were curious as to the speed increase for the Sapphire Sparks, so we decided to try the discreet filters. Taking two filters from a 100 frame clip, we found that the burn renders were considerably faster, even for NTSC work.



Test 13

59 frame batch render

This setup from a recent project has a couple of action nodes, each with multiple MK layers (and layers within the MK. An edge node and some quick comp nodes are also thrown into the mix. All clips in the setup are HD resolution or higher.

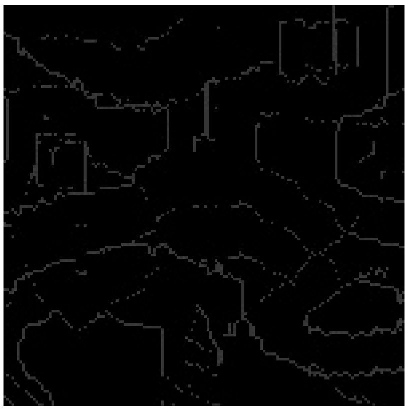
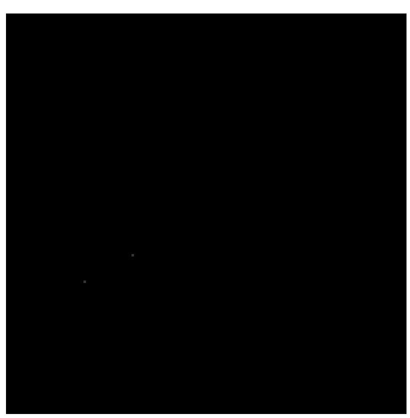
Batch render (HD)	15:20
Burn render (HD)	12:08

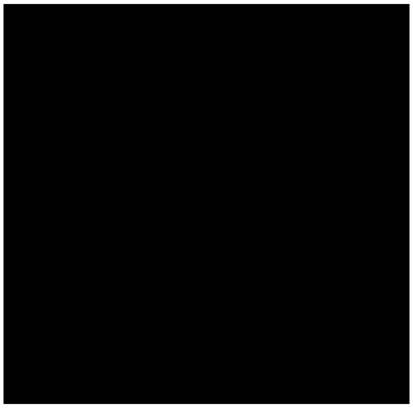
For a recent project, I used burn a great deal – some render times were faster than batch and others were slower. In this specific case, I think some motion blur that I had in the upstream action node tipped things in batch's favor. That being said, burn was indispensable on this job because I was able to continue working finessing comps while renders were being done in the background. Take this render as an example – I was able to continue working instead of being locked out of the machine for 10 minutes or so. Fantastic. Unless, of course, you want a coffee break…..

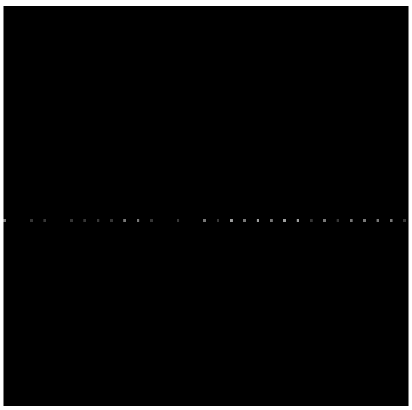
Burn image quality tests

Next, we were interested in seeing any differences between renders on the burn versus the effects platforms. In order to explore cross platform rendering issues (since renders are different between effects systems based upon the graphics cards), we also threw a flame into the mix. The difference module was set to YUV, softness of 100 for each, gain of 2000. This is cranked up considerably in order to see the differences between the renders. We used four of the speed test setups to check the differences between the various modules:

Test 4A (action render): inferno, flame, burn
Test 4C (action render):: inferno, flame, burn
Test 2A (Sapphire spark): inferno, burn
Test 3 (SpeedSix spark): inferno, burn

 **4A inferno vs. 4A burn**	 **4C inferno vs. 4C burn**	This result shows the difference between a render done on inferno versus a render done on burn.
 **4A inferno vs. 4A flame**	 **4C inferno vs. 4C flame**	As a frame of reference, this shows the difference between the same renders on inferno and flame.
 **4A inferno vs. 4C inferno**	 **4A burn vs. 4C burn**	Finally, this shows the difference between the Texture On and Texture Off renders on each platform. Note that the burn difference is quite minimal.

 ***2A inferno vs. 2A burn***	Rendering differences between the Sapphire spark renders are too small to be seen. Quality is very consistent on this particular spark.

 ***3 inferno vs. 3 burn*** (crashzoom spark)	 ***3 inferno vs. 3 burn*** (defocus spark)	There were some rendering anomalies on the SpeedSix CrashZoom spark render. The difference showed very thin white lines forming four quadrants on the screen. This was not visible to the naked eye. To see if this was apparent on other SpeedSix sparks, defocus was used. There were no differences between the renders using this spark

The test systems

Network connection: 100baseT, switched

SGI Onyx2IR3 inferno – Version 5.3

```
apu: /usr/discreet/inferno_5.3 % hinv
4 400 MHZ IP27 Processors
CPU: MIPS R12000 Processor Chip Revision: 3.5
FPU: MIPS R12010 Floating Point Chip Revision: 3.5
Main memory size: 3072 Mbytes
Instruction cache size: 32 Kbytes
Data cache size: 32 Kbytes
Secondary unified instruction/data cache size: 8 Mbytes
Integral SCSI controller 9: Version Fibre Channel AIC-1160, revision 2
  Disk drive: unit 3 on SCSI controller 9
  Disk drive: unit 4 on SCSI controller 9
  Disk drive: unit 5 on SCSI controller 9
  Disk drive: unit 6 on SCSI controller 9
  Disk drive: unit 7 on SCSI controller 9
  Disk drive: unit 8 on SCSI controller 9
  Disk drive: unit 9 on SCSI controller 9
  Disk drive: unit 10 on SCSI controller 9
Integral SCSI controller 0: Version QL1040B (rev. 2), single ended
  Disk drive: unit 1 on SCSI controller 0
  Disk drive: unit 2 on SCSI controller 0
  Disk drive: unit 3 on SCSI controller 0
  CDROM: unit 6 on SCSI controller 0
Integral SCSI controller 1: Version QL1040B (rev. 2), single ended
  Tape drive: unit 6 on SCSI controller 1: 8mm(8500) cartridge
Integral SCSI controller 10: Version Fibre Channel AIC-1160, revision 2
  Disk drive: unit 4 on SCSI controller 10
  Disk drive: unit 5 on SCSI controller 10
  Disk drive: unit 6 on SCSI controller 10
  Disk drive: unit 7 on SCSI controller 10
  Disk drive: unit 8 on SCSI controller 10
  Disk drive: unit 9 on SCSI controller 10
  Disk drive: unit 10 on SCSI controller 10
  Disk drive: unit 11 on SCSI controller 10
Integral SCSI controller 15: Version QL1040B (rev. 2), differential
Integral SCSI controller 14: Version QL1040B (rev. 2), differential
Integral SCSI controller 13: Version QL1040B (rev. 2), single ended
  Tape drive: unit 2 on SCSI controller 13: 8mm (AIT) cartridge
Integral SCSI controller 16: Version QL1040B (rev. 2), differential
Integral SCSI controller 8: Version QL1040B (rev. 2), single ended
  Disk drive: unit 1 on SCSI controller 8
Integral SCSI controller 12: Version Fibre Channel AIC-1160, revision 2
  Disk drive: unit 3 on SCSI controller 12
  Disk drive: unit 4 on SCSI controller 12
  Disk drive: unit 5 on SCSI controller 12
  Disk drive: unit 6 on SCSI controller 12
  Disk drive: unit 7 on SCSI controller 12
  Disk drive: unit 8 on SCSI controller 12
  Disk drive: unit 9 on SCSI controller 12
Integral SCSI controller 11: Version Fibre Channel AIC-1160, revision 2
  Disk drive: unit 1 on SCSI controller 11
  Disk drive: unit 2 on SCSI controller 11
  Disk drive: unit 3 on SCSI controller 11
  Disk drive: unit 4 on SCSI controller 11
  Disk drive: unit 5 on SCSI controller 11
  Disk drive: unit 6 on SCSI controller 11
  Disk drive: unit 7 on SCSI controller 11
  Disk drive: unit 8 on SCSI controller 11
Graphics board: InfiniteReality3
Integral Fast Ethernet: ef0, version 1, module 1, slot io1, pci 2
Iris Audio Processor: version RAD revision 7.0, number 1
Iris Audio Processor: revision 2
```

```
Origin FIBRE CHANNEL board, module 1 slot 11: Revision 4
Origin BASEIO board, module 1 slot 1: Revision 4
Origin MSCSI board, module 1 slot 4: Revision 4
Origin PCI XIO board, module 1 slot 2: Revision 4
Origin FIBRE CHANNEL board, module 1 slot 6: Revision 4
XT-HDIO Video: controller 0, unit 1, version 0x0
DIVO Video: controller 0 unit 0: Input, Output
IOC3/IOC4 external interrupts: 1
PCI card, bus 0, slot 0, Vendor 0x9, Device 0x1
```

Burn systems – 4 total

```
IBM X-335 Xeon rack mount PC
2 x  2.80Ghz 400Mhz 512KB L2 Cache Xeon Processor Upgrade
2 Gigabytes RAM
IBM 18.2 GB 10K-rpm Ultra160 SCSI hot_Swap SL HDD
```

SGI Octane2 Flame – Version 5.3

```
troy 1# hinv
2 400 MHZ IP30 Processors
CPU: MIPS R12000 Processor Chip Revision: 3.5
FPU: MIPS R12010 Floating Point Chip Revision: 0.0
Main memory size: 3328 Mbytes
Xbow ASIC: Revision 1.4
Instruction cache size: 32 Kbytes
Data cache size: 32 Kbytes
Secondary unified instruction/data cache size: 2 Mbytes
Integral SCSI controller 0: Version QL1040B (rev. 2), single ended
  Disk drive: unit 1 on SCSI controller 0
  Disk drive: unit 2 on SCSI controller 0
  Disk drive: unit 3 on SCSI controller 0
Integral SCSI controller 1: Version QL1040B (rev. 2), single ended
  Tape drive: unit 5 on SCSI controller 1: DAT
Integral SCSI controller 2: Version Fibre Channel AIC-1160, revision 2
  Disk drive: unit 1 on SCSI controller 2
  Disk drive: unit 2 on SCSI controller 2
  Disk drive: unit 3 on SCSI controller 2
  Disk drive: unit 4 on SCSI controller 2
  Disk drive: unit 5 on SCSI controller 2
  Disk drive: unit 6 on SCSI controller 2
  Disk drive: unit 7 on SCSI controller 2
  Disk drive: unit 8 on SCSI controller 2
Integral SCSI controller 3: Version Fibre Channel AIC-1160, revision 2
  Disk drive: unit 1 on SCSI controller 3
  Disk drive: unit 2 on SCSI controller 3
  Disk drive: unit 3 on SCSI controller 3
  Disk drive: unit 4 on SCSI controller 3
  Disk drive: unit 5 on SCSI controller 3
  Disk drive: unit 6 on SCSI controller 3
  Disk drive: unit 7 on SCSI controller 3
  Disk drive: unit 8 on SCSI controller 3
IOC3/IOC4 serial port: tty1
IOC3/IOC4 serial port: tty2
IOC3 parallel port: plp1
Graphics board: EMXI
Integral Fast Ethernet: ef0, version 1, pci 2
Iris Audio Processor: version RAD revision 12.0, number 1
Digital Video: unit 1, revision 5.2, TMI: revision 2, CSC: revision 1
PCI card, bus 0, slot 0, Vendor 0x9, Device 0x1
```