GAIUS: A New Mobile Content Creation And Diffusion Ecosystem For Emerging Regions

Talal Ahmad
New York University
ahmad@cs.nyu.edu

Yasir Zaki
NYU Abu Dhabi
yasir.zaki@nyu.edu

Thomas Pötsh
NYU Abu Dhabi
thomas.poetsch@nyu.edu

Jay Chen
NYU Abu Dhabi
jchen@cs.nyu.edu

Arjuna Sathiaseelan
GAIUS Networks
arjuna.sathiaseelan@gaiusnetworks.com

Lakshminarayanan Subramanian
New York University
lakshmi@cs.nyu.edu

ABSTRACT

Despite increasing mobile Internet penetration in emerging regions, the growth of web page complexity combined with inadequate content delivery mechanisms and lack of relevant local content make the web experience poor for users in these regions. In this paper, we propose GAIUS, a content ecosystem that enables the efficient creation and dissemination of locally relevant web content. GAIUS aims at creating a community content exchange that enables users to easily create and consume relevant web content with the help of a mobile application. Based on an early deployment of GAIUS in various regions, we show that our ecosystem enables localized content creation and diffusion efficiently.

CCS CONCEPTS

• Networks → Location based services; World Wide Web (network structure); Public Internet;

KEYWORDS

Mobile, Content, Local

ACM Reference Format:

1 INTRODUCTION

Many challenges constrain web access in developing regions. A measurement study by Zaki et. al. [20] showed that developing regions like Ghana suffer from high page load times (multi-seconds), due to HTTP redirects, DNS lookups and TLS/SSL connection setups. They found the actual time spent downloading content represents only a small fraction of the end-to-end page download time. Korn et al. [17] have shown that cellular data connectivity in India suffers from significantly high latencies. Other works have pointed out traffic engineering and lack of infrastructure as reasons for high delays [5, 11, 14, 15]. In extreme cases, power outages can also lead to network disruptions [3]. To overcome the challenges of poor web performance in developing countries, several systems and techniques have been proposed in recent years to optimize web browsing over poor network connections including network-level optimizations, caching techniques and content distribution mechanisms [6–9, 16, 19]. In addition to this, Content Distribution Networks (CDNs), which traditionally aim to replicate relevant content in the network edges [1, 2], are ineffective for enhancing performance in the developing world [4, 12, 18]. The underlying factors are three-fold: poor connectivity [10], high latencies, and web complexity.

Beyond the well-known technical and infrastructural issues, one major problem that has not been explored in detail is the lack of cheap, locally relevant content. The vast majority of web pages on the Internet are from developed countries, far removed from audiences in developing regions. This shortage of relevant local content makes the Internet less attractive for users in developing regions. The existing web ecosystem also makes it hard for people to publish locally relevant content. Furthermore, bloated web content and remote hosting also undermine web performance and user experience.

To address these problems, we propose GAIUS, a mobile ecosystem for local content creation and diffusion. Beyond localized content creation and diffusion, GAIUS also improves localized web performance and provides support for local advertisers. The GAIUS ecosystem is an immense improvement over the status quo in the context of localized content.

Architecturally, GAIUS consists of a set of locally distributed servers that host GAIUS pages (interchangeably referred to as channels) and an application running on a user’s smartphone device that enables consumption and creation of local content. The GAIUS pages are represented in a highly concise language format instead of HTML and JavaScript, to minimize communication overhead. Finally, the GAIUS app on an end user’s device receives, interprets, and renders the pages. The GAIUS application is also used to create content and publish it on the GAIUS servers. Together, these components enable the easy creation, diffusion, and consumption of
content at low cost. In summary, the GAIUS ecosystem is designed around a number of key concepts:

- Hosting and diffusing content close to the users.
- Using a novel specification language.
- Coordinating a mobile application and an edge server to efficiently view and create content.

We have preliminary deployment of GAIUS in a few small communities around the world. Our early deployment shows that GAIUS enables users to create and publish pages without extensive training or web publishing experience.

2 GAIUS ECOSYSTEM

The GAIUS ecosystem solves the problem of content delivery from local and global content/advertisement providers to mobile users. The main features of GAIUS are:

- GAIUS enables local content creation and diffusion by allowing users to create content within a mobile application.
- GAIUS content is defined using our own specification language called MAML, instead of JavaScript and HTML. This makes the pages simpler and makes them load faster.
- GAIUS provides a new way to identify and address content. In GAIUS content is divided into channels, where each channel is maintained by a single users or content provider. Within the channel a user/content provider can create several pages of his/her liking.
- GAIUS introduces a new ad ecosystem by providing support for local ads. Different tiers of advertisers can be accommodated in such an ecosystem varying from small local individuals to large corporations.

The GAIUS ecosystem as shown in Figure 1 comprises a set of distributed edge servers that help generate and serve different local pages at the edge based on user requests. Each edge is envisioned to serve a ‘region’ (e.g., a city). One proposal for deployment is to place a edge servers at a packet gateway level in a cellular core network.

![Figure 1: A high level view of the GAIUS ecosystem](image)

2.1 Mobile App Markup Language (MAML)

In GAIUS, we use a novel way to define content, we call it Mobile App Markup Language and refer to it as MAML in the rest of the paper. In regular HTML pages, the complexity of rendering a web page after the downloading of the first index.html is dictated by the different DNS resolutions, redirects and inherent complexity in the object dependency tree embedded in the web page. In contrast, MAML removes this complexity by simplifying the way a web page is written. This makes the rendering of the MAML pages extremely simple and enables prioritization of object downloads.

To retain functional compatibility, MAML pages must be similar in look and feel as conventional HTML web pages. Consequently, the current MAML specification supports six different object types: image, text, video, rectangle, text-field, and button. For example, the MAML image object represents the mapped HTML image tag, which contains the URL of the actual image that needs to be prefetched from the network. The image object has the current attributes: URL, x and y coordinates, width and height of the image, and an optional hyperlink in case the image is clickable.

MAML object specification incorporates styling and interactivity, thus eliminating the need for JavaScript and CSS. This helps cut down the size of the page and reduces page load time. Upfront specification of web objects in a MAML page enables prioritization of downloads and eliminates recursive requests while rendering a page. The flattened tree structure simplifies the object loading, object parsing, and the object rendering, thus enabling faster page rendering.

2.2 GAIUS Channels

The legacy Internet relies on the notion of Uniform Resource Locators (URLs), where users type in the URL in a web browser to access the associated page. This requires the browser to do a DNS lookup to map the requested URL into the IP address of the machine that hosts the information. GAIUS defines a new way for users to access and search for content, that is the notion of information channels. A channel is associated with a content provider and contains specific information that they want to disseminate to users. A channel can contain multiple sub-pages each with a unique topic or sets of topics. These topics are used to help users search for relevant information. Users can also subscribe to channels to easily view newly generated content.

The local ecosystem takes input from different stake holders (i.e. content providers, global and local advertisers, and users) and implements on-the-fly page creation. Figure 1 shows the inputs of the GAIUS ecosystem. From these inputs, GAIUS generates a web page and any associated advertisements for the user. The user’s interface consists of a mobile application which enables user to connect to an edge and request web pages and information channels. The GAIUS app displays the web page using MAML instead of JavaScript and HTML.

2.3 GAIUS Mobile App

Mobile phones and mobile-broadband are among the top used access technologies for users in developing regions to get on the Internet. While GAIUS leaves the door open for content providers to optimize their pages, GAIUS also gives users more control over their web experience and allows them to: (i) view and search for information channels and content, (ii) control the fidelity of the
 requested content, thus controlling their data consumption, (iii) create their own local content and distribute it within the community, and (iv) advertise within their community at low cost.

Content Creation: The GAIUS ecosystem changes how content is created by allowing users to generate their own content using predefined templates that do not require significant technical knowledge. Templates can be filled with text, images, and videos through a simple WYSIWYG interface, where users can create their own information channel hosted at the nearest GAIUS server. Once the channel is published, all users within the community are able to browse through this generated content. Since all of the local content is hosted nearby, the hosting and bandwidth costs can be minimized. GAIUS relies on the local ad marketplace to cover the costs of local content.

2.4 Content Policy Specifications

One of the motivations behind the GAIUS ecosystem is to serve users different versions of the same content dictated by policy inputs. For example, based on the user budget, he/she can request a version of a page that is optimized for lower data consumption. However, user-side economics is not the sole determining factor, a number of additional parameters also play a role in page creation. The page layout and the content dependencies are typically determined by the content provider and the priority of the content. Hence, content providers specify the overall layout of the page as well as how the content is related and structured within the page. On the other hand, advertisers determine which ads need to be displayed to the user based on their own economic considerations (e.g., relevance to the users, revenue, advertisers budget). To fast page loads, the users’ network conditions also play a role. All of these factors are combined to produce the final page.

2.5 Advertisements

GAIUS is a collection of highly localized ecosystems where each ecosystem has 100-100k users, so our ad model is completely different from conventional content providers. The GAIUS application allows individual local users to easily publish their own advertisements in form of videos, images or text from their phone. The ad is then added to the local repository to be displayed to users within their community. In GAIUS we implement the aggregators logic (how relevant and ad is for a user) as part of our edge logic. At the edge, ads are chosen based on location, content providers input, and users interests. Unlike conventional aggregators that collect a fixed fraction of ad revenue generated [13], we plan to introduce pricing that won’t burden the local advertisers.

3 IMPLEMENTATION

3.1 GAIUS Edge Server

The GAIUS edge server is based on the Apache Server implementation, using the the Common Gateway Interface (CGI). When a specific user requests a MAML index file, Apache redirects the request to our GAIUS policy engine implementation. Based on various specifications, the policy engine creates and delivers a custom version of the requested page. In addition, the GAIUS ad selection policy is responsible to mix the local content with relevant local advertisements from the ad market place.

The edge server maintains a list of the information channels it hosts. This is used to answer user queries about the available information channels. Based on a user’s request, a specific version that matches the user preferences is served to the user. In our prototype we emulate content providers by creating multiple versions of several popular websites, where each version contains different image sizes by manipulating their resolution. Figure 2 shows two versions of the CNN web page with different fidelity levels (high vs low resolution). The figure also shows the overall page size that is significantly smaller for the lower resolution version, i.e. about 80% reduction in size.

2.4 Content Policy Specifications

One of the motivations behind the GAIUS ecosystem is to serve users different versions of the same content dictated by policy inputs. For example, based on the user budget, he/she can request a version of a page that is optimized for lower data consumption. However, user-side economics is not the sole determining factor, a number of additional parameters also play a role in page creation. The page layout and the content dependencies are typically determined by the content provider and the priority of the content. Hence, content providers specify the overall layout of the page as well as how the content is related and structured within the page. On the other hand, advertisers determine which ads need to be displayed to the user based on their own economic considerations (e.g., relevance to the users, revenue, advertisers budget). To fast page loads, the users’ network conditions also play a role. All of these factors are combined to produce the final page.

3.2 GAIUS Mobile App

For end-users we built the GAIUS Android app shown in Figure 3. The app has a simple user interface with multiple features: channel browsing, content creation/editing, ad creation. The channel browsing feature displays the different information channels to the user sorted based on a mixture of the content proximity to the user and the content popularity. The list displays the channel title and icon for each information channel. When the user clicks on a specific channel, the app requests the corresponding information channel. This can lead to one of the following: if the channel consists of a single page then the corresponding MAML page is sent to the user and is rendered by the app browser; on the other hand, if the information channel consists of several sub-pages then a list of sub-pages is sent to the app from the edge server and is displayed to the user to choose from. The GAIUS app user is able to set his own fidelity level preference between three different levels: high, medium and low. Upon setting the fidelity level, the corresponding image quality from the edge server changes. In addition, the edge server changes the type of the ads within the requested
MAML pages, from video ads at the high fidelity level, images at the medium fidelity, and plain text ads at the low fidelity level.

Figure 3: GAIUS Android application interface

4 EARLY DEPLOYMENT

In this section, we discuss our early deployment experiences. We deployed the GAIUS ecosystem in several communities in different countries. We specifically targeted small scale deployments in select communities. The deployments consisted of a single GAIUS server, where the content was hosted, and a number of smart-phone users from different communities who volunteered to test out our system. The users were asked to download the GAIUS Android application and start using it to browse various channels, and to create their own local content and advertisements. We currently have about 38 active users that span across 10 different countries. The server hosts more than 55 information channels, some of them were created by us, while others were created by our users.

To inspire the newly joined users and in order to have local relevant content for them to browse through, we pre-populated our server with local content specific to user locations. This initial content was created by translating popular websites and RSS feeds within the users’ region. The content was chosen from a wide variety of categories ranging from news, sports, entertainment, events, food, etc.

To evaluate the current deployment of the GAIUS ecosystem, we record statistics for each individual page request in GAIUS. The following statistics were recorded: the overall page load time, the overall page size, the page fidelity requested by the user, the user’s geographic-location, the user’s network type and mobile phone model.

Figure 4a shows the page load time box plot for the overall page requests as well as for the different individual fidelity levels. The x-axis shows the fidelity level, whereas the y-axis displays the page load time. It can be seen that the median page load time is about 1.6 seconds, with the 75% not exceeding 3 seconds. In addition, it can be seen that the “high” fidelity level has the highest median page load time, in contrast to the “medium” fidelity which has a lower median value, and the “low” fidelity having a median page load time below one second.

Figure 4b shows the page size box plot again overall and across the multiple fidelity levels. The figure clearly shows how effective GAIUS is in reducing the overall page size with a median smaller than 40kB. The figure also shows how the medium and low fidelity manages to reduce the overall page load time by more than 6x.

Figure 5 shows the distribution of how page load times varied as a function of network used to access the pages. The lowest page load times were experienced by users connected over WiFi which has the lowest median value. Looking at median values, we can see that users connected via 4G and 3G networks experienced page load times slightly more than WiFi but still in the acceptable range of few seconds. The 2G network had the worst performance and had a page load time of around 10 seconds but this is still an improvement over status quo.

In summary, this paper shows that the GAIUS ecosystem can improve web experience in emerging regions by providing efficient mechanism for localized content creation and diffusion. GAIUS enables this by leveraging a mobile application and a novel content specification language. We observed that GAIUS content loads faster than conventional web content under various network conditions.

Figure 5: Box plot showing the distribution of page load times for different network types.
ACKNOWLEDGMENTS
We thank the Cisco Research Grant for partially supporting Lakshminarayan Subramanian, and Talal Ahmad on this project. We would also like to thank the anonymous referees for their valuable comments and helpful suggestions.

REFERENCES