Intel i7-2720Q, **2.2GHz**, **Turbo Disabled**
Windows 7, 64-bit OS, on MacBook Pro

Test Description	**Application**	**KDU-S7.0**			**KDU-7.0**	**VS 6.3.1**	**V 6.4.1**	**KDU-S7.0 Relative Performance Advantage (relative to most relevant other**		
		FPS	Msamples/s	Mbit/s	FPS	FPS	FPS	w.r.t. KDU-7	w.r.t. VS6.3.1	w.r.t. v6.4.1
2K Digital cinema decompress/render: content is **2048x857**, full XYZ, 24fps, encoded at max allowable bit-rate of 244Mbit/s = **5.8bits/pel**; 240 frames, rendered twice, using "-display" option.	**kdu_vex_fast** (2 engines, 4 threads each)	21.978	115.7	223.4	15.35	19.95	13.71	**143%**	**110%**	**160%**
	kdu_vex_fast (8 engines, 1 thread each)	NB: This mode involves high delay and does not scale as well to machines with many CPUs; included for fairest v6/vs6 comparison.				20.8	14.81		**106%**	**148%**
2K Digital cinema decompress/render: content is **2048x857**, full XYZ, 48fps, encoded at max allowable bit-rate of 244Mbit/s = **2.9bits/pel**; 240 frames, rendered twice, using "-display" option.	**kdu_vex_fast** (2 engines, 4 threads each)	40.81	214.9	207.5	28.33	34.38	23.26	**144%**	**119%**	**175%**
	kdu_vex_fast (8 engines, 1 thread each)	NB: This mode involves high delay and does not scale as well to machines with many CPUs; included for fairest v6/vs6 comparison.				38.37	26.98		**106%**	**151%**
4K Digital cinema decompress/render: content is **4096x1714**, full XYZ, 24fps, encoded at max allowable bit-rate of 244Mbit/s = **1.45bits/pel**; 240 frames, rendered twice, using "-display" option.	**kdu_vex_fast** (2 engines, 4 threads each)	18.78	395.5	190.9	12.74	14.51	9.04	**147%**	**129%**	**208%**
	kdu_vex_fast (8 engines, 1 thread each)	NB: This mode involves high delay and does not scale as well to machines with many CPUs; included for fairest v6/vs6 comparison.				17.34	12.09		**108%**	**155%**

Intel i7-2720Q, **2.2GHz**, **Turbo Disabled**
Windows 7, 64-bit OS, on MacBook Pro

Test Description	**Application**	**KDU-S7.0**			**KDU-7.0**	**VS 6.3.1**	**V 6.4.1**	**KDU-S7.0 Relative Performance Advantage (relative to most relevant other**		
		FPS	Msamples/s	Mbit/s	FPS	FPS	FPS	w.r.t. KDU-7	w.r.t. VS6.3.1	w.r.t. v6.4.1
2K Digital cinema decompress/render: content is **2048x857**, full XYZ, 24fps, encoded at max allowable bit-rate of 244Mbit/s = **5.8bits/pel**; 240 frames, rendered twice, using "-in_memory'.	**kdu_v_expand** (NB: all threads work together on each frame)	21.835	115.0	222	15.15	15.89	11.71	**144%**	**137%**	**187%**
2K Digital cinema decompress/render: content is **2048x857**, full XYZ, 48fps, encoded at max allowable bit-rate of 244Mbit/s = **2.9bits/pel**; 240 frames, rendered twice, using "-display" option.	**kdu_v_expand** (NB: all threads work together on each frame)	40.34	212.4	205.1	27.84	24.12	18.11	**145%**	**167%**	**223%**
4K Digital cinema decompress/render: content is **4096x1714**, full XYZ, 24fps, encoded at max allowable bit-rate of 244Mbit/s = **1.45bits/pel**; 240 frames, rendered twice, using "-display" option.	**kdu_v_expand** (NB: all threads work together on each frame)	18.6	391.7	189.1	13.01	9.074	7.32	**143%**	**205%**	**254%**

Intel i7-2720Q, **2.2GHz**, **Turbo Disabled**
Windows 7, 64-bit OS, on MacBook Pro

Test Description	**Application**	**KDU-S7.0**			**KDU-7.0**	**VS 6.3.1**	**V 6.4.1**	**KDU-S7.0 Relative Performance Advantage (relative to most relevant other**		
		FPS	Msamples/s	Mbit/s	FPS	FPS	FPS	w.r.t. KDU-7	w.r.t. VS6.3.1	w.r.t. v6.4.1
Big Image Compress, 2bpp: content is **13,333x13,333**, 24bit RGB; Cblk={64,64}, Clevels=9, ORGgen_plt=yes, Cprecincts={256,256}; **2 bits/pel**	**kdu_compress** (with -rate 2)	0.240	128.0	85.36	0.177	0.117	0.084	**136%**	**205%**	**285%**
Big Image **Decompress**, 2bpp: content is **13,333x13,333**, 24bit RGB; Cblk={64,64}, Clevels=9, ORGgen_plt=yes, Cprecincts={256,256}; **2 bits/pel**	**kdu_expand** (output file writing skipped)	0.557	297.3	198.2	0.407	0.341	0.273	**137%**	**163%**	**204%**
Big Image compress, **Incr. flush**, 2bpp: content is **13,333x13,333**, 24bit RGB; Cblk={64,64}, Clevels=7, ORGgen_plt=yes, Cprecincts={256,256},…,{4,128}, flush_period=24; **2 bits/pel**	**kdu_compress** (with -slope and -flush_period)	0.275	146.8	97.84	0.170	0.119	0.085	**161%**	**231%**	**322%**
Big Image Compress, 2bpp: as above, but using "**-slope**"	**kdu_compress** (with -slope)	0.319	170.1	113.4	0.223	Note 2	0.124	**143%**	**Note 2**	**257%**
Big Image Compress, **Incr. flush**, 2bpp: as above, but using "**-slope**"	**kdu_compress** (with -slope and flush_period)	0.332	177.2	118.1	0.206	Note 2	0.129	**161%**	**Note 2**	**258%**
Big Image Compress, 2bpp: as above, but using "**Qstep**"	**kdu_compress** (with Qstep)	0.427	227.9	151.9	0.322	0.223	0.161	**133%**	**191%**	**265%**
Big Image Compress, **Incr. flush**, 2bpp: as above, but using "**Qstep**"	**kdu_compress** (with Qstep and -flush_period)	0.478	255.2	170.1	0.291	0.227	0.165	**164%**	**211%**	**290%**

Intel i7-2720Q, **2.2GHz**, **Turbo Disabled**
Windows 7, 64-bit OS, on MacBook Pro

Test Description	**Application**	**KDU-S7.0**			**KDU-7.0**	**VS 6.3.1**	**V 6.4.1**	**KDU-S7.0 Relative Performance Advantage (relative to most relevant other**		
		FPS	Msamples/s	Mbit/s	FPS	FPS	FPS	w.r.t. KDU-7	w.r.t. VS6.3.1	w.r.t. v6.4.1
Big Image **Lossless** Compress content is **13,333x13,333**, 24bit RGB; Cblk={64,64}, Clevels=9, ORGgen_plt=yes, Cprecincts={256,256}; Creversible=yes	**kdu_compress**	0.1105	58.9	193.7	0.085	0.099	0.078	**130%**	**112%**	**142%**
Big Image **Lossless Decompress** content is **13,333x13,333**, 24bit RGB; Cblk={64,64}, Clevels=9, ORGgen_plt=yes, Cprecincts={256,256}; Creversible=yes	**kdu_expand** (output file writing skipped)	0.1335	71.2	234.1	0.0942	0.128	0.093	**142%**	**105%**	**144%**
Big Image **Lossess** Compress, **Incr. flush**: content is **13,333x13,333**, 24bit RGB; Cblk={64,64}, Clevels=7, ORGgen_plt=yes, Cprecincts={256,256},…,{4,128}, flush_period=24; Creversible=yes	**kdu_compress** (with -flush_period)	0.1235	65.9	216.5	0.0814	0.101	0.077	**152%**	**122%**	**160%**

Intel i7-2720Q, **2.2GHz**, **Turbo Disabled**
Windows 7, 64-bit OS, on MacBook Pro

Test Description	**Application**	**KDU-S7.0**			**KDU-7.0**	**VS 6.3.1**	**V 6.4.1**	**KDU-S7.0 Relative Performance Advantage (relative to most relevant other**		
		FPS	Msamples/s	Mbit/s	FPS	FPS	FPS	w.r.t. KDU-7	w.r.t. VS6.3.1	w.r.t. v6.4.1
4CIF Video Compression, 2bpp content is **704x576** (4:2:0 YCbCr), compressed to **2 bits/pixel**	**kdu_v_compress**	156.7	95.3	127.1	111.71	48.78	48.31	**140%**	**321%**	**324%**
4CIF Video **Decompression**, 2bpp content is **704x576** (4:2:0 YCbCr), compressed to **2 bits/pixe**l	**kdu_v_expand**	252.5	153.6	204.8	167.9	143.7	102.6	**150%**	**176%**	**246%**
4CIF Video Compression, 2bpp as above, but using **-slope**	**kdu_v_compress** (with -slope)	168.6	102.6	136.7	118.7	Note 2	56.82	**142%**	**Note 2**	**297%**
4CIF Video Compression, 2bpp as above, but using **Qstep**	**kdu_v_compress** (with Qstep)	216.1	131.4	175.3	146.9	64.1	62.5	**147%**	**337%**	**346%**

1080p Video Compression, 2bpp content is "Aspen": **1920x1080** (4:2:0 YCbCr), compressed to **2 bits/pixel**	**kdu_v_compress**	34.57	107.53	143.4	24.77	13.44	13.33	**140%**	**257%**	**259%**
1080p Video **Decompression**, 2bpp content is "Aspen": **1920x1080** (4:2:0 YCbCr), compressed to **2 bits/pixel**	**kdu_v_expand**	52.5	163.30	217.7	35.96	39.83	30.09	**146%**	**132%**	**174%**
1080p Video Compression, 2bpp as above, but using **-slope**	**kdu_v_compress** (with -slope)	36.07	112.19	149.6	25.85	Note 2	15.15	**140%**	**Note 2**	**238%**
1080p Video Compression, 2bpp as above, but using **Qstep**	**kdu_v_compress** (with Qstep)	44.45	138.26	184.3	31.67	15.38	15.38	**140%**	**289%**	**289%**

Intel i7-2720Q, **2.2GHz**, **Turbo Disabled**
Windows 7, 64-bit OS, on MacBook Pro

Test Description	**Application**	**KDU-S7.0**			**KDU-7.0**	**VS 6.3.1**	**V 6.4.1**	**KDU-S7.0 Relative Performance Advantage (relative to most relevant other**		
		FPS	Msamples/s	Mbit/s	FPS	FPS	FPS	w.r.t. KDU-7	w.r.t. VS6.3.1	w.r.t. v6.4.1
2160p Video Compression, 2bpp content is "ParkJoy": **3840x2160** (4:4:4 RGB), compressed to **1 bits/pixel**	**kdu_v_compress**	10.51	261.52	87.17	6.91	3.571	3.597	**152%**	**294%**	**292%**
2160p Video **Decompression**, 2bpp content is "Aspen": **3840x2160** (4:4:4 RGB), compressed to **1 bits/pixel**	**kdu_v_expand**	21.18	527.03	175.7	15.39	12.8	10.49	**138%**	**165%**	**202%**
2160p Video Compression, 1bpp as above, but using **-slope**	**kdu_v_compress** (with -slope)	11.1	276.20	92.07	7.19	Note 2	3.788	**154%**	**Note 2**	**293%**
2160p Video Compression, 1bpp as above, but using **Qstep**	**kdu_v_compress** (with Qstep)	17.74	441.43	147.1	12.15	5.102	4.878	**146%**	**348%**	**364%**
Average Performance Advantage of Speed-Pack version KDU-S7.0 [NB: this is just the average of all the speedup percentage values found within the relevant column; individual values vary widely for the comparison with Kakadu v6, but are always large]								**145%**	**186%**	**236%**

Note 1: On this platform, TURBO mode is normally enabled. **With TURBO enabled, throughputs are about 40% higher than quoted** for all experiments, unless the machine is running for a while, in which case the processor reduces the level of Turbo boost. For example, observed throughput for Digital Cinema tests is **31 fps (2K x 24fps content)** and **25 fps (4K x 24fps content)** -- i.e., real-time playback on a laptop computer!

Note 2: "-slope" results not obtained for speed-pack **VS6.3.1** because the **definition of "-slope" changed** between versions 6.3 and 6.4 of Kakadu

Explanation of the observed performance results

The following points should help you to understand the observations presented above:

1. The platform being used here has 4 CPU cores with hyperthreading. This means that the hardware has 8 Virtual CPU's. In order to fully exploit the processor's resources, it is necessary to create at least 8 threads of execution. This is because 2 virtual CPU's can do more work than 1 virtual CPU, even when they share the same physical CPU core. If some of the threads of execution go idle some of the time, the total throughput will of course be lower than what could potentially be achieved otherwise. However, this phenomenon is more exaggerated for processors with hyperthreading, because it can happen that both the threads that are running on one core go idle while both the threads that are running on another core remain active. Worse still, if a core goes idle for a significant amount of time, the operating system may power gate the core to save power, and recovering from this state incurs a substantial transition penalty. As a result of these phenomena, a multi-threaded application that involves substantial contention between threads, such that many threads go idle episodically, can have a throughput that actually deteriorates as the number of threads is increased to the naturally optimal value (8 in this case).
2. The phenomenon described above is exactly what happens to Kakadu versions prior to 7.0. In fact, as the number of threads grows, it becomes very difficult to totally avoid this phenomenon. At lossless compression, or very high bit-rates, the V6 and VS6 applications perform quite well with 8 threads, because each thread has a lot of work to do before it must contend for access to a critical section. At lower bit-rates, however, it can and does happen that the performance decreases somewhat as the number of threads is increased.
3. KDU-7.0, and especially Speed-Pack KDU-S7.0, contain radically new core multi-threaded sub-systems that are largely lock-less and involve relatively little thread contention. KDU-S7.0 in particular should scale well to large numbers of CPU cores, so long as the source is large enough to offer sufficient independent jobs -- although test results to reveal this scalability, however, have yet to be properly assembled and documented.
4. The core technology that differentiates regular Kakadu from the Speed-Pack releases is approximately 40-50% faster (~1.5x) when running in single-threaded mode. Between Kakadu v6.3.1 and KDU-S7.0, this core technology has been further optimized, leading to additional improvements of about 5% to 15%, depending upon the compressed bit-rate. This is most easily observed by looking at the second last column of the results, for cases where thread contention is not an issue (kdu_vex_fast with 8 independent single-threaded processing engines, or kdu_compress/kdu_expand for lossless imagery).
5. In many cases, the benefits of the Speed-Pack technology are masked by thread contention bottlenecks that affected earlier versions of Kakadu. In fact there are even cases where the accelerations in speed-pack cause the inversion phenomenon described above to set in earlier than for regular Kakadu, so that VS6.3.1 could appear to run slightly slower than V6.4.1. Reducing the number of allocated threads would correct this phenomenon but the processor would remain under-utilized.
6. The ultimate message here is that the superior threading technology is KDU-7 and KDU-S7 releases is critical to effective utilization of modern processors, where multi-core technology is becoming the main way to achieve ongoing performance scaling in a power efficient manner.