

Linux SMP Lock scalability

Some interesting data for Linux lock:

We create (1, 2, 4, 8, 16) CPU affinity threads on 16 cores system, each thread did nothing but increase a local counter, number in the table is average loop-count / sec per thread.

Global spinlock

All threads are serialized by a global spinlock

Global readlock

All threads read_lock a global rwlock

Spinlock table

Allocating a contiguous spinlock table, each thread only access it's own lock in the table. So there is no lock contention but has cacheline conflict

Global atomic

No lock, just atomic_inc the global counter

Private spinlock

Each thread has it's own spinlock

	1 thread	2 threads	4 threads	8 threads	16 threads
global spinlock	18,444,000	2,082,000	440,750	101,000	25,937
global readlock	12,189,000	2,363,000	885,250	419,750	215,937
spinlock table	18,256,000	3,629,000	1,761,000	839,375	809,250
global atomic	30,001,000	4,070,000	1,912,500	1,117,875	745,500
private spinlock	18,469,000	18,187,500	17,982,000	18,451,500	18,485,750