Empirical Performance Evaluation of the Zereal Massively Multiplayer Online Game Simulator

Amund Tveit

Division of Intelligent Systems
Department of Computer and Information Science,
Norwegian University of Science and Technology,
N-7491 Trondheim, Norway
amundt@idi.ntnu.no

Abstract. This paper presents a factorial experimental design for testing which factors controlling the scalability of the Zereal Massively Multiplayer Online Game simulator. The analysis show that the factors explain approximately 97% of the model (measured in square of errors), which is surprisingly high since the simulation is stochastic.

1 Introduction

In this paper we will investigate the scalability of the Zereal Massively Multiplayer Online Game Simulator, and how much of the response variable that can be explained by the factors (e.g. simulator parameters).

The selected experimental method is factorial design since it enables the measurement of both single impact *and* interaction effects of the parameters used to control Zereal. Selected outcome measurement is wallclock time for a fixed number of simulation steps. More details about the statistical analysis and data can be found in appendix C

1.1 Purpose of Zereal

The main purpose of the Zereal simulator is to provide a scalable (research) testbench for testing models of players, monster intelligence, and various data analysis and data mining approaches to massively multiplayer online games [2]. The alternative way of performing such research would require tight cooperation with major massively multiplayer game vendors like Microsoft, Sony, Electronic Arts, which requires extensive negotiation rounds since (real) player data is one of their greatest assets.

1.2 Goal of analysis

Get a detailed overview of how (and how much) various parameters (i.e. factors) affect the runtime of the simulation.

2 Choice of Factors and Levels

Five factors believed to have effect on the outcome of the experiment are chosen

1. number of agents of type MarkovKillers (per CPU)
2. number of agents of type PlanAgents (per CPU)
3. number of agents of type Monsters (per CPU)
4. number of CPUs
5. vision Radius of agents (for all types)

Table 1. Factor levels

Level	MarkovKillers	PlanAgents	Monsters	CPUs	Radius
1	500	500	50	10	10
2	1000	1000	200	17	20

The number of CPUs and monsters are likely to be very significant for the runtime since they are controlling the amount of movement that MarkovKillers and to lesser extent PlanAgents do. Movement between CPUs leads to (timewise costly) network traffic. Since monsters can't move, they are not likely to increase network traffic, and with computationally cheap Markov-based action-selection they are not likely to increase the CPU load much either. The effect of sight radius is uncertain, but it can potentially lead to more network traffic and more CPU load.

Interaction between factors is likely to occur, in particular between CPUs and monsters, since increasing the number of CPUs and monsters together will give more network traffic due to more doors between subworlds at the CPUs. Note that the number of agents are (initally) *per* CPU (i.e. not total number of agents. This can potentially reduce the effect of the CPUs since the information is somehow present in the factors describing the number of agents.

The selected levels presented in appendix C has been determined in discussions with the implementors (MSc students) of the simulator.

Since this is a computer simulation the selected factors can be completely controlled and set to the wanted levels. Examples of factors that can't be controlled is the load on the computer cluster, but the batch job scheduler minimizes the risk of high load from other applications or systems simultaneous with our experiments.

Choice of Response Variable The selected response variable is the (wall clock) time for running simulations of 100 cycles (i.e. simulating 1 second per cycle) with the variations of levels shown in 1.

Other response variables of interest could be the load of CPUs of the computer systems or type of actions performed by the agents. These are however more computationally expensive to measure than the wallclock time.

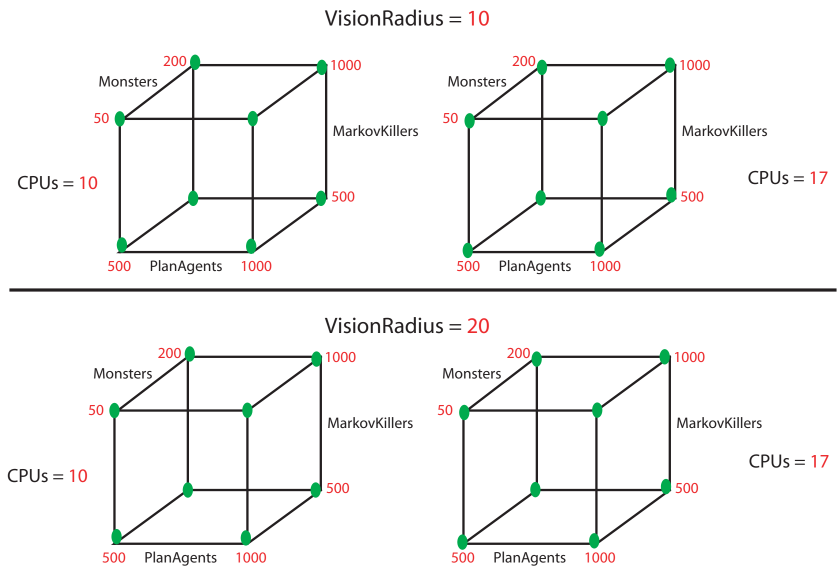

Fig. 1. 2^5 Factorial Combinations

Wall clock time response is measured using the unix shell method *time*. The accuracy of *time* is in milliseconds.

Choice of Experimental Design Selected experimental design is a factorial design with two levels [1]. The design is shown in figure 1.

The experiment is complete factorial (full resolution). There is no need to create this design with blocks.

Experimentation The order of runs, one repeat of 2^5 combinations = 32 runs, are fully randomized using a uniform distribution. The purpose of the randomization is to lower the potential effect on load caused by non-simulator processes on the cluster.

3 Analysis

3.1 Variance Model

The wall clock time T_{ijklm} is assumed to be a function of the parameters in the following way:

- MarkovKillers: α_i
- PlanAgents: β_j

– Monsters: γ_k
– CPUs: δ_l
– Radius: ϵ_m

$$
\begin{aligned}
T_{ijklm} = {} & \mu + \alpha_i + \beta_j + \gamma_k + \delta_l + \epsilon_m \\
& (\alpha\beta)_{ij} + (\alpha\gamma)_{ik} + (\alpha\delta)_{il} + (\alpha\epsilon)_{im} \\
& (\beta\gamma)_{jk} + (\beta\delta)_{jl} + (\beta\epsilon)_{jm} + (\gamma\delta)_{kl} + \\
& (\gamma\epsilon)_{lm} + (\delta\epsilon)_{lm} + \zeta_{ijklm}
\end{aligned}
$$

The error ζ_{ijklm} come from stochastic elements of the experiment (e.g. cpu load) in addition to higher order interaction effects.

Calculations was performed using Minitab's General Linear Model (output in section 6.2). By selecting a confidence level of 5%[1], the following factors have significant effect (checking the P-value):

1. MarkovKillers
2. PlanAgents
3. Monsters
4. CPUs
5. Radius
6. MarkovKillers*Radius
7. PlanAgents*Radius
8. Monsters*Radius

The main factors were likely to have effect, but that 2-factor interactions involving the vision radius are not directly obvious.

The significant main and interaction factors together explain approximately 97.6% of the model based on calculations with Sum of Squares, the exact expression is (1.0 - 206214/211197)*100. The rest of approximately 2.4% is due to true variance from stochastic elements in the experiments.

A plausible explanation of the high main effect of vision radius (fig 1) is that the amount of vision processing increases proportional to the square of the vision radius (processing area = $\pi \cdot (visionradius)^2$). Since adding more agents *and* increasing the vision radius rapidly increases the load of an agents vision algorithm (i.e. sees a larger area with higher density of other agents), the significant 2-factor interactions involving agents and vision radius (fig 1) makes sense.

The main effects satisfy the initial assumptions, but the 2-factor interactions involving vision radius were quite surprising at first.

There is no need to perform more experiments in order to deal with alias structures since the experiment is a based on full factorial design (i.e. disabling alias structures).

[1] not the same α as in the model above!

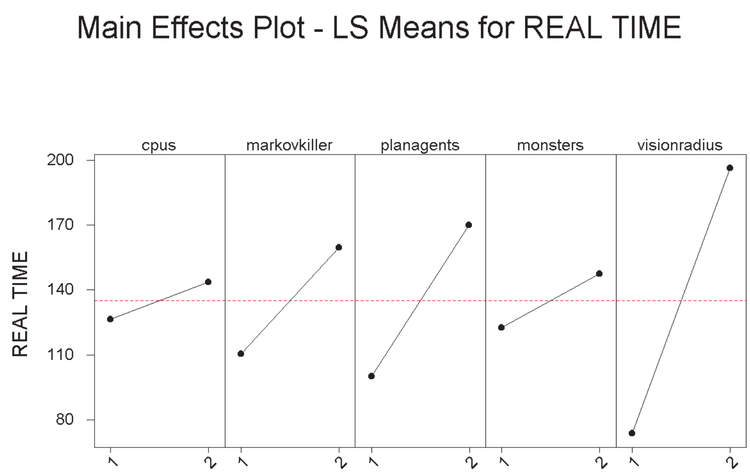

Fig. 2. Main Effects Plot

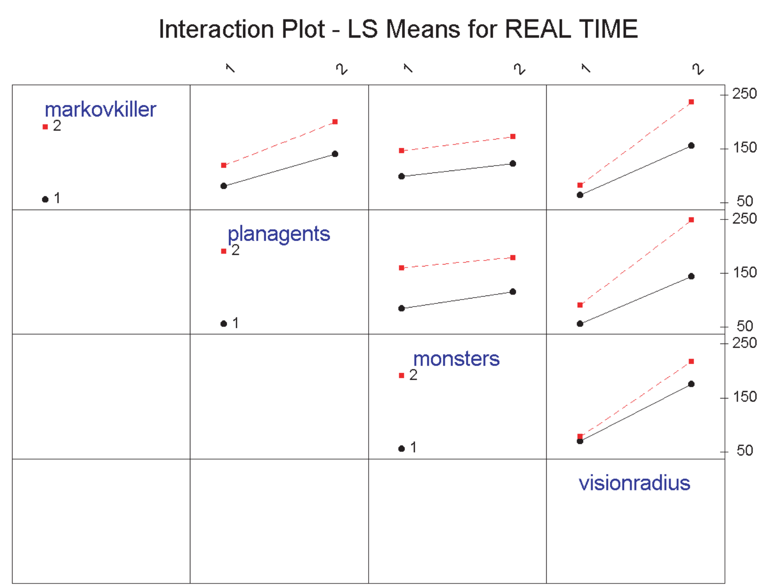

Fig. 3. Interaction Plots

3.2 Scalability of Zereal

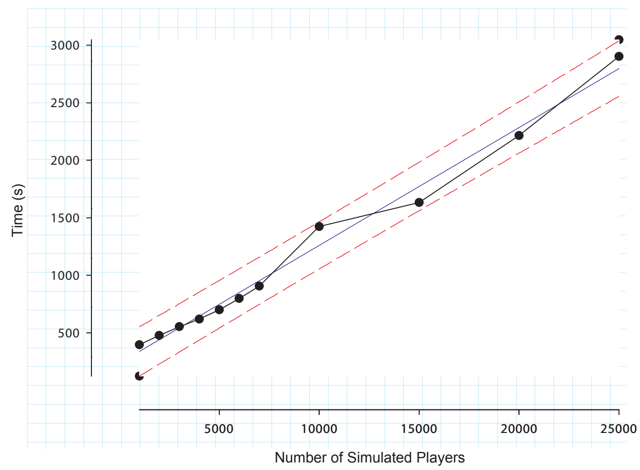

Fig. 4. Zereal scalability (with regression line and prediction interval)

In figure 1 the wallclock runtime performance for 5 minutes simulated time in the game (i.e. 300 simulated seconds) as a function of the number of simulated players. The simulation is performed on 5 Athlon 1.6 GHz CPUs on an Linux-cluster. The largest simulation performed so far with Zereal is with 160 thousand agents (simulated players and NPCs) on 20 CPUs. The setup for the experiment was with simulated players and NPCs with Markov chain type reasoning (the largest number of hierarchical planning agents tested so far is 50000). By doubling the number of players and keeping the number of simulated cycles fixed the number of sense-reason-act cycles to be performed doubles, this can explain the close-to-linear scalability.

4 Conclusion and Recommendations

It has been determined that there is significant interaction between factors in the experiments (MarkovKillers*Radius, PlanAgents*Radius and Monsters*Radius)

The main factors and including all interactions between them explain approximately 97% of the response variable, this is a surprisingly high value since the simulator has highly stochastic behavior of killers and monsters. However, since each run is over a period of 100 cycles, and for each cycle the killers (and monsters) perform one action (or inaction), one gets an "averaging" effect on the measured wallclock time.

In future experiments it would be useful to track the movements of agents between CPUs, and measure how the agents are balanced between CPUs, another factor that also would be interesting to add is the load of the cluster operating system (explaining some of ζ_{ijklm}), and a third interesting factor is topology of the simulated game world and how it is mapped onto the CPUs.

4.1 Scalability of the Zereal Platform

We have used factorial experimental design and identified eight factors that have effect on the **scalability** of the simulator. These significant factors are:

1. MarkovKillers
2. PlanAgents
3. Monsters
4. CPUs
5. Radius
6. MarkovKillers*Radius
7. PlanAgents*Radius
8. Monsters*Radius

References

1. Douglas C. Montgomery. *Design and Analysis of Experiments*, chapter 10, pages 461–466. John Wiley & Sons, Inc., 4th edition, 1997.
2. Amund Tveit, Øyvind Rein, Jørgen Vinne Iversen, and Mihhail Matskin. Scalable Agent-based Simulation of Players in Massively Multiplayer Online Games. In *Proceedings of the 8th Scandinavian Conference on Artificial Intelligence*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2003.