Concurrency & Parallelism in UE4

Tips for programming with many CPU cores

Gerke Max Preussner

max.preussner@epicgames.com

Synchronization Primitives

Atomics

Locking

Signaling

Waiting

FPlatformAtomics

- InterlockedAdd
- InterlockedCompareExchange (-Pointer)
- InterlockedDecrement (-Increment)
- InterlockedExchange (-Pointer)

64- and 128-bit overloads on supported platforms

Synchronization Primitives

- Atomics
- Locking
- Signaling
- Waiting

```
// Example

class FThreadSafeCounter
{
public:
    int32 Add( int32 Amount )
    {
        return FPlatformAtomics::InterlockedAdd(&Counter, Amount);
    }

private:
    volatile int32 Counter;
};
```

Synchronization Primitives

Atomics

Locking

Signaling

Waiting

Critical Sections

- FCriticalSection implements synchronization object
- FScopeLock for scope level locking using a critical section
- Fast if the lock is not activated

Spin Locks

- FSpinLock can be locked and unlocked
- Sleeps or spins in a loop until unlocked
- Default sleep time is 0.1 seconds

Synchronization Primitives

Atomics

Locking

Signaling

Waiting

Semaphores

- Like mutex with signaling mechanism
- Only implemented for Windows and hardly used
- API will probably change
- Use FEvent instead

Synchronization Primitives

Atomics

Locking

Signaling

Waiting

FEvent

- Blocks a thread until triggered or timed out
- Frequently used to wake up worker threads

FScopedEvent

- Wraps an FEvent that blocks on scope exit

```
// Example for scoped events
{
   FScopedEvent Event;
   DoWorkOnAnotherThread(Event.Get());

   // stalls here until other thread triggers Event
}
```

High Level Constructs

Containers

Helpers

General Thread-safety

- Most containers (TArray, TMap, etc.) are not thread-safe
- Use synchronization primitives in your own code where needed

TLockFreePointerList

- Lock free, stack based and ABA resistant
- Used by Task Graph system

TQueue

- Uses a linked list under the hood
- Lock and contention free for SPSC
- Lock free for MPSC

TDisruptor (currently not part of UE4)

- Lock free MPMC queue using a ring buffer

High Level Constructs

Containers

Helpers

FThreadSafeCounter

FThreadSingleton
- Singleton that creates an instance per thread

FMemStack
- Fast, temporary per-thread memory allocation

TLockFreeClassAllocator, TLockFreeFixedSizeAllocator
- Another fast allocator for instances of T

FThreadIdleStats
- Measures how often a thread is idle

Parallelization

Threads

Task Graph

Processes

Messaging

FRunnable

- Platform agnostic interface
- Implement Init(), Run(), Stop() and Exit() in your sub-class
- Launch with FRunnableThread::Create()
- FSingleThreadRunnable when multi-threading is disabled

FQueuedThreadPool

- Carried over from UE3 and still works the same way
- Global general purpose thread pool in GThreadPool
- Not lock free

Parallelization

Threads

Task Graph

Processes

Messaging

Game Thread

- All game code, Blueprints and UI
- UObjects are not thread-safe!

Render Thread

- Proxy objects for Materials, Primitives, etc.

Stats Thread

- Engine performance counters

Parallelization

Threads

Task Graph

Processes

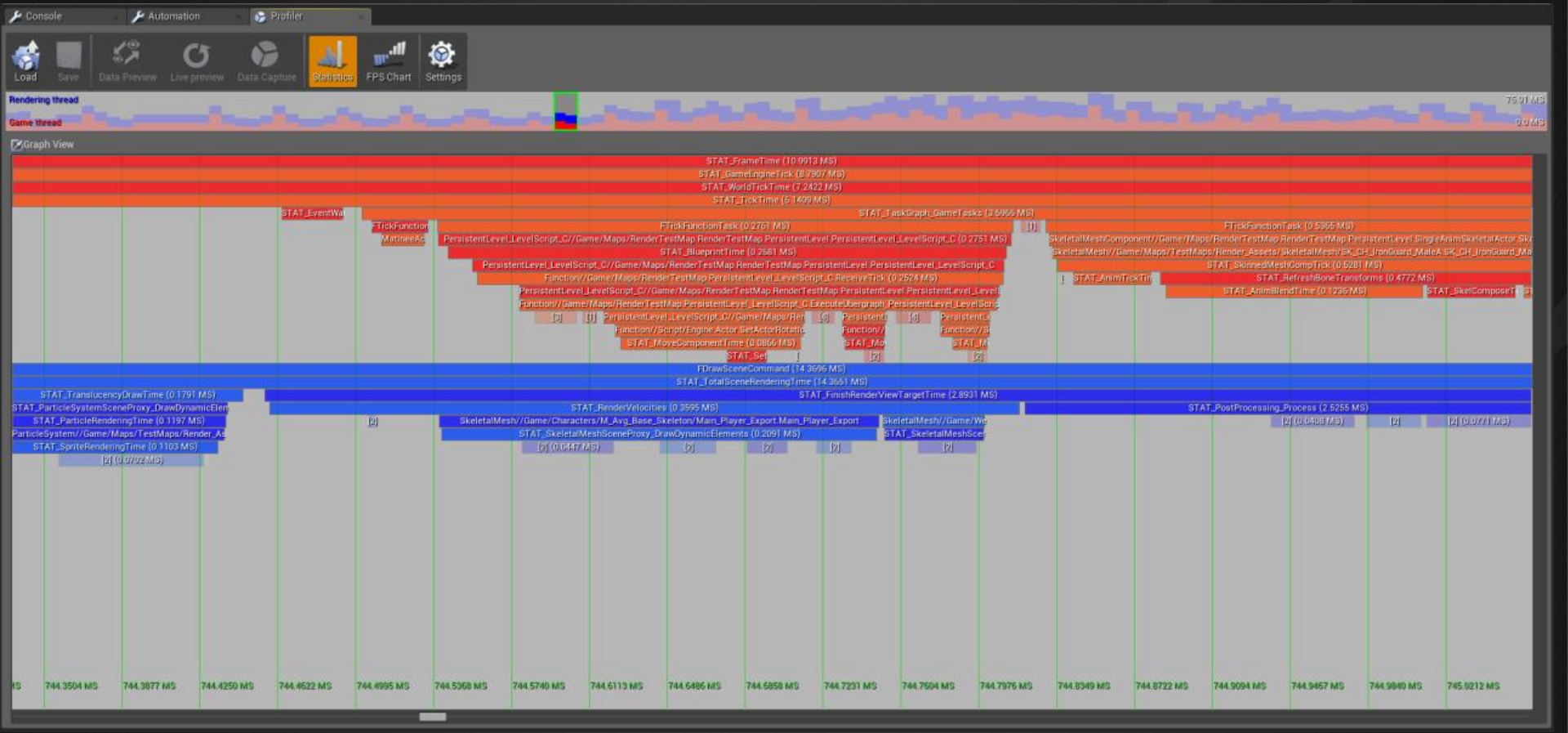
Messaging

Task Based Multi-Threading

- Small units of work are pushed to available worker threads
- Tasks can have dependencies to each other
- Task Graph will figure out order of execution

Used by an increasing number of systems

- Animation evaluation
- Message dispatch and serialization in Messaging system
- Object reachability analysis in garbage collector
- Render commands in Rendering sub-system
- Various tasks in Physics sub-system
- Defer execution to a particular thread



UNREAL ENGINE

Parallelization

Threads

Task Graph

Processes

Messaging

FPlatformProcess

- CreateProc() executes an external program
- LaunchURL() launches the default program for a URL
- IsProcRunning() checks whether a process is still running
- Plus many other utilities for process management

FMonitoredProcess

- Convenience class for launching and monitoring processes
- Event delegates for cancellation, completion and output

Parallelization

Threads

Task Graph

Processes

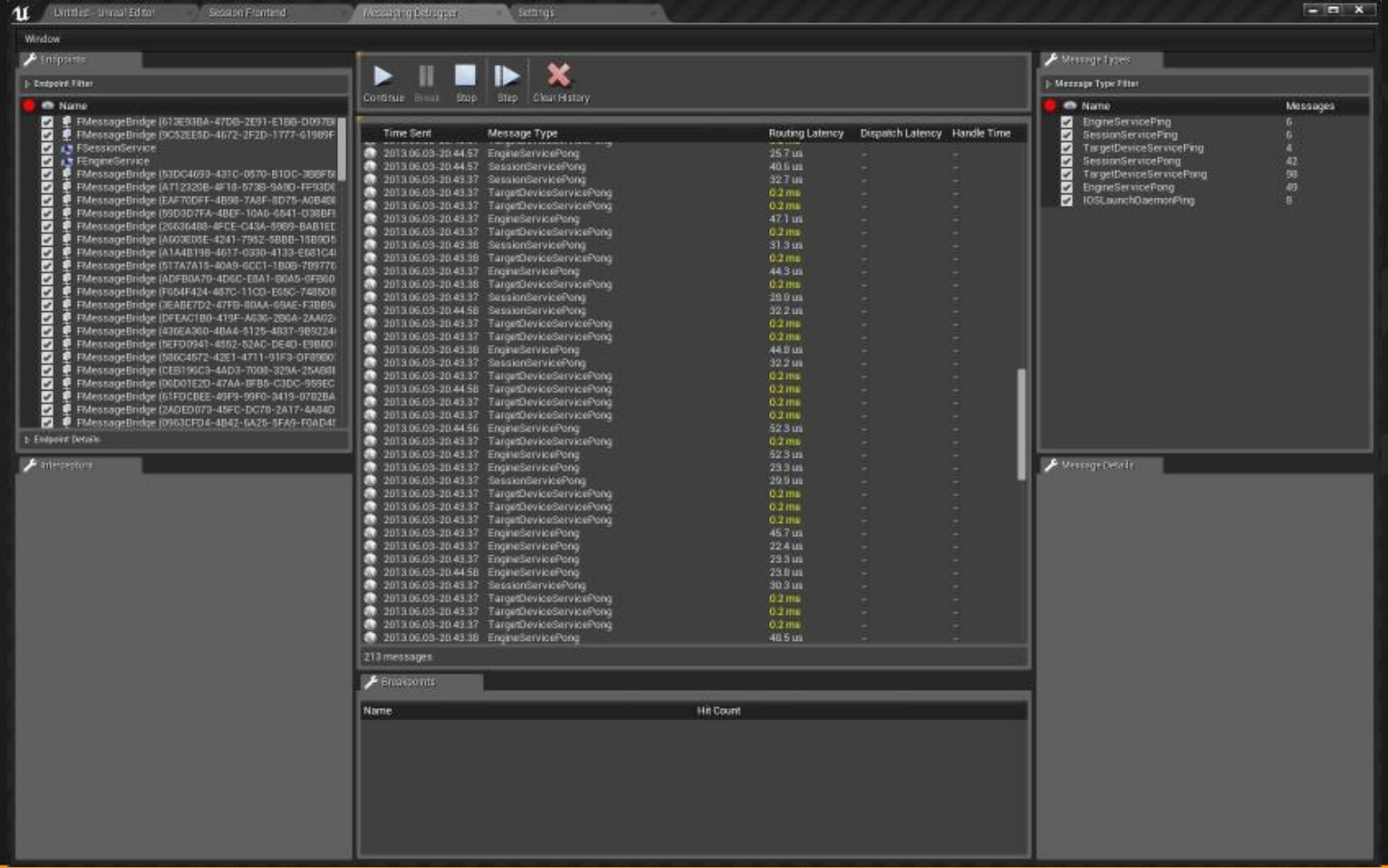
Messaging

Unreal Message Bus (UMB)

- Zero configuration intra- and inter-process communication
- Request-Reply and Publish-Subscribe patterns supported
- Messages are simple UStructs

Transport Plug-ins

- Seamlessly connect processes across machines
- Only implemented for UDP right now (prototype)



UNREAL ENGINE

Upcoming Features

Critical sections & events
- Better debugging and profiling support

Task Graph
- Improvements and optimizations

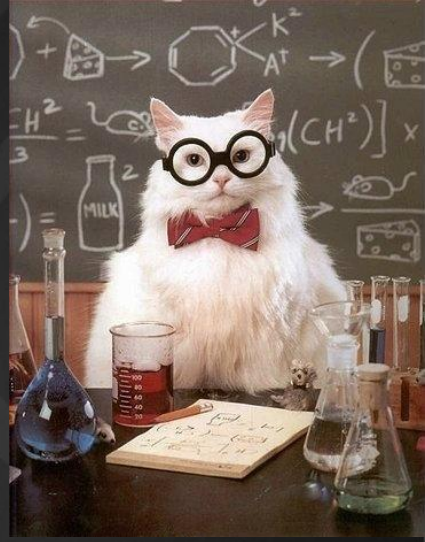
UObjects
- Thread-safe construction and destruction

Parallel Rendering
- Implemented inside renderers, not on RHI API level

Messaging
- UDP v2 (“Hammer”), BLOB attachments, more robust, breakpoints
- Named Pipes and other transport plug-ins

More lock-free containers

Questions?



Documentation, Tutorials and Help at:

- AnswerHub: http://answers.unrealengine.com
- Engine Documentation: http://docs.unrealengine.com
- Official Forums: http://forums.unrealengine.com
- Community Wiki: http://wiki.unrealengine.com
- YouTube Videos: http://www.youtube.com/user/UnrealDevelopmentKit
- Community IRC: #unrealengine on FreeNode

Unreal Engine 4 Roadmap

- lmgtfy.com/?q=Unreal+engine+Trello+