Enhanced Matchmaking in Multiplayer Game using Expectation-Maximization Algorithm with Multiple Cluster Player Pool

[1]Jayashree. R, [2]Venkata Subramanian. J, [3]G.Babu, [4]B.Prakash

[1,2]Faculty of Science and Humanities,
[3,4]Directorate of Online Education, Dept of Computing Technology,
SRM Institute of Science and Technology, Kattankulathur, Chennai, India.
jayashrr@srmist.edu.in, venkatj@srmist.edu.in, babug2@srmist.edu.in, prakashb@srmist.edu.in

Abstract:

The Multiplayer online game industry is growing tremendously. The matchmaking procedure is a very complex task which determines the success and satisfaction of the players. The player's potential is determined by his/her win count and game rating. A team with equivalent potential players makes the game more interesting. Equal skill score competitors have a probability of a 50% success rate.Principal cloud service provider enables online games to gain efficacy and popularity with serverless technology. The expectation-Maximization Clustering approach categorizes the players into different levels depending on their ability. Matchmaking among players in the same cluster enhances the quality of multiplayer games. The 0.5 million sample dataset from Kaggle shows only a 2% difference between players who dropped and won the game. In this paper, a matchmaking method is proposed for multiplayer online 5v5 games with minimum drop-risk using Expectation-Maximization clustering on the player pool.

Keywords: *Expectation-Maximization algorithm, Online Game Matchmaking, Clustering, Server-less technology.*

DOI: 10.48047/ECB/2023.12.SI8.585

INTRODUCTION

Games that are played over the internet instead of playing on a local network or a single machine are called online multi-player games. Online multiplayer games are not just a widespread form of entertainment but an appealing business with millions of concurrent players [10]. For example, the "Counter-Strike: Global Offensive" game contains 400,00 concurrent players. In 2016, worldwide online games were estimated worth $36 billion in the game market [11]. This game market consists of different types of online games, such as fighting, puzzle, action, adventure, and simulation [12 – 15]. Though the significance of the games differs, in most multi-player games one team of players play against the other team [2]. Considering the features of multiplayer games, it is possible to increase players' satisfaction and playtime by enhancing the game quality and allowing players to interact with each other's [3]. The skill matching among the players is a vital factor for players' satisfaction. Matchmaking without a game server provides scalability with elasticity to handle unexpected demands [19]. This serverless architecture manages infrastructure in various aspects automatically [19]. The most crucial factors in the online game are waiting time, Matching accuracy, and Response time [8].

Though the average waiting time is less than 90 seconds, waiting time makes the experienced players scarce and restless. This waiting time makes new players feel helpless [8]. Matchmaking algorithms make matching from the pool of players according to their arrival time [25]. The

Expectation-Maximization (EM) algorithm identifies parameters' maximum likelihood approximations even with missing or unobserved data [22]. In the anticipated model, the players' pool is categorized depending on their play skills in similar types of games like players' rating and win rate using the EM algorithm. Drop-risk is one of the major attributes that influence game rating. The objective is to maximize the probability function. In this paper, a Matchmaking with Multiple Cluster in Player Pool (MMCPP) algorithm is proposed to reduce the drop-risk of a player and a team in 5v5 games.

1. LITERATURE REVIEW

Multiplayer online game is a giant business that is extending day by day. In 2015, the number of online games is 2 billion and currently, almost 3 billion games are existing [7]. Dhupelia et al. provide a complete lobby construction with properties of modern online games [16]. Multi-player games using listen, server model that is without paying for dedicated servers face 2 challenges [4]. They are cheating and lagging. Zander et al. [9, 18] explain the importance of improving the matchmaking method by including additional attributes. In the matchmaking system, one top-level player is matched with another top-ranked player. Thus, top-ranked players are treated as skilled though the new player is much more skilled than the top-ranked existing players, who incorrectly remain at the top of the rank [6]. In [1] the author recommends a method to reduce random matchmaking time with queues. Matching the potential player with non-potential player results in a high drop-risk as [8]:

- Unskilled players are uncomfortable with or against much stronger partners or opponents.
- Potential players may spend time on an unchallenging session.

In general, multi-player games organize players according to their skills [3]. Some games need additional complex attributes for matchmaking such as number of kills, number of deaths, latency, region and style of play [3]. EM algorithm simultaneously optimizes a large number of variables and identifies better estimates with missing information in the dataset [23, 24]. Find the combination of N Gaussians to model observed data where each Gaussian is represented as in (1)

$$G(x) = \left(\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_i|^{1/2}}\right) e^{-\frac{1}{2}y^T \Sigma^{-1} y} \quad (1)$$

Where $|\Sigma_i|$ is the determinant of the (n × n) covariance matrix Σ_i for the ith Gaussian, $\mu_i$the meanvalue and $y=x- \mu_i$.Let pi is theprior probability with the ith Gaussian concerning its contribution to the combination N [23]. Cluster the interested players into disjoint clusters where each cluster corresponds to a Gaussian. Rapid, seamless, scalable and reliable are the required features in any multiplayer game to support 3 billion players [19]. The minimum expectation of game participants is to gain the best gaming experience with a high-quality matchmaking model.Various algorithms with any variables match the players in a server-less matchmaking system [19].Matchmaking is the opening point for connecting multiple players to online game servers [4]. Online game matchmaking consists of [4]:

- Set of players interested toplay
- Clustering players into teams using EM algorithms
- Minimizing players’ waitingtime
- Providing backend updates of game server to the players withabstraction

Additionally, features such as session details, back-fill, join-in-progress, and statisticalinformation.

Building a matchmaking logic needs 3 key variables [1],

- Best skillmatch
- Bestlatency
- Best queueingtime

Apart from the overall skill level of all the players, the expertise in that game is considered to get the best matchmaking system. The matching distance of the players is identified in terms of latency. Queueing time specifies the waiting time of a player before joining the game. Additionally, in the online game, the following criteria are considered [1],

- To allow or deny a player in a game session
- Players joining the session after it has started
- Game sessions cancel automatically after a fixed time limit when the game takes more loading time than the fixed time limit.

Cloud service provides such as Amazon web service, Microsoft Azure, and Google are providing server-less platform for online multi-player games [19]. The server-less computing enables developers to develop and execute real-time analytics easily on various data streams. Moreover, there is no need to manage the infrastructure [19].

Overall rating of a player is the major factor in determining skill level in matchmaking [1].Online game players are clustered based on this matching profile through the gamebackend.

2. MULTIPLE CLUSTER MATCHMAKING ONLINE GAME PLAYERS

Construct a graph G for the set of players interested to play a 5v5 multi-player online game. Let v_i is a vertex to represent the player ‘i’ in G at player level s_i. The player level is determined by skill level, win rate, loss rate and dropped from the game record (drop-risk). Players v_i, i=1,.., 5, are associated when an edge connects both the players and the sum of the objective metric is the total weight of both the players. This weight value depends on players' skill level and winrate and is represented as $w(s_i)$.

Let G is a complete graph where players can be connected with multiple-cluster according to skill level.Participants list L = {v_1, ..,v_n }, where n=5 in a 5v5 game, denote a matchmaking result in which every players are connected in 5V5 game. Note that participants are connected only once.Graph G represents a connection of players from the multiple cluster pools, $P_{1,2,...n}$. The goal is to find an optimal connection assignment with minimal drop-risk as in (2) :

$$L^* = \text{argMaxL}\sum_{i=1}^{n} v_i € \text{ L } * w(s_i) \quad (2)$$

Drop-risk can indicate the probability of a player not playing any games within a periodor quitting a game before completing it. From (3), L* is equivalent to minimizing the sum of drop-risks of the connected players. Thus,the optimization objective function is represented as:

$$L^* = \text{argMinL}\sum_{i=1}^{n} v_i € \text{ L}[d(s_i)] \quad (3)$$

Where $d(s_i)$ is the drop-risk of player i.

Denoting player v_i's skill level in clusterki using (1) the probability,P, of the game result$R_i$between players$v_{1,..,5}$can be represented as :

$$P(Ri,j \mid si,sj) = P(Ri,j \mid \mu 1 \mu j) \quad (4)$$

When i=1, j=(i+1),..,n. The drop-risk of user's combination is predicted using (3) and (4) as given,

$$d(s_i, s_j) + d(s_j, s_i) = \sum P(R_{i,j} \mid s_i,s_j)(d(s_i \mid R_{i,j}) + d(s_i \mid R_{j,i})) \quad (5)$$

The construct $d(s_i|R_{i,j}$) may represent the drop-risk of player v_i after matchmaking, where the conditional independence of di on s_i given R_i is used. Graph G is generated for players within a cluster where skill score and win rate are represented as edge weight based on the selected objective function as depicted in Fig. 1. The selected players omitted in all clusters are known as outliers. In such cases, players associated with outliers are included in another cluster with alternative players in the subsequent performance of the process. The users may be placed back in a pool of players depending on the skill rate awaiting assignment to an instance of the game. Thus, each node in the graph is associated with a selected edge.

3. EXPERIMENTS AND RESULTS

The players' pool is clustered based on player skill score and matched as per request classification to a respective game session which is depicted in Algorithm.1 with the following taxonomies.

PD-Player Device
TM – Traffic Manager
SK-Player skill score
S1- server
MPP- Matchmaking Player pool
DB- Database
GS - Game Server
GN- Game Session

Algorithm.1:MMCPP

```
# PDs are client
# Initially all servers are available.
Available(S_1) =True
Connect PD and TM
#Add PD to MatchmakingQueue
#n is the number of playerswaiting in the queue
Matchmaking_Queue[n+1]= PD
# Connect TM to Regional Zone with Minimum Latency
          TM ->Min_Latency(Regional_Zone)
          Request ->Event_Hub
          PP =EM_clustering(SK)
              Add PP in MPP
                 #Connect PD to a GS
                        PD ->Game_Server
                      #Add PD to Database
                      DB[n+1] =PD
                   If no game then
create new session
                   Else
#AddPD in a game session
# m is the number of Players  in GN
                   GN(m+1)=PD
#Find any session with 5 PD to start a 5v5 #game
Find S1 in database
Fetch IP:Port
Available(S1)=False
Session_Timer=True
       Event_Hub=List[PD_i, S_1]  # i=1 to 5
```

PD receives S_1(IP:Port)
#PD connect to GS
PD -> GS
#Delete PD from Matchmaking Queue
Delete(Matchmaking_Queue[PD])

In Alogrithm.1, players' devices are the client machines and initially, all servers are available for connection. First, connect the PD to the traffic manager and add this PD to the queue to perform matchmaking. This queue is a waiting queue for all PDs. Next, find the regional zone with minimum latency and connect the traffic manager in that zone. The request from PD is handover to the event hub which contains a list of servers that are connected with PDs. Once the PD request reaches the event hub, apply the EM algorithm the cluster all PDs in the waiting queue according to players' skill scores. Clusters together represented as player pool, PP. Connect PD from PP to the game server and update the PD status in the database. If there is no new game then a new session is created, otherwise, add PD in any of the game sessions. In a 5v5 game, the maximum number of PD required is 5, thus game session starts once 5 PDs with similar SK requests are available. Finally, fetch a server, S1 with port and Internet Protocol (IP) address using cloud service provider's server-less model for auto-scaling and abundant playtime. Start the session timer and update the event hub with respective server and PDs detail. Connect PD to the game server and delete this PD from the waiting queue.

Dataset collected from Kaggle [21] contains 474417 online games on over 50 platforms including mobiles with 27 attributes. The attributes of overall rating and potential are significant in game player matchmaking. Players' skill levels and win rates are identified and clustered into teams. The Hit-Ratio (HRatio) and Average Precision (AP) are used as evaluation metrics for game matchmaking using the EM algorithm with multiple cluster player pools as shown in figure 1, when HRatio and AP are inversely proportional to the drop-risk of a player.

The HRatio and AP of rating-based matchmaking systems for online games produce almost less than 10% and less than 50% for the number of iterations respectively. EM algorithm-based Multiple-cluster matchmaking system provides HRatio and AP are above 50% and 100%.

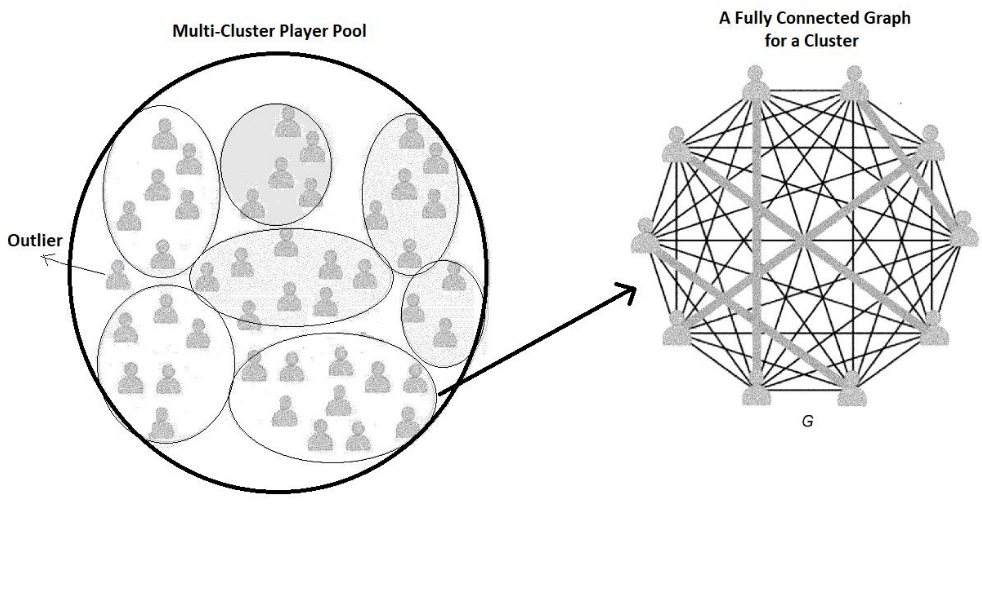

Figure. 1. Multiple cluster player pool is mapped to a fully connected graph for a single clusterG

4. CONCLUSION

Online game is the major entertainment-oriented business. Effective matchmaking for players into a team and also as an opposed team is very important to engage users in gameplay. Players' pool is clustered based on their skill level and win rates using the EM algorithm with no server management and auto-scaling. Matchmaking among players of the same cluster reduces the drop risk and makes the game more interesting.

REFERENCES

[1] Joannaleecy. (n.d.). Azure gaming documentation - Azure Gaming. Microsoft Learn. https://docs.microsoft.com/en-us/gaming/azure/(accessed Aug. 31, 2022).
[2] Boroń, M., Brzeziński, J., &Kobusińska, A. (2020). P2P matchmaking solution for online games. Peer-to-peer Networking and Applications, 13(1), 137–150. https://doi.org/10.1007/s12083-019-00725-3
[3] Zamzami, E. M., Tarigan, J. T., Jaya, I., &Hardi, S. M. (2018). Openlobby: an open game server for lobby and matchmaking. Journal of Physics. https://doi.org/10.1088/1742-6596/978/1/012069
[4] Heddings, A. (2020). How to Build Your Multiplayer Game's Server Architecture. How-To Geek. https://www.cloudsavvyit.com/2586/how-to-build-your-multiplayer-games-server-architecture/(accessed Aug. 31, 2022).
[5] Da Silva, L. C. L. C., Costa, K. a. P., Papa, J. P., De Rosa, G. H., & Gupta, D. (2018). Fine-tuning restricted Boltzmann machines using quaternions and its application for spam detection. IET Networks. https://doi.org/10.1049/iet-net.2018.5172
[6] Halite by Two Sigma | Kaggle. (n.d.). https://www.kaggle.com/c/halite/discussion/168495(accessed Aug. 31, 2022).
[7] Dataset Search. (n.d.). Google Dataset Search. https://datasetsearch.research.google.com/
[8] MaximeVéron - Home. (n.d.). Author DO Series. https://dl.acm.org/profile/82259354057(accessed Aug. 31, 2022).
[9] Proceedings of Network and Operating System Support on Digital Audio and Video Workshop. (2014). https://doi.org/10.1145/2597176.
[10] Boroń, M., Brzeziński, J., &Kobusińska, A. (2020). P2P matchmaking solution for online games. Peer-to-peer Networking and Applications, 13(1), 137–150. https://doi.org/10.1007/s12083-019-00725-3
[11] Number of Steam users 2021 | Statista. (2022, November 24). Statista. https://www.statista.com/statistics/308330/(accessed Aug. 31, 2022).
[12] Research, V. M. (2022, May 23). PC Games Market Size Worth USD 31.52 Billion by 2028 | PC Games Industry Expected CAGR 1.3% | Vantage Market Research. GlobeNewswire News Room. https://www.globenewswire.com/en/news-release/2022/05/23/2448225/0/en/PC-Games-Market-Size-Worth-USD-31-52-Billion-by-2028-PC-Games-Industry-Expected-CAGR-1-3-Vantage-Market-Research.html
[13] Apperley, T. (2006). Genre and game studies: Toward a critical approach to video game genres. Simulation & Gaming, 37(1), 6–23. https://doi.org/10.1177/1046878105282278
[14] Ahmad, A., & Law, E. L. (2021). Educators as Gamemasters. Proceedings of the ACM on Human-computer Interaction, 5(CHI PLAY), 1–29. https://doi.org/10.1145/3474657
[15] Hrabec, O., &Chrz, V. (2015). Flow Genres. International Journal of Gaming and Computer-mediated Simulations. https://doi.org/10.4018/ijgcms.2015010101.
[16] Dorland, M. (2004). Digital Play: The Interaction of Technology, Culture, and Marketing (review). University of Toronto Quarterly, 74(1), 378–380. https://doi.org/10.1353/utq.2005.0031
[17] Kim, M. S., & Lee, S. H. (2020). A Case Study on the Development of Design and Programming Convergence Courses for Game Production. Design Research, 5(4), 72–81. https://doi.org/10.46248/kidrs.2020.4.72.
[18] Kinicki, J., & Claypool, M. (2008). Traffic analysis of avatars in Second Life. https://doi.org/10.1145/1496046.1496063

[19] Fitting the Pattern: Serverless Custom Matchmaking with Amazon GameLift | Amazon Web Services. (2022, May 23). Amazon Web Services. https://aws.amazon.com/blogs/gametech/fitting-the-pattern-serverless-custom-matchmaking-with-amazon-gamelift/
[20] BrianPeek. (2022, September 29). Multiplayer Serverless Matchmaker - Azure Gaming. Microsoft Learn. https://docs.microsoft.com/en-us/gaming/azure/reference-architectures/multiplayer-matchmaker-serverless(accessed Dec 31, 2022).
[21] Video Game Dataset. (2020, December 23). Kaggle. https://www.kaggle.com/datasets/jummyegg/rawg-game-dataset?resource=download.
[22] Columbia University in the City of New York. (n.d.). http://www.columbia.edu/
[23] Tutorials by AviKak. (n.d.). https://engineering.purdue.edu/kak/Tutorials/
[24] Index of /~filkov/classes/234-S08/secure. (n.d.). https://www.cs.ucdavis.edu/~filkov/classes/234-S08/secure
[25] Jayashree R, Vaithyasubramanian S (2021) Restricted Boltzmann Machine-Driven Matchmaking Algorithm With Interactive Estimation of Distribution for Websites. In Handbook of Research on Innovations and Applications of AI, IoT, and Cognitive Technologies. IGI Global. pp. 258-276.