Concurrent Poker Player

Milestone One

The Parallel Poker Team
Mark Jenne
Ian Roberts
Bennie Waters
Sarah Jabon

October 2nd, 2009

Instuctors:
Sriram Mohan
Rose-Hulman Institute of Technology

Revision History

Date	Revision	Comment
09-23-09	1.0	Initial draft
09-24-09	1.1	Revised after project manager's feedback
09-25-09	1.2	Revised for grammar and formatting
10-02-09	1.3	Revised after Sriram's and Boby's feedback
10-03-09	1.4	Revised after Sriram's final comments
10-08-09	1.5	Revised after Boby's final comments

Signatures

Mark Jenne	
Ian Roberts	
Bennie Waters	
Sarah Jabon	

Contents

Part I

Executive Summary

The parallel poker player project involves designing and developing an application that exhibits the benefits of parallel computing [1] over sequential computing [2]. The application will showcase this through a demonstration of different computer-based players involved in a Texas Hold 'Em poker game [12]. The two key elements discussed in this document are the user information and a product overview. The user information includes:

- Client Background

- Current System

- System Users and Their Needs

The second key element is a product overview, which describes the features and constraints of the project:

- Product Perspective

- System Features

- System Constraints

The poker application will be a visually stimulating demonstration of the differences between parallel [1] and sequential computing [2].

Part II
Introduction

Milestone 1 is the first documentation of the poker application's requirements and specifications. The features described in this document will be the basis of the project's use cases, functional requirement descriptions, and supplementary specifications. The use cases and supplementary specifications, in turn, will help the team develop test cases. Based on the test cases, the team will develop an interaction prototype, which will be completed by November 2009. The design of the actual application will be determined by the use cases, supplementary specifications, and the interactive prototype. Once the design is established, the team will create a prototype. This prototype will be fully completed by February 2010. Finally, both the prototype and the incorporated design will be used to create the complete system. The system will be complete by the end of the academic year. So, the objectives of this document are to concisely summarize the established requirements of the poker application and clearly state how the application will be utilized. With this core information, it will be possible to develop aspects of the project in fuller detail in the future.

Part III

User Information

1 Client Background

Our client, Boby George, is a member of the Concurrency Development Platform (CDP) team at Microsoft Corporation [6]. All associates at Microsoft [6] that are involved in the development process have one of three primary engineering roles:

- Project Manager - manages the project features

- Tester - tests the projects for functionality and usability

- Developer - programs the actual features and other software components

Both the client himself, Boby George, and his fellow associate, Sherif Mahmoud, are part of the test phase for the CDP team's newest application programming interface (API) [10]. Sherif oversees the department for the testing of this particular project. The testing phase is an essential element of the development cycle as testing ensures the quality and usability of the software.

The CDP API [10] is expected to premier with the release of Visual Studio 2010 [7] in the first quarter next year. However, the Beta II version will be available in October of 2009.

Boby George is more specifically in charge of the testing of the concurrent data structures to be included with the release of the API [10]. These data structures will allow for more efficient use of memory and processor time when a programmer makes calls to the API [10].

2 Current System

The current system for implementing parallel applications involves developing parallel constructs using rudimentary support, such as threads, provided by today's mainstream programming languages. Programming language libraries generally provide the tools required to create multiple threads and the ability to communicate between such threads. However, this requires an understanding of how to develop efficient parallel functions while taking the risks of using multiple threads into account. Another issue is automatic scaling. When an application that was written for two cores is loaded into a machine with more than two cores, problems with the efficiency of the application arise.

One instance of the difficulty is Bethesda's [3] game, Fallout 3. Fallout 3 is a role-playing game (RPG) based in a post-apocalyptic Washington D.C. and the surrounding area, the game features a modern graphics and simulation engine. Bethesda [3] failed to make their artificial intelligence (AI) [11] for Fallout 3 scalable to more than two cores. Microsoft's [6] CDP API [10] and libraries will help programmers solve this issue. It will make certain that the multiple cores are utilized, and that the program remains stable.

3   System Users and Their Needs

There are three main types of users that will work with the poker application:

1. A Microsoft [6] associate that is on the parallel computing [1] team

2. A Microsoft [6] associate that is unfamiliar with the parallel computing [1] team

3. A potential Microsoft [6] customer that is interested in learning about Microsoft's [6] products

All three users have different backgrounds and needs. Microsoft [6] associates that are familiar with parallel computing [1] could be developers, testers, or project managers. They would expect to see a clear demonstration of the key differences between parallel [1] and sequential computing [2]. Since they are familiar with the efficiency of parallel computing [1], they would desire the application to accurately display the benefits over sequential computing [2]. They would also assess the application for its consumer appeal. Their definition of success would be seeing an application that could convince potential customers that parallel computing [1] is applicable to their work and that they should use the CDP API [10]. Therefore, they would expect the application to be visually appealing.

Table 3.1 - User Needs: Potentials of Parallel Computing

Element	Description
The problem of...	customers not realizing the potentials of both the CDP API [10] and parallel computing [1] as a whole
Affects...	potential Microsoft [6] customers, current Microsoft [6] customers, Microsoft [6] shareholders, Microsoft [6] associates, and competing software companies
And results in...	potential customers not using Microsoft's [6] API [10]. If they do not see how effective Microsoft's [6] tools are, they have no reason to use them. If customers are not buying Microsoft's [6] products, then the shareholders will be affected by lower stock prices. Also, Microsoft [6] associates will be affected by how much revenue is coming into their company. Finally, competing software companies may receive more business that would have gone to Microsoft [6].
Benefits of a solution include...	Microsoft [6] acquiring customers if the application impresses them. The poker application is a great way to showcase how efficient parallel computing [1] is. Microsoft [6] will also be showing customers the API [10]. If the company gains more customers, then associates and shareholders of Microsoft [6] will benefit. Finally, competing companies may lose business to Microsoft [6].

The Microsoft [6] associates on the CDP team would also want to improve the API [10], which involves receiving feedback from new users. The poker application provides a great opportunity for programmers with no experience in parallel computing [1] to provide suggestions and comments.

Table 3.2 - User Needs: Improving Microsoft's [6] API

Element	Description
The problem of...	Microsoft [6] needing to improve their API [10]
Affects...	potential Microsoft [6] customers, current Microsoft [6] customers, and competing software companies
And results in...	unhappy Microsoft [6] customers. If Microsoft [6] does not develop their API [10] to the greatest extent possible, their customers may choose competitors' products.
Benefits of a solution include...	Microsoft [6] improving their API [10] with a good understanding of what needs to be changed. The solution is to have programmers who are inexperienced with parallel computing [1] test out the API [10] and provide feedback about it. The poker application will be a great opportunity to acquire this kind of feedback.

The Microsoft [6] associates that are unfamiliar with the CDP team will have a different background. They may be in any department at Microsoft [6], so there is no guarantee that they would be experienced with programming at all. They would be excited to see the capabilities of parallel computing [1] and learn about what the CDP team is developing. Their idea of success would be seeing an application that explains what the CDP team does. The presentation of the poker application could improve collaboration between departments at Microsoft [6].

Table 3.3 - User Needs: Collaboration between Departments

Element	Description
The problem of...	Microsoft [6] associates in other departments not knowing about the parallel computing [1] division
Affects...	current Microsoft [6] associates
And results in...	less collaboration between departments. If Microsoft [6] associates in other departments do not know what the parallel computing [1] team is working on, they cannot collaborate or make suggestions for extensions of their work.
Benefits of a solution include...	Microsoft improving communication between departments. The Microsoft [6] associates will see an application of what the CDP team is working on, so they may have a better understanding of what they do. This poker application should be flashy enough to catch any associate's attention - even if they were unfamiliar with parallel computing [1].

The potential Microsoft [6] customers that are interested in learning about Microsoft products would usually understand the basics of parallel computing [1]. Their deliverables would vary depending on the type of development team with which they work. However, their work would involve

elaborate programs and high-end hardware. One of the main advantages of parallel computing [1] is that it can take advantage of high-end computers to a fuller extent than sequential computing [2] can. Not using hardware up to its full potential is a problem for the customers, so the poker application could show them a potential solution. The customers' definition of success will be seeing that the poker application that uses parallel computing [1] to solve a real-world problem in a more efficient way. Seeing an application that was created using the CDP API [10] will help them see how relevant Microsoft [6] products are to their work.

Table 3.1 - User Needs: Customers' Programming Efficiency

Element	Description
The problem of...	Microsoft's [6] potential customers not using their computing power up to the full potential
Affects...	Microsoft's [6] potential customers, the programmers that work for them, and the people who buy the customer's products
And results in...	less efficient programming techniques, which leads to companies producing less efficient products
Benefits of a solution include...	customers realizing how to use Microsoft's [6] API [10] to improve the efficiency of their systems. They would learn this by interacting with the poker application and looking at the code behind it.

All users have specific needs that will be addressed with the poker application. The CDP team's needs will be met by the application clearly showcasing the benefits of parallel computing. Also, their need for feedback will be met by poker application's team members providing comments and suggestions in the development period. The solution for the needs of both Microsoft associates who are not on the CDP team and potential Microsoft customers is the visually appealing poker application itself. All user needs will be met.

Part IV

Project Overview

The project overview portion of the document provides a high level overview of the project.

4   Product Perspective

4.1   Alternatives and Competition

The poker application will not be replacing, upgrading, or creating a new system. Instead it will be used to demonstrate the capabilities of the CDP API [10]. This will in turn be used to convince customers that using Microsoft's [6] API [10] will provide a simple programming environment and optimize their applications. Since this project is for demonstrative purposes, there is little competition. Microsoft's [6] CDP team has chosen Rose-Hulman students to create this demonstration. An alternative to this approach is for the CDP team at Microsoft [6] to outsource the project to a software company. The CDP team could also assign the project to team within Microsoft [6].

4.2   Project Perspective

The primary premise of this project is to prove the capabilities of parallel computing [1]. Parallel computing [1] is a relatively new technology that allows multiple processes to be completed simultaneously. This allows faster completion of programming tasks, resulting in greater efficiency and lower costs. Microsoft's [6] CDP API [10] will support parallel computation [1], and the CDP team would like a demonstration of its capabilities.

The end goal of this project is to prove that parallel computing [1] is superior to sequential computing [2]. Microsoft [6] will potentially use this project as a demonstration and marketing tool to convince consumers that parallel computing [1] is worth an investment and is truly superior to sequential computing [2]. The audience for this application will likely already have a basic knowledge of parallel computing [1] and will thus be more easily swayed by the success, or failure, of our results.

While the primary purpose of this project is demonstrating how parallel computing [1] is superior to sequential computing [2], our group's perspective also includes the aspects of developing a competent AI [11] poker player, as well as providing the Microsoft [6] developers with feedback regarding their API [10]. In this project, we will need to design two AI [11] engines that are capable of playing Texas Hold 'Em [12] poker against each other. To do this, we will have two options. The first is to create our AI [11] players from scratch, implementing all of the different rules of the game in their behavior. The other option is to find an existing AI [11] with open source code and modify its code to make it parallel.

The other perspective of this project is to provide feedback to Microsoft on the CDP API [10]. The team will provide this feedback to Microsoft [6] throughout the development process of the project. To provide this feedback, the team will write up comments about the tools used to develop the parallel poker player application.

4.3 Elevator Statement

We are currently developing an application that demonstrates the capabilities of parallel computing [1] using Microsoft's [6] parallel computing [1] platform. The application will show a side by side analysis of parallel computing [1] and sequential computing [2]. The comparison will involve two AI [11] engines who play against each other in Texas Hold 'Em poker [12]. One AI [11] will be written using parallel computing [1], and the other will be written using sequential computing [2]. This system will give Microsoft [6] a visually appealing application to clearly demonstrate the superiority of parallel computing [1] and the benefits of the CDP API [10].

4.4 Summary of Capabilities

Customer Benefit	Supporting Features
Microsoft has a demonstration for potential customers	Parallel [1] and sequential computing [2] based AI [11] poker players
The superiority of parallel computing [1] is clear to customers	Real-time computational display, demonstrating the time taken by each AI [11] to perform tasks

4.5 Assumptions and Dependencies

The poker application is not based on any assumptions. There is, however, one dependency. The developing team is dependent on Microsoft [6] for training on how to use the CDP API [10].

4.6 Cost Estimate

Financial constraints are not applicable to the poker application. Since Microsoft [6] will provide the programming tools for our team, there will be no associated costs for this project.

5 System Features

The following tables contain all features that have been proposed at this time.

Feature Number	Feature	
1	Interactive Graphical Interface	
Status	**Priority**	**Effort**
Proposed	Medium	Medium
Risk	**Stability**	**Target Release**
Low	High	First release
Assigned To	**Rationale**	
Benjamin Waters	To provide an attractive and interactive interface for users who are utilizing the system. The priority is medium because it is not required for the system, but it would be beneficial to the user for demonstration purposes. The user interface features were designated to Benjamin due to his development experience of front-end systems.	

Feature Number	Feature	
2	Computational Performance Display	
Status	**Priority**	**Effort**
Proposed	Critical	High
Risk	**Stability**	**Target Release**
Low	High	First release
Assigned To	**Rationale**	
Benjamin Waters	Constantly provides performance statistics for the analysis of the parallel computing [1], which will reinforce the superiority of parallel computing [1] to the viewer. This feature has a priority of critical because it is necessary to showcase the superiority of parallel computing [1].	

Feature Number	Feature	
3	Parallel Computing [1] Based Player	
Status	**Priority**	**Effort**
Proposed	Critical	High
Risk	**Stability**	**Target Release**
Low	High	First release
Assigned To	**Rationale**	
Ian Roberts	Essential for the demonstration of the parallel computing [1] technology in comparison to traditional sequential computing [2]. This feature has a priority of critical because it is a core element of the system. Ian has some experience in working with parallel computing [1] and so will be in charge of the application's usage of and connection to the parallel computing [1] API [10].	

Feature Number	Feature	
4	Sequential Computing [2] Based Player	
Status	**Priority**	**Effort**
Proposed	Critical	High
Risk	**Stability**	**Target Release**
Low	High	First release
Assigned To	**Rationale**	
Ian Roberts	Essential for performance comparison between parallel [1] and sequential computing [2] in the application. This feature has a priority of critical because it is a core element of the system. Ian will also be in charge of the sequential computing [2] player to keep it consistent with the parallel computing [1] player.	

Feature Number	Feature	
5	Support for Human Player	
Status	**Priority**	**Effort**
Proposed	Medium	Medium
Risk	**Stability**	**Target Release**
Low	High	Second release
Assigned To	**Rationale**	
Ian Roberts	User can interact with the application and play the game with the automated players. The priority is medium because it is not required for the system, but it would be beneficial to the user for demonstration purposes. The target release is the second release; the application will first be developed with only two computer-based players.	

Feature Number	Feature	
6	Artificial Intelligence Engine Driving Computer-player Decisions	
Status	**Priority**	**Effort**
Proposed	Critical	High
Risk	**Stability**	**Target Release**
High	Medium	First release
Assigned To	**Rationale**	
Mark Jenne	The automated players need to demonstrate basic analysis and reasoning skills and be able to make informed decisions. The artificial intelligence [11] engine poses a high risk. If the engine is developed and not yielding intended or appropriate results in the decision-making process, the structure may need to be reconsidered with a different reasoning system. Since all of the actions of the automated players rely on this feature of the system, it is of critical importance that it be developed properly despite the level of risk present. The AI [11] engine is also the least stable as it may require alterations or redesign if it is not yielding desirable results. Mark has experience in AI [11] development and so will be in charge of the development of and implementation of the AI [11] engine.	

Feature Number	Feature	
7	Support for Two or More Automated Players	
Status	**Priority**	**Effort**
Proposed	Critical	High
Risk	**Stability**	**Target Release**
Medium	Medium	First release
Assigned To	**Rationale**	
Mark Jenne	The system, particularly the AI [11] engine should be able to support two automated players and preferably more. This feature has a priority of critical because it is a core intended function of the system.	

Feature Number	Feature	
8	No-Limit Texas Hold 'Em Poker	
Status	**Priority**	**Effort**
Proposed	Critical	High
Risk	**Stability**	**Target Release**
Low	High	First release
Assigned To	**Rationale**	
Sarah Jabon	Game structure of the application which offers a sandbox environment to demonstrate the parallel computing [1]. All the rules of Texas Hold 'Em [12] poker will be implemented. This feature has a priority of critical because it is necessary to satisfy core requirements of the system as specified by the client.	

Feature Number	Feature	
9	Ability to Select Type of Player Types	
Status	**Priority**	**Effort**
Proposed	High	High
Risk	**Stability**	**Target Release**
Low	High	First release
Assigned To	**Rationale**	
Sarah Jabon	Gives the user the ability to designate automated players using sequential [2] or parallel computing [1]. The priority is high because it is not required for the system, but it is important for the user to interact with the application.	

Feature Number	Feature	
10	Ongoing Summary of API Feedback	
Status	**Priority**	**Effort**
Proposed	Critical	High
Risk	**Stability**	**Target Release**
Low	High	First release
Assigned To	**Rationale**	
All team members	Provides the CDP team with feedback that they can use to improve the API [10]. The priority of this feature is critical because it is one of our client's core requirements.	

5.1 System Features Scale Key

Status - The status of a feature pertains to the progress or current state of the feature during the planning and design stages of the system specified by the project.

Priority - Priority here refers to the importance of the development of a feature based on its significance to the overall system. Of the features listed, their priorities range from medium to critical levels. The critical features are labeled as such because they are necessary for the proper functionality intended of the application as specified by the clientele.

Effort - The effort attribute of a feature describes the amount of time and resources that need to be designated to the development of that feature based on the level of priority and overall importance to the structure of the system. The critical features compose the base functionality of the entire application and represent the goal of the project as a whole. All of the high priority features are integral to the proper functioning of the application as well, but are not quite of dire importance.

Risk - This attribute represents the likelihood that a feature will cause unintended results, such as cost overruns or delays. A low risk represents a lack of potential for undesirable events.

Stability - The stability of a feature reflects the probability that the feature will change or the team's understanding of the feature will change throughout the development of the feature. The stability of a feature relates to the risk that feature presents. If a feature causes unintended results, it will very likely be unstable and need to be redesigned.

Target Release - This records the intended version of the product or system in which the feature will first appear and usually reflects the priority of and effort delegated to that particular feature.

Assigned To - Refers to who within the development team will be responsible for the implementation of the feature. Given that we have four members on this project team, each with capabilities maybe not seen in the others, the features must be distributed for development to the most capable of individuals.

Rationale - The reason or reasons for intended implementation of a feature can be required by the system, mandated by the client, or just a feature decided upon to better the user experience in using the system.

6   System Constraints

Numerous constraints have been established.

Constraint	Source	Rationale
Developed on Windows[8] platform	System	Since we are working with Microsoft's [6] CDP API [10], we must use a Windows [8] platform.
Developed in a native language (such as C++ [9]) or managed language (such as C# [5])	Systems	The .NET framework [4], which the developing team will use, supports these types of languages.
Timeline for deliverables	Environment	The team must have a clearly planned progression
Finished by June 2010	Environment	The team members will not be available to do project work after that date.

Part V
References

7 Glossary

- C++ - an object-oriented programming language that consists of both low-level and high-level language features [9]

- C# - an object-oriented language that can be used to create a wide variety of applications, services, and tools [5]

- Microsoft Corporation - A multinational corporation that has five business segments: client, server and tools, online services business, Microsoft business division, and entertainment and devices division [6]

- .NET Framework - A platform build by Microsoft that provides a common set of APIs and a consistent programming model [4]

- Parallel Computing - using multiple resources simultaneously to solve a computation problem [1]

- Sequential Computing - solving a computational problem with sequential processing [2]

- Visual Studio - an integrated development environment produced by Microsoft [7]

References

[1] Blaise Barney. What is Parallel Computing?, 2009. `https://computing.llnl.gov/tutorials/parallel_comp/#Whatis`.

[2] Barney Blaise. Overview: Sequential Programming, 1995. `http://www.hku.hk/cc/sp2/workshop/html/parallel-intro/ParallelIntro.html#sequential%20programming`.

[3] Bathesda Softworks LLC. Fallout 3, 2009. `http://www.bethsoft.com/eng/games/games_fallout3.html`.

[4] Microsoft. .NET Framework Overview, 2008. `http://www.microsoft.com/net/Overview.aspx`.

[5] Microsoft. Getting Started with Visual C#, 2009. `http://msdn.microsoft.com/en-us/vcsharp/dd919145.aspx`.

[6] Microsoft. Microsoft, 2009. `http://www.microsoft.com/en/us/default.aspx`.

[7] Microsoft. Microsoft Visual Studio, 2009. `http://www.microsoft.com/visualstudio/en-us/default.mspx`.

[8] Microsoft. Windows, 2009. `http://www.microsoft.com/WINDOWS/`.

[9] Juan Soulie. A Brief Description, 2009. `http://www.cplusplus.com/info/description/`.

[10] Wikipedia. Application Programming Iinterface, 2009. `http://en.wikipedia.org/wiki/Application_programming_interface`.

[11] Wikipedia. Artificial Intelligence, 2009. `http://en.wikipedia.org/wiki/Artificial_intelligence`.

[12] Wikipedia. Texas Hold 'Em, 2009. `http://en.wikipedia.org/wiki/Texas_hold_%27em`.

Index