

## **Hazel McKendrick** - Project Proposal Draft Outline 2

### **Introduction**

- Concurrency prominent issue
  - Multi-core/Multi-CPU hardware commonplace (Valve 2009, Davies 2009)
  - Unable to reap benefits of Moore's Law without parallelism (Sutter 2004)
- Relevancy of distribution in particular
  - Simulations
  - Calculation (SETI@home)
  - Games servers
  - Handling entities which have separate tasks
- Load Balancing
  - Dividing virtual space over physical nodes
  - Keeping processing consistent and required computers minimal

### **Motivation**

- Problem with single servers
  - Scaling and expansion limits
  - Power consumption – cost (Atwood 2007)
  - Geographic limitations, cannot be split over locations
- Real time vs Long term
  - Render farms – hours per frame (Christensen et al. 2006)
  - Games – fast responses required
  - Frequent data transfers
  - Synchronisation, latency huge effect
- Specific problem being dealt with (characters in 2d world)
  - Represents problem in games and simulations (numerous entities)
  - demonstrate task parallelism vs data parallelism
  - Expansion and extension opportunities
  - Visualisation can clearly demonstrate problem and results

### **Literature Review**

- In Games and MMO servers
  - Different to other distributed computing – “fun”, no thin clients (Waldo 2008)
  - Parallelism in games
    - Meticulous planning, specific to game
    - (Rhalibi, Costa and England 2005)
  - Frameworks for MMOs
    - Opposite end of spectrum to specific games
    - Not explicit parallelism to user (Wollrath et al 2004)

- DarkStar (Sun Microsystems 2007)
    - task processing, not separate “zones” or ”shards”
  - Hydra – P2P (Chan et al 2007)
- P2P systems
  - Split world into zones and process over P2P nodes (Iimura, Hazeyama, and Kadobayashi 2004)
  - Scalability not limited by hardware costs for single company
  - Greater concerns in terms of access and redundancy
- Client-Server (Assiotis and Tzanov 2006)
  - Events, areas of interest for characters eg players and NPCs
    - Server can lock and modify areas of space
  - Division of world
    - Process sections on separate nodes
    - Microcells (Vleeschauwer 2005)
- Concurrency in general
  - Data vs task parallelism
  - CSP (Hoare 2004)
    - (*don't know if I will have read enough to actually include this*)
    - Approach to parallelism
    - Logical processes with their own states
    - Data sharing through messages
    - Spawned languages like Occam but relevant to most parallel systems
- Concurrency/distributed computing techniques
  - OpenMP (OpenMP 2008)
    - User directed parallelism
    - Suited to adaption of serial programs
    - Shared memory systems (so within one node, not distributed)
  - CORBA (OMG 2009)
    - Access objects in multiple languages across multiple systems & platforms
    - Not typically used for games, extensive use elsewhere
    - (Broekhuizen 2004) found it suitable for game systems
  - MPI (MPI Forum 2009)
    - API to allow communication between systems
    - Standard for HPC
    - Dictates approach to application design, ties in with CSP approach
  - .NET Remoting (Microsoft 2009)
  - Agent systems
    - Alternative approach, consider system of autonomous agents to handle data flow, task completion (Liu 2001)
- Laws concerning concurrency (for evaluation)
  - Amdahl's law (Amdahl 1967)
    - $Speed = 1 / (S + (P / N))$  where  $S = 1 - P$
    - Theoretical max speedup given % of program which must be in parallel
    - Paper highlights other problems with parallelism/distribution

- Amdahl favoured single processors
- (Hill and Marty 2008) showed relevance but changes due to modern hardware
- Gustafson's law (Gustafson 1988)
  - $Speed = N - S(N - 1)$
  - Assumes P will be scaled to suit N
  - Solve larger problem in same time rather than same problem faster with more CPUs
- Karp-Flatt Metric (Karp and Flatt, 1990)
  - Measure of parallelisation of code in parallel system
  - Diagnostic – can check for sequential/parallel bottlenecks

## **Research Question**

*How can the processing of autonomous characters in a real-time virtual environment benefit from parallelisation over multiple distributed computer systems?*

## **Methodology**

- Architecture – following from (Assiotis and Tzanov 2006)
  - Central server (and DB)
    - Oversee and distribute work
  - Multiple computing nodes
    - 1 node per machine vs 1 node per CPU core
    - LAN software cluster
    - Process and visualize
  - Relation to MMO server (in theory, not part of project)
    - connections to central server then moved to connect to nodes
- Technology
  - .NET framework
    - Clarity of threading/network support
  - Language: C#.
    - .NET makes working in multiple languages easy; can use best suited
    - Functional languages, Erlang, well suited to parallel programming (Reymont, 2005)
    - F# available, not enough time to get accustomed to it
- Networking
  - TCP-IP
  - Protocol packets may cause an issue (Esbensen, 2005)
  - Event messages occasional
    - Important these are not missed
- Concurrency
  - MPI for communications
    - Between nodes mainly
    - Also relevant between threads
  - Thread pooling for task parallelism
    - .NET thread pool available, alternative/custom one may be investigated

- Work distribution
  - Processing work split through spatial partitioning
  - Characters stored in quad-tree vs world split into microcells
  - Passed on to new nodes as they moves
- Simulation
  - Autonomous characters seen as simple state machine
  - Process each character (data parallelism) vs process each task each character wishes to perform (task parallelism)
  - Extension: consider character interaction/avoidance (data interdependency)

## **Evaluation**

- Intended deliverable
  - Server/Node applications
- Measure performance and calculating theoretical performance
  - Amdahl's Law, Gustafson's Law
  - Reference Karp-Flatt metric for continuous evaluation
- Success
  - Performance gains – possibilities:
    - Multiple nodes
    - Spatial partitioning approach
    - Data vs task parallelism
  - Or if recommendations/conclusions can be drawn
- Other considerations which might be written about but won't be implemented
  - Possible benefits from clients on other architectures (Cell, GPGPU)
  - Nodes moved from LAN to WAN
  - Use of external clients to control characters – MMO Server-like

## **Issues and Constraints**

- Not an investigation into redundancy and correctness
  - Obviously will be considered to get a working application

## **Requirements**

- Computers
  - Multiple lab computers (Windows) to test
    - Can develop on just one
  - Multi-core CPUs preferable
  - LAN
- Software
  - .NET framework, Visual Studio
  - MPI.NET install needed

## **Schedule**

## References

- Amdahl, G. M. 1967. *Validity of the single processor approach to achieving large scale computing capabilities*. AFIPS Conference Proceedings (30): 483–485. <http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf>.
- Assiotis, M. and Tzanov, V. 2006. *A distributed architecture for MMORPG*. Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support For Games (Singapore, October 30 - 31, 2006). NetGames '06. ACM, New York, NY, 4. <http://doi.acm.org/10.1145/1230040.1230067>
- Atwood, J. 2007. *When Hardware is Free, Power is Expensive* [online] <http://www.codinghorror.com/blog/archives/000868.html>
- Broekhuizen, J. and Ssekibuule, R. 2004. [online] *CORBA in a Real-Time Game Environment*. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.7313&rep=rep1&type=pdf>
- Chan, L. et al. 2007. *Hydra: a massively-multiplayer peer-to-peer architecture for the game developer*. Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support For Games, Melbourne, Australia, September 19 - 20, 2007. NetGames '07. ACM, New York, NY, pp. 37-42. <http://doi.acm.org/10.1145/1326257.1326264>
- Christensen, P. H. et al, 2006. *Ray Tracing for the Movie 'Cars'*. In: Proceedings of the IEEE Symposium on Interactive Ray Tracing 2006, Salt Lake City USA September 18-20. IEEE. pp.1-6. [online] Available from: <http://graphics.pixar.com/library/RayTracingCars/paper.pdf>.
- Davies, L and Shrouf, R. 2009. *Who moved the goal posts? The rapidly changing world of CPUs*. [online] Available from: [http://www.gamasutra.com/view/feature/4168/sponsored\\_feature\\_who\\_moved\\_the\\_.php](http://www.gamasutra.com/view/feature/4168/sponsored_feature_who_moved_the_.php)
- Esbensen, B. 2005. [online] *Online game architectures: Back-end strategies*. Available from: [http://www.gamasutra.com/gdc2005/features/20050310/esbensen\\_01.shtml](http://www.gamasutra.com/gdc2005/features/20050310/esbensen_01.shtml)
- Gustafson, J. L. 1988. *Reevaluating Amdahl's law*. Commun. ACM, 31(5), pp. 532-533. <http://doi.acm.org/10.1145/42411.42415>
- Hill, M. and Marty, M. 2008. *Amdahl's law in the multicore era*. Computer, 41(7), pp. 33-38, [http://www.cs.wisc.edu/multifacet/papers/tr1593\\_amdahl\\_multicore.pdf](http://www.cs.wisc.edu/multifacet/papers/tr1593_amdahl_multicore.pdf)
- Hoare, C. A. R., 2004. *Communicating Sequential Process*. Prentice Hall International. <http://www.usingcsp.com/>
- Iimura, T., Hazeyama, H., and Kadobayashi, Y. 2004. *Zoned federation of game servers: a peer-to-peer approach to scalable multi-player online games*. Proceedings of 3rd ACM SIGCOMM Workshop on Network and System Support For Games (Portland, Oregon, USA, August 30 - 30, 2004). NetGames '04. ACM, New York, NY, pp. 116-120 <http://doi.acm.org/10.1145/1016540.1016549>

- Karp, A. H. and Flatt, H. P. 1990. *Measuring parallel processor performance*. Commun. ACM, 33(5), pp. 539-543. <http://doi.acm.org/10.1145/78607.78614>
- Liu, J. 2001. *Autonomous agents and multi-agent systems*. World Scientific.
- MPI Forum. 2009. [online] Available from: <http://www.mpi-forum.org/>
- Microsoft. 2009. [online] *.Net remoting*. Available from: <http://msdn.microsoft.com/en-us/library/kwdt6w2k%28VS.71%29.aspx>
- OMG, 2009. [online] *CORBA*. Available from: <http://www.omg.org/gettingstarted/corbafaq.htm>
- OpenMP, 2008. [online] *OpenMP application program interface*. Available from: <http://www.openmp.org/mp-documents/spec30.pdf>
- Rhalibi, A., Costa, S. and England, D. 2005. *Game Engineering for a Multiprocessor Architecture*. Digital Games Research Conference 2005, Changing Views: Worlds in Play, June 16-20, 2005, Vancouver, British Columbia, Canada. <http://digra.org:8080/Plone/dl/db/06278.34239.pdf>
- Reymont, K. 2005. [online] *Writing low-pain massively scalable multiplayer servers*. Available from: <http://www.devmaster.net/articles/mmo-scalable-server/>
- SETI@home. [online] *SETI@home*. Available from: [http://setiathome.berkeley.edu/sah\\_about.php](http://setiathome.berkeley.edu/sah_about.php)
- Sun Microsystems 2007. [online] *Project Darkstar: Changing the game*. Available from: [http://projectdarkstar.com/w/images/3/35/Project\\_Darkstar\\_Changing\\_the\\_Game.pdf](http://projectdarkstar.com/w/images/3/35/Project_Darkstar_Changing_the_Game.pdf)
- Sutter, H. 2004. *The free lunch is over*. [online] Available from: <http://www.gotw.ca/publications/concurrency-ddj.htm>
- Valve Corporation. 2009. *Steam Hardware Survey* [online] Available from: <http://store.steampowered.com/hwsurvey/>
- Vleeschauwer, B. D. et al. 2005. *Dynamic microcell assignment for massively multiplayer online gaming*. NetGames '05: Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games, pp. 1-7, New York, NY, USA, 2005. ACM Press. <http://portal.acm.org/citation.cfm?id=1103599.1103611&coll=GUIDE&dl=GUIDE&CFID=66577707&CFTOKEN=30493528>
- Waldo, J. 2008. *Scaling in games & virtual worlds*. *ACM Queue*. 6(7), pp.10-16. <http://portal.acm.org/citation.cfm?id=1483101.1483105&coll=ACM&dl=ACM&CFID=58718348&CFTOKEN=53308178>
- Wollrath et al. 2004. [online] *Note on Distributed Computing*. Available from: <http://research.sun.com/techrep/1994/abstract-29.html>

## **Bibliography**

- Andrews, G. 2000. *Foundations of Multithreaded, Parallel and Distributed Programming*. Addison Wesley.
- Caltagirone, S., Keys, M., Schlief, B., and Willshire, M. J. 2002. *Architecture for a massively multiplayer online role playing game engine*. J. Comput. Small Coll. 18(2), pp. 105-116. <http://portal.acm.org/citation.cfm?id=771322.771339>
- Mattson, T., Sanders, B. and Massingill, B. 2004. *Patterns for parallel programming*. Addison Wesley.
- Moore, G. 1965. *Cramming more components onto integrated circuits*. Electronics, 38(8).