The chapter discusses about web search engine and its relation with web information retrieval system. We will see basic operation of web search engine. The World Wide Web (www) keeps a lot of facts such as business, entertainment, education etc. But this relevant fact is not kept properly in an arranged way. Better facts are hidden somewhere in one of the pages on the internet. Hence web mining is the solution for finding relevant fact. It plays a key part in generating useful facts from the data available on the net. Therefore there is a requirement of a good spider [127].

6.1 Web Search Engine:

Web search engine recovery stages to get forms depending on user’s question [1]:

i. The person using the look-up engine enters the question. It has to return completely or limited list of contents. This is similar to a dictionary in English language. Listing is not for all words. Consider important words.

ii. The lists are arranged in the form of layers and sections. It difficult to scrutinise all.

As shown in figure 4.7. The users enter the inquiry, the spider indexes the fact available on the web and retrieves and extracts the relevant information searched by the user. The indexed page need not always be retrieved, as it may be restricted on the general web searches but can be available on site specific search. Availability of site specific search is a norm now-a-days.
6.2 Spider:

Spider is a technique for collecting feeds from the internet. Then arrange them so as to retrieve the feeds to give better outcome using the search approach [1]. Spiders are huge approaches that find thousands of feeds every moment. Spiders are also known as internet spiders, internet automation and so on. This fetched data is then indexed. This is the basis for search engines. The earliest search came into existence in 1994 a research project [128]. Spider does not return data in arbitrary order for a give inquiry as in relation data. It returns optimized information. Various components of web search engine are shown in figure 6.1.

Figure 6.1: Various parts of Internet explorer machine [1]
6.2.1 Working of the spider:

The information to be extracted from the internet is not an easy task. The applications deal with extracting text data effectively. The explore machine is used to scan wide range of records. The explore engines can be used for finding particular information and for scanning lakhs of records. The application may be written for specific retrieval or multi-information retrieval. Table 14 gives the extraction of text using explore machine. Some specific words can be used to get the following information. Cent percent, completely are the words; theses are used as pair of words with sure, definite, certain and positive respectively. The numbers of hit are mentioned in the tables. These are calculated using the rate as shown in the table.

The explore engines work using some words or a words. The records are scanned for some text and not the whole text that one is looking. Therefore it’s important to use keys in the document being generated. This will help in scanning the specific information and executing better outcome. The explore machines give different information each time the explore words are executed. Hence the outcome of the inquiry may not me the same each time the inquiry is executed.

Table 1: Extraction of text

<table>
<thead>
<tr>
<th>Shots</th>
<th>sure</th>
<th>Definite</th>
<th>Certain</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cent percent</td>
<td>340,000</td>
<td>240,000</td>
<td>1,000</td>
<td>34,000</td>
</tr>
<tr>
<td>Completely</td>
<td>231,000</td>
<td>184,000</td>
<td>3,000</td>
<td>500,000</td>
</tr>
<tr>
<td>Rate</td>
<td>1.47:1</td>
<td>1.30:1</td>
<td>1:0.33</td>
<td>1:0.68</td>
</tr>
</tbody>
</table>
For generating better outcome for a given inquiry use program code interface. This will help in giving specific information. Program code interface is a set of rules for writing a good program. It deals with bulk information separately, and then combines these bulks to give better outcome.

### 6.2.2 Types of program code interface (PCI)

There are a lot of Program code interfaces for operating structure. Some companies selling operating structure provide program code interface. The developers can use these interface to generate better outcome. The interfaces are used for internet based execution machines. For example: A programmer creates a site named e-market which wants to give best deals available on branded sites. Here the programmer uses the interfaces available and produces the results of best deals of branded sites on e-market. The thesis deals with specific interface.

PCI are to be available for implementation as tertiary equipment or accessible for use on the web. The programs are coded in varied languages therefore midway is used so as to help the applicant work on all types of systems and machines. The figure 6.2 depicts the PCI. The function information is not visible to the other function.

PCI are chunks of program that developers utilize as a part of their implementation.

**The aim of the PCI: Working code**

PCI tries to help the programmer to overcome his/her difficulty. PCI should be a working code and not some part of a code that does not work as per requirement. Therefore PCI needs to:

- Be simple to understand and implement
- Be workable
- Be able to generate the required results to the end-user issues
- Be less problematic and not generate unnecessary compulsions
The graceful group feels “innovation is economical”. It is economical if it’s done innovation is done in bits. These bits make up a whole of the development. The PCIs are implemented by one developer of a group of developer of a particular issue. But these PCIs are used by a tertiary user hence the issues are different. Care needs to be taken while using a PCI in one’s code to make the process simpler. For the end-user the reorganization may create issues while using in community blend.

Reorganization is costly in PCIs community blend:
The PCIs have a personal and a community blend. The end-user may not like the reorganization that may detach the existing code. The reorganization of personal APIs can be done without the end-user getting affected. But there is a need to be careful. The community components are the ones that the end-user interacts with. Hence any reorganization needs to be properly done keeping the community into consideration.
Beforehand layout:
There is a need to focus on understanding the scenario before the PCI is implemented. The blends need to interact properly with the available components. The community aspect needs to be considered while designing PCI.

The problem that need to be focused for beforehand design:

- Community blends visual.
- PCI set up.
- PCIs required features.
- Conditions available for PCIs conceptual level.

Steps when there is a requirement to update community blend of PCI:

- Give another blend, and let the previous blend of PCI as it is.
- Give end-users the choice to change to the new version of the PCI blend.

It’s required to update the new community PCI blend automatically after a few number to version updated of PCI community blend. But the end-user needs to be given a chance to update according to his preference.

**PCI layout: Should not be easy**

PCI layout should not be clear for the developer; depending on the requirement the developer should be able to create a better blend.

The spider and the list are used together while designing a explore engine. But it’s better to design the spider separately; the developers need not understand the generation of lists, as the lists are complicated to implement.

However the spider and the list are available while implementing the PCI. In most of the cases its not required to understand the list generation, the developer just needs to focus on the spired for implementation. But to provide the list information is helpful while examining the code. This will help in detecting the drawbacks

**PCI layout: Give reasonable required fields**
There are some required functions, specifications of functions, blends that are used most of the times. There is wastage of space, time and energy in taking there or writing them again and again. Therefore it's better to develop an inbuilt method that can be used as and when these required field need to be used.

Give required reliance:
The required methods are inbuilt as seen in the previous paragraph; the same is applicable for reliance methods. The reliance takes the different components of the application to be created. These components generated reliance that is not easily visible. The reliance is coupled to other blends; there is no necessity of decoupling it.

**PCI layout: choose abstraction**

It’s a method that makes the PCI more elastic in nature. These need to be chosen carefully. The layout for the abstraction is an amalgamation of Program as level, PCI as level, chooses abstraction.

**Program as level:**
There are levels in internet set of rules, network set of rules etc. Similarly there is a program as a level. The highest level generally interacts with the lower levels to establish with other programs as per the requirement. Each level is compressed; it makes dialogue with other levels faster. PCI may be considered as one of these levels. These are illustrated in the program languages that one uses for implementation of PCIs. Table 15 gives the layout of program as level. Each level in the table 15 is a consolidation of the level below it.

**PCI as level:**
Table 15 shows the PCI as a single level. PCI in itself comprises of different levels. Table 16 shows PCI as a level. The last level is for starting and stopping the information service. The first level scans and composes the item; this is helpful for scanning the picture. The question to generate picture scan is not generated from one location but from multiple locations.
Chooses abstraction: It’s known that the levels communicate with the level on top of it or level beneath it. But choosing compression gives it flexibility to skip some levels and go to other level. The operation is to start and stop PCI, here the level item scan and item compose are not required and there is no requirement to scan the picture. For starting and stopping of PCI service is also not required. Here the PCI information link is used for start and stop of the application. Another instance is to directly activate the PCI service and have the item scanned and composed according to the requirement.

Here the item-to-sequential gauging is mechanized; this technique is used to identify the retrieval of items from a particular sequence, along with the information of the rows and columns in the whole data. It is created to make the whole process simpler for developer.

**Table 2: Program as level**

<table>
<thead>
<tr>
<th>Program Use</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Program PCIs</td>
<td></td>
</tr>
<tr>
<td>Program In-essence</td>
<td></td>
</tr>
<tr>
<td>Engine</td>
<td></td>
</tr>
<tr>
<td>Operating Structure</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3: PCI as level**

<table>
<thead>
<tr>
<th>Item</th>
<th>Picture Scan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan/Compose</td>
<td></td>
</tr>
<tr>
<td>Java PCI service</td>
<td></td>
</tr>
<tr>
<td>Java PCI information link</td>
<td></td>
</tr>
</tbody>
</table>

It is important that the terms are same in the function as well as those used in the back-end. If the terms are different in the function retrieving it and the
back-end then it becomes difficult to process the data. If the terms in function and terms in back-end are different, then human intervention is required. The solution is to combine the mechanized and the human intervention process for better outcome.

**PCI layout: Mid mark of admission**

The goal of the PCI is to make acquiring of information simpler. Try and combine as many functions as possible and have fewer functions. The developer needs to have an idea of the working of PCI; this is possible by having a mid-mark of admission to the PCI. Few ways that gives the mid mark of admission to a PCI.

Analyze program achieve:
The developer can get admission to mid mark of admission using the analyze program achieve. If the developer figures the ways in which analyze program achieve is used via mid mark of admission, it easier to get information of other characteristics. Hence it also has information of PCI documents.

Industry as mid mark of admission:
An industry function can be accessed as mid mark admission. There is a need of a single industry through which the items can be retrieved in a PCI. Every item outside the factory should also to be accessible. The number of levels in the information generation is immaterial. The developer can access the PCI using program achieve, with industry as a beginning point.

Administrator as mid mark of admission:
In few PCIs, Industry as a mid-mark of admission may not be feasible. There is a need to bring together industry and an item with some PCI characteristics.

Appearance as mid mark of admission:
PCI’s can also access mid mark of access using appearance. This will not get information of PCI straight, but will approach the aids given by the PCI
through the industry function. Appearance is better than having a mid-industry or administrator function. There can be different industry each representing a component that takes care of items needed to generate aids the PCI supports. The industry layout can be changed, here the mid industry function is not required.

Aid alternate as mid mark of admission:
Here aid oriented design (AOD) is the mid mark of admission for aid that could be an alternate. There are aids that use aid oriented design protocol (AODP) that may give a mechanical outcome to an aid alternate from the Internet aid phrase program (IAPP) to gain admission.

AOD as a aid alternate is as good as appearance. Therefore using a mid-mark of admission or appearance for communication is not of great concern.

**PCI layout: Developer need not combine the parts**

Reliance infusion is used in abundance. All the reliance’s need not be infused particularly the criteria of reliance considered in a PCI. Take a function L it has reliance’s M, N, O, P. Here L is available to the outside world. Though the reliance infusion can be used it need not be displayed. If the overall L, M, N, O, P layout is given out; it takes a lot of time to understand the overall architecture, making the PCI implementation complicated.

**PCI layout: Refrain outside reliance**

While implementing PCI outside repository are available in plenty. Using outside repository makes the program bulky and execution sluggish. Therefore it affects the implementation.

The code of the outside reliance may not gel well with the existing PCI layout. Outside reliance of repository may be used at the end of PCI implementation.

**PCI layout: Refrain signing in PCIs**
Signin is used to refrain from outside reliance. As you enter in the PCI you signal a Signin PCI. The Signin PCI is used instead of the end-user PCI, therefore the developer deals with two PCIs. A blend is required to signin into PCIs. The use can join as many PCIs that the user needs. Signin exclusion: There is no need of Signin exclusion or blend.

**PCI layout: Layout for examination**

As is the case with any application, examination needs to be done before releasing PCI. The examination needs to be done for different components and examined for the entire PCI. The PCIs examination is done for:

- Examination of PCI
- Examination of the program that utilizes PCI

Layout for examination of PCI:

The PCI should be able to mimic the program examination. If the examination is for an organization, it should be able to mimic different parts of the program and safe to introduce these mimics into the parts that need to be examined. For introducing the mimics into the program there is a need to write a separate function to do the introduction of mimics in different functions of PCI. The functions that are used to introduce mimics in other functions need to be safe.

Layout for examining of program utilizing the PCI:

The previous paragraph gives the means of examining of PCI, but there is a need to examine the program that uses the PCI. The developer has to keep this in mind that the examination needs to be mimicked while using the PCI available for community.

Utilize blends:
Blends need to be used while mimicking the scenario to be tested. The blends need to be checked live while chunk examination.

Utilize expandable programs:

If blends cannot be programmes then try creating a subfunction. Every PCI function will have a subfunction that will mimic.

**PCI layout: Layout for effortless arrangement**

Arrangement is required for any aid to execute in a better form. The PCI needs an executor, location, identity and secret word.

The PCI arrangement methods are:

- Function call on parts
- Glossary
- Near real limit
- Records
- Flexible hyper-document record
- Area related language

The arrangement method to be utilized depends on the application one is developing. The arrangement selection methods could be many:

- What are requirement of arrangement in the code?
- Is arrangement a programming criteria or execution criteria?
- Is arrangement method simple to understand, utilize and program?
- What are the different compulsions of arrangement methods?

The arrangement selection for one program may not be the same for another program. The arrangement selection inquiry needs to be replied before implementing the PCI application.

What are requirement of arrangement in the code?
Better to have a less arrangement program written. There is a need to check for the need of arrangement. The program needs to be thoroughly studied for understanding the positive answer or a negative answer for used of arrangement in the program.

If the arrangement is from highest to lowest order then hyper-document seems to be fitting the requirement.

Is arrangement a programming criteria or execution criteria?

PCI arrangement programming may be related to execution requirement. The path where the information stored may change. Here the execution may also change accordingly.

Execution needs to be done outside the program, segregated as separate documents, data etc. IT’s better to direct PCI to be arrange through a program.

Is arrangement method simple to understand, utilize and program?

The arrangement should be such that the PCI should be simple to understand to implement. Most developers have the knowledge of flexible hyper-document; therefor using it in the program may be simple for the end-user.

What are the different compulsions of arrangement methods?

Selecting an arrangement could bring along with it different compulsions on the arrangement. There might be availability of glossary to get specific information of the way in which data is stored for utilization of the program.

Glossary is functioning specific; one function cannot have separate arrangement of the same function. This is one of the compulsions of an arrangement method.

Rearranging the program is not an easy task; therefore flexible hyper-document is good for arranging highest to lowest. This is another compulsion while writing a program.
The PCI gives steps to the programmer for properly dealing with the input. The programmer connects the PCI to the OS. The PCI asks assistance from the OS or different programs. PCI’s are programmed using job calls using nominal or action word. PCI’s are mainly used by tertiary party for implementation as a part of the whole. These are mainly available on the web, the user has to abide by the specifications mentioned in the PCI’s. They are mainly stage independent. A midway is used where money is involved.

6.2.2.1 Midway:

Midway is a program that connects program parts or applications. Midway is a code level that is positioned between the operating structure and the function at each end of a wide network. Midway is a business application and an internet aid. Suppose e-market is a site that sells mobiles, it uses different PCI’s to gather best deals provided on different sites. User needs to pay money to buy the product. Since the buying and selling happens online there is a requirement of internet aid, here it’s the bank. The user has to provide his/her credit card number in the field to complete the process. Once the card number is authenticated by the bank application, the mobile will be dispatched to the user at the specified address. The figure 92 deals with the architecture of midway.

Usually the midway provide text communication aids, this helps the different programs to communicate using communication layout which are internet aid, java aid, simple article retrieval rule, symbolized state deportation. Midway is an aid that connects different programs together to provide simultaneous, business dealing, tendril and text communication. It also deals with hazard management and liability management.

The online websites are growing fast on the internet. There is a lot of application dealing with electronic dealing. This makes midway program an integral and important part for doing business online. Organizations are relying on midway for the authenticity and security aspect for business dealings that are mainly dealing in business. There might be some old applications that cannot be blended with the existing midway. Here the code
needs to be updated and modified to fit into the existing system. In some cases the code cannot be modified but has to be changed completely. Changing may be helpful if the old system is generating a lot of profit. If rewriting the whole code of the old system equal the profit or it may not generate profit, then using an off-the-shelf model is a better solution.

![Figure 6.3: The architecture of midway](image)

As shown in figure 6.3 the midway is an interconnection of different exclusive tools connected via internet or network. They execute modules for live applications and far-flung systems. These could be providing current 24/7, telephone system, data processing machine, de-pivotal systems etc. Therefore the intercommunication is between lots of systems, it needs to give good results for all the intercommunicating systems. It needs to work in
different network capacity. It can be used as a private data processing machine.

Modules of midway
The midway program is used on top of operating structure and transmission rules to execute the following:

- The user does not get an idea of the disseminate nature of the program. The program is a combination of parts that are working in a disseminate areas that are far flung.
- The midway is diverse in nature which is not known to the end user. It connects different parts as depicted in figure 92.
- The specifications are same for similar programs. Therefore the program need to be coded in a way that is easy to use, works on any operating structure(OS), can be change and intercommunicate.
- Use usually available aids to execute different modules and avert duplication. The module should work together.
- Midway makes coding easier; it gives normal code conceptual by making it diverse to the tools and OS.

The work of the midway is to negotiate between other components of a program, or between programs. The framework of the midway is pivotal to the midway layout. The framework layout deals with the complete working, complete outline and interacting design for both the module and midway. The interaction between systems could between independent-dependent or between associate-associate levels. The communication in midway could use direct, indirect or between mutual entity embedded in varied components.

Types of midway
The midway is also used for security purpose; the independent component tries to connect to dependent component to complete a process, here the dependent component needs to be sure that it is providing information to a secure user and not to an unauthorized user. Once authorized the independent component proceeds in dealing management. Figure 6.4 gives an
insight into dealing management. The communication midway makes connect with all other data and updates it regularly.

While dealing with the midway there is a need to make payment and make sure that that backend data is not altered. Communication line has to be lightly joined to other components. The communication takes place both the ways from independent to dependent and from dependent to independent parts. The figure 6.5 deals with communication midway.

Figure 6.4: Dealing management
Then there is a program dependent; it’s a dependent that holds the PCI, which works as per the need of the organization for which solution is provided. The flow of operations is as per the requirement of the organization. There needs to be a shared flow of programs. The internet dependent is a data processing machine program. The program has to accept the request of the web viewing software and respond and display the result on the web pages. Disseminate working is depicted in figure 6.6. The programs in the disseminate working can be executed from any location in the network. In communication midway the information is sent on request, but here the communication takes place as and when the parts require data. Disseminate database midway is DEdealer. Midway is a normal interaction with the enquiry and other parts as shown in figure 6.7. The communication can be with the header or footer. The information is compressed to make the information flow quick.
Normal Blend program is establishes connection between different programs, these are also used as midway. This is shown in figure 6.8. The Accessible Data storage Attachment(ADA) makes programs to take a quality order to the other ADA blends to perform task in a better way.
Program Dependent Midway: As shown in figure 6.9 it acts as a connection between different programs on the internet that connects user, dependent and independent components. The user can use a static system or a ubiquitous system to run programs or get information from the internet. As Java language and related programs are used a lot of program executions can be dealt with easily.

Ubiquitous evaluation is the mother of evaluating programs; it links the program to an operating structure. The figure 6.10 depicts the ubiquitous evaluation. The program cannot work without an operating structure. The end user may be from an internet viewing software or program written in other languages.

Usually JAVA as a language is preferred by any organization providing resources and an environment to work, as it gives the flexibility to run on any operating structure at any given time. Hence there is no need of using a universal operating structure or a specific tool on which the JAVA program.

Internet login is a midway for internet. Figure 6.11 shows the process of internet login. It has to recognize the end-user, if the end-user is recognized then it permits the end-user to go through the lists to and use them if necessary.
Figure 6.8: Normal blend
Figure 6.9: Program dependent midway

Figure 6.10: Ubiquitous evaluation
The threats to midway are:

- There are a lot of interrupts and ambiguous working that affect the execution of midway. Flexible midway have more interrupts making the execution to suffer.
- As the interactivity and dependency increases of the program increases, the entities, user and tools increase. Taking care of large range affects the interaction and entity organization methods and management becomes complex. The quality of service is affected. These are usage, safety, execution, parallel process, ready to use, dependability.
- There is a need for midway to be available to the ubiquitous systems.

**Aid oriented design (AOD)**

AOD is a design that tries to implement light interaction between modules of varied communicating programs, it tries to use the existing systems to its maximum. It tries to create better communication between systems that will help the organization in doing good business. It used the functional aspect aid that can be clubbed and reiterate leading to a adaptable design. AOD helps in taking care to known and untoward incidents in any business process. The highlights of AOD are:
• It is known to establish light dependency among modules.
• Shortens the period and amount: The aids can be reworked and put in new modules.
• Less money used in upkeep: The aids can be reworked reducing the difficulty in the operations.
• Greater characteristic aids: The Aids reworking helps in maintaining greater characteristic aids via varied tests using varied aid users.
• Lesser clubbing amount: Even though the programs are written in different languages there is no problem in communicating amongst each other.
• Low threat: Less rework able aids help in giving better mastery over organizations, government rules and complete threat management.

How does AOD achieve light dependency between different program mediators?

• A tiny group of interactions are used in all cooperating mediators. Only useful portions are embedded at the interactions. There interactions are there for all users and suppliers.
• Well-structured message is transmitted through the mediator. All components do not work, only the ones which use and supply are in tandem. Addition of new aids does not hamper the working of the existing system.

6.2.3 Web spider and indexes

Idea of the spider is to return information quickly and efficiently. Hence, index the text where information search can be done using keywords. To perform indexing, one needs to fetch all pages to be indexed using a spider.
6.2.4 Spider architecture:

For a spider to perform well, it should fetch all available “web pages”. The URL’s of each site needs to be known. For information of URL’s scan collected web pages and find hyperlinks to other pages. Hyperlinks which are not yet stored have to be collected. Writing a spider program for small data fetching is easier than writing for huge data. The data fetched by a spider should be stored in discs. This mechanism will help during spider crash.

For a spider to perform effectively in an IR system the following steps mentioned in the figure 89 will improve the efficiency[1]:

![Image](image_url)

**Figure 6.12: The basic spider architecture**

i. The spider has to fetch new URL’s continuously, in the process some URL’s might be re-fetched.
ii. Web server is determined by DNS resolution module to generate page specified by the URL[129].

iii. Http set of rules is used by the fetch module to retrieve web page from a URL[130].

iv. A yielded web page should have a dissecting broadness which remove content and agreed upon connections.

v. A broadness which finds and deletes the similar connection from recently yielded fact.

Spider architecture has to take care of the time spent in scanning pages to detect outliers. The steps mentioned above continue until sufficient pages are retrieved along with outliers. Mentioning the number of sufficient pages is a tough take in the internet scenario. Organizations with less storage space need to stop the spider as the spider goes on fetching the pages.

6.2.5 Engineering Large-Scale spiders

Large scale spiders [128][129][130] extract huge sets i.e. millions of pages. Overview of an internet spider is as follows:

i. Is an automated browsing program whose main purpose is to mirror the web and can be used for site maintenance and harvesting.

ii. Spider architecture comprises of modules, these modules are URL frontier, DNS resolution, Fetch, Parsing, Duplicate elimination by URL normalization.

iii. Has policies such as Selection (e.g. restricted followed links, path ascending crawling, focused crawling), Politeness Re-visit, Parallelization (multithreaded spider).

iv. Can be blocked from ones website by using robots exclusion protocol.

The main concerns of large scale spiders are

i. Fetching one page takes several seconds of network latency, hence fetching several pages at the same time one utilizes the available CPU
and network bandwidth. In multithreading the thread locks URL frontier to pick a URL. After URL is picked, it unlocks URL frontier as it can be used by other threads. To avoid duplicates of same URL’s it’s shared among crawling loops; the URL frontier is again locked once a URL is added for consistency.

ii. For fetching pages simultaneously DNS lookup has to be streamlined concurrent which run concurrently and be replicated on few DNS servers.

iii. Care is need to avoid “spider traps”, these keep the spider locked in a graph with infinite set of fake URL’s.

### 6.2.6 Performance of crawler/spider

Let’s look at three crawling models that can achieve better performance [131]. Figure 6.13 gives overview of a simple crawler using web.

**Crawl Stop**: The spider needs to be stopped after a certain process. So as to update the content, reset the index and then start the crawling process afresh to get new updated results.

**Crawl and Stop with Threshold**: A threshold has to be selected when the spider can stop crawling [132]. There are a lot of data available to crawl, hence crawl only a few at a time by setting a threshold. As is already seen in crawl and stop which will help if the spider is stopped so that content is updated and index is reset.

**Limited Buffer Crawl**: Here consider a limited stored information on the internet needs to be crawled[133]. Assume only some buffer which the spider crawls. When the information is stored in buffer the crawler needs to keep the fresh pages and delete the remaining.

**Specific Sites spider**: Until now the discussion was on spider stop, spider stop with threshold and Limited buffer spider. Rather than considering a buffer the thesis deals with comparing the prices and giving best price deal of a particular product. Crawl only specific sites which give best deals which is the need of the hour.
Here the Spider needs to display the best deals region wise. This will help in dispatching the product on time to the customer. The region specific Spider will get the location of the user. Depending on the location the Spider generates best deals available in that region.

![Diagram](image)

**Figure 6.13: Overview of simple internet spider**

### 6.2.7 DNS Caching, Prefetching and Resolution

For engineering large scale spider DNS catching, prefetching and resolution have to be considered. Entries are translated from domain name (google.com) to IP address in DNS cache[134]. Even if the location of DNS server changes the domain name will still remain the same. The domain name system uses client-server model which is maintained in distributed database.
6.2.8 Aspects a spider should cater:

An internet spider should give high quality results to an inquiry fired by the user. The following features have to be considered to satisfy the user’s needs for a spider to perform well[1]:

i. Distributed: The ability of a spider to execute on different machines in a distributed environment.

ii. Scalable: The ability to give good performance after including extra instrument and capacity.

iii. Attainment and ability: System resources such as processor, storage and network capacity must be efficiently used by the spider.

iv. Quality: When a inquiry is generated the spider should generate relevant information first.

v. Freshness: Each time the spider should fetch recent updated copies of earlier retrieved pages.

vi. Extensible: Spider architecture should comply with new data formats, new protocols etc.

6.2.9 SVM social group shopping

“Social media is currently the center of technological innovations and research. The plethora of actionable data made available by modern social networks has brought forth the need for use of intelligent approaches that can process such volumes of data. Group shopping websites are one of the innovations of such social existence of the web. Many websites such as groupon, livingsocial, dealster, buywithme etc. currently offer some form or the other of group shopping. The thesis presents a model of applying SVM approach to a case of group shopping with two aims:

i. Predicting potential customers for a given product which shall enable us to launch group shopping campaigns more effectively or consider whether they should be launched at all, in the first place.

ii. Rather than having open ended campaigns, implementing targeted marketing.
An application that implements the above is also presented. If the admin of the website realizes the potential of a product sold by the site or a vendor selling products and services complementary to the social shopping site, he or she may choose to launch the campaign. On doing so, the deal shall be available to all and the potential customers will be specially notified. “

6.3 Discussion

Internetspider is an important aspect of web mining. Finding the correct URL for the specific crawler is a task hence a restriction. It is recommended to grade the URL to get better retrieval outcome. There is huge information available on the web. Crawling through web text data to extract relevant information is an important task.