



EE Podcast #90 Caching with Aaron Waldon

[Music]

Lea Alcantara: You are listening to the unofficial ExpressionEngine Podcast Episode 90. Today we're talking about EE caching with special guest, Aaron Waldon. I'm your host, Lea Alcantara, and I'm joined by my fab co-host...

Emily Lewis: Emily Lewis

Lea Alcantara: This episode is sponsored by the Responsive Web Design Summit. Check it out at RWDSummit.com. Use the discount code EEPodcast for a 20% off your ticket and go beyond Google searches and those incomplete, out of date online tutorials. Bring the experts to your desktop with the Responsive Web Design Summit, RWDSummit.com.

Emily Lewis: The ExpressionEngine Podcast would also like to thank Pixel & Tonic for being our major sponsor of the year. [Music ends] Hi Lea, how is it going?

Lea Alcantara: All right, I'm kind of annoyed at the weather right now because it feels like I was hinted with a little bit of spring in town and then suddenly it's back to winter.

Emily Lewis: Oh, I'm lucky it's absolutely gorgeous here in Albuquerque. We got perfect weather. Grrr.

Lea Alcantara: [Laughs]

Emily Lewis: I have to say it's been, what, a week since we made the daylight savings times change and my system is still off. Everything about my body, my mind, my sleep, my schedule is just off still.

Lea Alcantara: Yeah, it's the same thing with me, like I guess I appreciate the extra hour in the evening, but at the end of the day I really don't care. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: I just want my extra hour of sleep.

Emily Lewis: Well, I also find it really annoying because like let's talk about daylight savings time in ExpressionEngine real quick. It's got that setting in there that at least for the EE Podcast site, we can't seem to get it to work correctly so we have to like schedule our podcast with a hard coded date or time that's not the actual time just to get it to trigger correctly. It's so frustrating that it's so complicated for something that's been – how long has daylight savings time been around? A hundred years or something back?

Lea Alcantara: Yeah. I know it's definitely annoying in the fact that you have to memorize what's happening. Do you know what I mean? Because if it's hard for us, and we're professional web developers, [laughs], can you imagine our clients if they have to deal with that? I think most of us just avoid even putting that on there. I think for most of us.

Emily Lewis: Yeah, I think that daylight savings time is one of those things that's annoying personally and technologically. I just avoid it at this point and have to adjust my server times in the settings in Expression engine at this point.

Lea Alcantara: What's interesting though, and I'm just looking at Twitter at the #eecms, EllisLab just tweeted this really mysterious image I thought about DST. So I have no idea what it really means. [Laughs]

Emily Lewis: Well, it looks like it's some changes to the config file or something to disable daylight savings time. That's what it looked like.

Lea Alcantara: Yeah, I hope this means that they've figured out how to deal with it. I'm not a 100% sure, but considering that it's only been a week of daylight savings time, and I'm hoping that's what that means.

Emily Lewis: That would be nice. I'd like to not to deal with this again next year.



Lea Alcantara: [Agrees]

Emily Lewis: Or wait six months. Is that when we knew it all?

Lea Alcantara: Yeah, yeah, exactly. [Laughs]

Emily Lewis: So enough about daylight savings time. Let's talk a little bit about some ExpressionEngine news. It's actually not too much in terms of news, Ryan Ireland posted on EE Insider, I guess about a little over a week ago, where he wrote about loyalty, being loyal to tools, and his basic message is that the only loyalty we as designers and developers should be concerned about is the loyalty to our clients, to help them achieve their goals regardless of the tools that we use.

Lea Alcantara: And I totally agree with that sentiment, and we all do this all the time, not just with our software, but with the hardware that we use. I used to be a PC shop and now I'm a Mac shop, and that was just mostly my own decision making based on my user experience and how it was positive to me. But that doesn't necessarily mean that there's anything wrong with the Windows box either.

Emily Lewis: Exactly. In fact, that same week that Ryan posted that article, I did a Q&A for Web Standards Sherpa on choosing the right CMS. Even knowing that I love ExpressionEngine and it is my current CMS of choice, I didn't once in my answer mentioned a specific technology or solution, instead sort of emphasizing the importance of understanding your client and your project. That's our job. Our job is not to pick a tool and stick with it forever because that's not necessarily going to serve our clients.

Lea Alcantara: Yeah, I totally agree.

Emily Lewis: So I think it's a good message to reiterate. I'm glad Ryan brought it up. I agree with it a 100% and I really encourage all of our listeners to focus on client loyalty. Maybe use the tools that work for the job, and that's as far as technology should go.

Lea Alcantara: I agree. That's all I can say about that. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: So let's dive in today's episode.

Emily Lewis: Sounds good. So today we're joined by a special guest, Aaron Waldon. Aaron is the founder of Causing Effect, a brand and web development company that also builds ExpressionEngine add-ons including CE Cache. Welcome Aaron. Thanks for joining the show.

Aaron Waldon: Thank you. It's my pleasure.

Lea Alcantara: Aaron, would you mind telling our listeners a bit more about yourself?

Aaron Waldon: I am a fellow nerd.

Lea Alcantara: [Laughs]

Aaron Waldon: I have a little girl and two little boys and a beautiful wife, and I am a code monkey, so that's now sums it up.

Lea Alcantara: So how did that all get started? Were you always working on ExpressionEngine, or how did that all begin?

Aaron Waldon: Actually, I went to college for business, and yeah, I got my degree, my bachelor degree in international business management, and I had a little bit of time between when I walked and like when I finished my main courses for my degree and kind of finished out my degree. So I'd always kind of played around with programming and I knew that I kind of liked it, and so I took several programming classes and I just never really looked back. I've just kind of been doing programming ever since. So I basically finished my degree and then was just pushed straight into programming.



Lea Alcantara: Interesting. So when did you first start playing around with ExpressionEngine?

Aaron Waldon: I first started using EE sometime in 2008.

Lea Alcantara: And what was the impetus for you to start using EE?

Aaron Waldon: I was actually doing some work for an agency on the East Coast and there is a guy that was really excited there about ExpressionEngine, and so I kind of was thrown into the whole thing. Do you know it was like that? And I'm like, "Oh, it's pretty cool." But nobody even knew me in the community, and so started writing ExpressionEngine add-ons about two years ago. I wasn't really – I don't know. I never really posted on anything. I just kind of used it as a silent user.

Lea Alcantara: So a lot of developers create add-ons, but not everyone actually releases those add-ons. So what made you decide, "All right, I've put all this effort in here. Let's see what happens out in the wild."

Aaron Waldon: Oh, it's an interesting question, and I have more add-ons that I've written that I haven't released than I have.

Lea Alcantara: Interesting.

Aaron Waldon: But there was an article that Bjorn. I hope I'm saying that right. I've even met him, but...

Lea Alcantara: Bjorn?

Aaron Waldon: Yeah.

Lea Alcantara: [Laughs]

Aaron Waldon: Whatever his name is, you know who you are, and everybody else does too. He actually wrote an article about like selling add-ons for ExpressionEngine and how it's favorable because the licensing is good. It's not GPL like WordPress and stuff like that, and I thought, "You know, I actually have some pretty cool things that I've done and I might give this a try." And so I had thousands of shortcomings like with image sizer, an add-on that I have used a lot of the time, and I'm like, "You know, I can start selling this add-on that I've made, that kind of is more reliable and does some more things." I really sit it for some meagerly amount of money like \$8 or something, and I didn't ever get rich quick off of it like I was hoping, but it's been fun and I've met a lot of awesome people all over the world, so it's been a good experience.

Emily Lewis: Well, and users seem to like your add-ons. In fact, you've won several of the Devot:ee Academy Awards for your add-ons.

Aaron Waldon: I spent a lot of money for those awards.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Aaron Waldon: No, I'm kidding. That's probably not a good rumor to spread. We actually have a tremendous community. I've fairly enjoyed being a part of it, and I have made some good friends literally all over the world that I talk to regularly, and it's been a lot of fun.

Emily Lewis: So let's segueway to the focus of today's topic which is caching. First, let's start with the basics. What is caching in ExpressionEngine?

Aaron Waldon: Well, essentially, ExpressionEngine does a lot of work to create a web page from the template. There's a lot of stuff that's going on. It's parsing variables, conditionals, sometimes PHP if you're using PHP in your templates. It's processing the plug-ins, model tags, all sorts of stuff. Obviously, every time somebody goes through your website, all that stuff is having to be created from your template. From the little tags you put in, those turn into the web page, and takes a lot of overhead, a lot of processing, and what caching does is it comes in and it says, "Okay, you know what, we're going to process this one time and then we're going to basically put it on the shelf. We are going to store it somewhere, and then



the next time that somebody comes to this web page, we're just going to grab what we've already processed the last time and we're going to give them that instead." So in some cases, it will make the page tremendously faster and pretty much all cases, that should make your page faster.

Emily Lewis: Now, in terms of caching in ExpressionEngine, what are the different types of caching that you get just out of the box?

Timestamp: 00:10:00

Aaron Waldon: Natively, ExpressionEngine offers a few different caching options. The main ones are really like tag caching and template caching. So with tag caching, the tags in ExpressionEngine, obviously, are like the little exp:whatever (channel entries or whatever). So with native tag caching, you can actually just specify a number of minutes that you want to cache the results of that single tag. So if you've got a channel entries loop, you can say, "Okay, you know what, instead of running this every page load, we're going to go ahead and cache this for the next hour or 60 minutes." So any users that visit the site during those 60 minutes will just get what was, going back to the original analogy, put on the shelf earlier.

Emily Lewis: [Agrees]

Aaron Waldon: And then after 60 minutes, some unlucky user will land on your website and they'll get the whole page load time and then it will be put on the shelf again, and everybody else for like 60 minutes will get that.

Emily Lewis: And that's just set with parameters, right?

Aaron Waldon: Yeah. So it's pretty easy. Just to cache equals yes and then refresh equals and then however many minutes you want to do it. So like refresh equals 30 or something like that.

Lea Alcantara: So why would you do tag caching versus just caching the entire template?

Aaron Waldon: Well, with tag caching, you can kind of just pick and choose like what tags you want to cache really. So in other words, you could keep like maybe some plug in or some add-ons, content dynamic, but you can still cache maybe some channel entry loop in your footer or something like that. Pretty much anytime you cache the entire template, it's going to be faster because you probably have multiple tags. It's having to basically get less stuff off the shelf. That's generally faster, but not as flexible.

Lea Alcantara: So do you believe that these native options are enough for most sites?

Aaron Waldon: That's a good question, and it kind of just depends on what enough is and what you can squeeze out of that. I mean, that's going to be kind of an arbitrary limit. It's going to be different for most sites, and also I mean, I don't know. I think some enterprise sites are obviously going to care a little bit more than like mom and pop shop down the road.

Emily Lewis: When you're looking at a site and you're trying to evaluate performance, what do you go through to decide whether if you're going to use something natively from EE, how do you decide whether you're going to do the whole template or just individually select the tags?

Aaron Waldon: Based on what you want to keep dynamic. If you can't cache the whole page, great. If you need parts of your page that are dynamic, then you are basically limited to just native tag caching with ExpressionEngine's native offerings.

Emily Lewis: Yeah, you had mentioned like in enterprise level site, so is that the kind of site that would need to have a closer look at caching because, I guess, it has more traffic or more data?

Aaron Waldon: Caching, you should use it whenever you can. So that's really the question because I think there a number of people that have the attitude or they consider caching to be a remedy for like a poorly written site, and I see people throw the term around like band aid like caching shouldn't just be a band aid, and caching definitely can be a band



aid for some sites, but it doesn't mean that a well-coded site doesn't need caching. So if you think about it, like why make your site's visitors wait to process information over and over again for every page load?

Emily Lewis: Right.

Aaron Waldon: Like why make them wait at all? I mean, if you don't need to process that stuff on every page load, you're just going to make the experience faster and better for your site's visitors.

Emily Lewis: Yeah, so it's something that, like you said, every site can benefit from it. It's not just something for like a high traffic site or really massive, huge corporate site.

Aaron Waldon: I think for myself, if a page is loading in more time in like half a second, you might want to consider caching. On a normal page, when you visit a web page, the first thing you're going to get back is the response for that page. So if it's normal HTML page, you're going to have to wait to get that page back from the server. The browser is going to have to wait to get that page back from the server before it can then even make any other request. It's going to have to get you HTML back and then it's going to be like, "Oh, you know what, they have these many style sheets in this JavaScript, and now we've got to go and get those things and we got to go and get these images." None of that is even going to start until it gets the HTML. That's important, so when your page is taking two or three seconds, I mean, people are just visiting your site and they are just kind of sitting there and staring at their browser, and then when they get the HTML, then it starts making those other requests, and then when it gets those other requests, then your DOM gets ready. DOM is loaded and then you can start using the page. ExpressionEngine where like template caching is set, that's definitely one part of performance, but then there's a lot more front end and server stuff that you want to consider too.

Emily Lewis: [Agrees]

Aaron Waldon: And those are pretty big topics.

Emily Lewis: Right

Aaron Waldon: So we can probably save those for another day.

Emily Lewis: So this probably is a dumb question, but I imagine it's something that I would have tried when I was first working with ExpressionEngine. Is there a situation when you would have template caching and tag caching in that template, or does the template caching override and sort of control everything even if you had individual tag cache parameters?

Aaron Waldon: If you're using the native options like you're using native template caching and native tag caching, there's really no need to do both because if you're already caching the entire page, you don't need to cache every individual tag. In actuality, you're probably creating a little bit more work if you cache every tag on the one time the page loads like every 30 minutes or whatever. There's probably like a fraction of a second more work because it's actually having to store those cache first. It's having to go and put those things on the shelf when it doesn't really need to.

Emily Lewis: Okay.

Aaron Waldon: Because the whole page is already going to be cached. So yeah, if you're using like full page caching, you don't really need to worry about tag caching for that page.

Lea Alcantara: So the next question would be once you've exhausted all your options with the native caching, at what point do you look at other caching options for EE like, for example, your own add-on or Template Morsels or anything like that? At what point do you say, "This isn't enough?"

Aaron Waldon: So you're assuming you want to start with the native options first then?

Lea Alcantara: Interesting, interesting, so...

Emily Lewis: Oh, is that now what you do?



Aaron Waldon: [Laughs]

Lea Alcantara: [Laughs]

Aaron Waldon: I'm just playing with you.

Emily Lewis: [Laughs]

Aaron Waldon: I mean, a lot of times, that could get you by, and again, it just depends on your client expectations and how fast you need the site. But like with CE Cache, the page load is going to be tremendously faster than the native options like with static caching. So for example, if you go to like the Causing Effect website and you look at the view source, the time is at the very bottom of that page, and so I just loaded it and the page was basically served in point 0.002 seconds, which is pretty fast, then the page actually got to me. I look in the debugger thing here. I'm not very good at this, the development stuff, give me a second.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Aaron Waldon: So the page actually got to me in like 81 milliseconds, which I mean it's fast. If you click to the site, it feels like things are just already at your fingertips. You don't have to be waiting around, and that's obviously a really nice thing to have. So I mean, some of the things that, and I'm not saying that CE Cache is the only add-on that can make this magical speed boost happen, although it is pretty darn good. But caching can be a tricky thing because like we're talking about before, like if you're using the native tag caching, it's kind of like lottery. If you're the person who hits it every 30 minutes or how it often refreshes, you're still going to get the full force of that page. You're still going to be kind of heavy for you. In some pages, that's okay. I've talked to some people who they have like a map that's being generated with ExpressionEngine add-on that will have like 10,000 data points. That page has to be cached or it takes 20 seconds to load. Do you know what I mean? And then with CE Cache, it will take, if they are using static caching, it will be super fast, and if not, it will take maybe a second or something just because the page is so huge to transfer.

Emily Lewis: You've mentioned static caching a couple of times. I'm not familiar with what that is. Can you explain all of it?

Aaron Waldon: Static caching is basically full page caching, and true static caching, it's basically like a flat size. It's almost as if you had coded just an HTML file and weren't using a CMS.

Emily Lewis: [Agrees]

Aaron Waldon: So like if your server gets a request and instead of like having around drew up the whole process of like, "Okay, what URL do we route through. Load all the CodeIgniter files. Load all the ExpressionEngine files. Do all the template parsing." Instead of doing all that, it just says, "Oh yeah, look, here's the part we need, send it back."

Emily Lewis: Okay.

Aaron Waldon: So it's fast. CE Cache, it has a couple of different ways of doing that, and I know that SolSpace has a static cache add-on as well called SolSpace Static Page Caching. But essentially what these add-ons strive to do is to take the cache content and put them in a file that can basically be quickly served up.

Emily Lewis: Now, I might be putting your on the spot here, but am I correct? Is that sort of how Statamic CMS works, that it turns these files into static files so that they are delivered faster?

Aaron Waldon: I don't really know the caching capabilities of the Statamic, so that might be a better question for Jack McDade. I honestly don't know. I actually haven't used it. Sorry.

Emily Lewis: No, that's okay. I just know that I've heard of a couple of things lately from some folks where that there's a CMS in the sense that the client or that the content authors entering data into a system, but what's shot out or whatever or what's generated is a flat file as opposed to a dynamic file. It's just an interesting direction.



Aaron Waldon: I think the whole idea behind Statamic, and again, I'm probably not the spokesperson for this and I hope that I don't get it wrong, this might be just hearsay. But I think that the whole point of Statamic, it just doesn't use a database. I believe that like when you request the page, it's not necessarily cached. It's still going to create it all dynamically, but it's just doing it all from flat files.

Timestamp: 00:20:13

Emily Lewis: Oh, I see.

Aaron Waldon: So there's no database involved, which can be awesome for some sites like if you want to keep everything in version control because there's no database at all, so basically everything can be revision controlled.

Emily Lewis: When you do have static caching as you were describing it, what are the scenarios where you decide to enable that?

Aaron Waldon: That's a really good question. A lot of times you'll have a site that like if it's a membership site like Devotee or even my site, you can log in and it will say something nice like, "Hi Emily." It will greet the user on every page. In that case, it would be hard to cache the entire page because if Lea went in and visited the site after Emily and it said, "Hi Emily," she might be a little taken back.

Emily Lewis: Right.

Aaron Waldon: She'll be like, "What kind of website is this?"

Lea Alcantara: [Laughs]

Aaron Waldon: So in that scenario, it might be kind of hard to do a full page or a static caching because you wanted to keep part of your page dynamic.

Emily Lewis: [Agrees]

Aaron Waldon: Now, with CE Cache, and again, this whole episode wasn't really meant to be a sales pitch for CE Cache, and remember, you guys asked me about it, to do this episode, right?

Emily Lewis: [Laughs] Yeah.

Lea Alcantara: Yeah, totally.

Aaron Waldon: Okay, but like with CE Cache like you can actually surround your entire template with the "cache it" tags and then you can actually escape certain parts inside your template, so you can say, "Okay, escape this greeting so this part will stay dynamic, but everything else would be cached."

Emily Lewis: Oh, okay.

Aaron Waldon: So instead of doing like tag caching in different places, you can basically cache it all and then...

Lea Alcantara: This one little area.

Aaron Waldon: Yeah, exactly, or multiple areas or whatever, you can keep those dynamic. Another thing that you can do like I do on my own site, I can make my pages statically cached if you're logged out, and then if you're logged in, I use the fragment caching, which is like we are just talking them out like the "cache it" tag where you can cache parts of your page. Does that make sense?

Lea Alcantara: Yeah. Now, that's really practical.

Emily Lewis: Yeah.

Aaron Waldon: And I'm nothing if not practical.



Emily Lewis: [Laughs] But why did you decide to create CE Cache?

Aaron Waldon: Another excellent question. So actually, when I first started my Causing Effect website, I actually coded in CodeIgniter and I was using this, I think by today's standards, it's pretty primitive, caching library by a guy named Alex Bilbie, I believe is his name. It basically just cache everything and it like used like a text file to like remember which files you would cache and stuff, which is funny only to me because I'm a nerd. When I started coding my site over to ExpressionEngine, I really felt the page difference, like the speed difference, and I think that a lot of sites at the time were just pretty slow, but everybody was just like, "Wow, look, our clients can use the site." And it didn't seem to me like a lot of people was really as conscientious about caching as or aware about caching as they are I think now.

Emily Lewis: [Agrees]

Aaron Waldon: And performance just wasn't really at the forefront of anything back in 2008 with ExpressionEngine, at least in my mind. Maybe it was for other people. So anyway, so I was like, "Well, you know..." Basically, I needed a faster site and it was driving me nuts, and so I had sat down and I started working on CE Cache, and I made it, and I was like, "Okay, this is pretty awesome because it dramatically, dramatically could make a page as fast even like the really heavy ones like some of my documentation pages where I was doing a whole lot of stuff. You know, I'm going to release this." And I told my wife, I'm like, "I'm going to release this out add-on and I don't think anybody is ever going to use it." Do you know like... [Laughs]

Lea Alcantara: [Laughs]

Aaron Waldon: Well, I'm just going to throw it out there though because I like it, and I felt like I'm getting really great results with it, and honestly it's surprised me because almost immediately after I released it, like Stan Hambo and other people were like, "This changes everything. This is the best add-on ever." And I got some really great feedback and some really funny reviews to it. If you read them, some of them were actually pretty funny.

Emily Lewis: So when you moved from your CodeIgniter system to ExpressionEngine for your site, did you try the native tag or template caching first and it just wasn't giving you what you needed?

Aaron Waldon: Yeah, I mean, I was familiar with those because I had used them for a long time, but sometimes even using those like for documentation pages. Some of those pages I think I probably have like a hundred tags on, and just because they are all CE Image tags and I also use an add-on to generate the examples and then I also in parsing all my example pages for markdown, and there is just a lot of stuff going on. I'm having to worry about the shopping cart, and there's just a whole bunch of different stuff that's going on in those pages, and so I mean, I think they're coded pretty efficiently but it's just the sheer volume of stuff that's on the page.

Emily Lewis: So you're able to cache stuff from other add-ons?

Aaron Waldon: Yeah, so I mean, you can just basically wrap. I mean, there are several different caching strategies, but yeah, I mean, you can just wrap your insider page basically with fragment caching. So you just do one opening tag at the very top, one closing tag at the very bottom and then you can cache everything between or escape certain parts, or whatever. It's pretty easy to implement early.

Emily Lewis: So with an add-on such as yours like CE Cache or maybe the one from Solspace you mentioned, they allow you to cache stuff from other add-ons where I'm guessing you don't have that with the tag caching with EE.

Aaron Waldon: Yeah, with tag caching, you're pretty much tied to the tag or the entire template.

Emily Lewis: Okay.

Aaron Waldon: And again, I think native caching is awesome, and if you're not getting a more robust solution, you should at least definitely use that as much as you can. There are a few times when it's not a good thing because like why process the entire page and be slower if you don't have to.



Emily Lewis: What do you mean sometimes it wouldn't be a good thing? What would be the situation that it actually makes things slower?

Aaron Waldon: Sorry, that's my speech impediment. It would always be a good thing.

Emily Lewis: Okay.

Aaron Waldon: Yeah. [Laughs] That sucked. I don't really have speech impediment, I don't think.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Aaron Waldon: But I will blame it on that because I'm not very clear or something.

Lea Alcantara: So I feel like we've been talking a lot about static and file-based caching options, and I know looking at your CE Cache page, that there are other caching options available that's more related to the server. For example, Database, APC, Redis, Memcache. I'm not really familiar with any of these. [Laughs] Because my limit of static caching and just caching in general is like, "Let's limit the time before it regenerates again, and let's create a new file that we can download." And that seems like very straightforward and easy to understand, but what are those other caching options and why are they even available?

Aaron Waldon: Really the point of those words is to intimidate people.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Aaron Waldon: And to scare them. They actually don't do anything different. No, I'm kidding, and that's a good question. They are basically just different technologies, so a lot of times your file system will be the bottleneck, like it will be what's slow, and sometimes your database will be what's really slow, and different servers obviously have different capabilities. So with some servers, it takes longer to read and write with flat files, and I mean, have you worked on a site where like the database just seemed super slow, and sometimes you don't really have control over that because your client might already be hosting somewhere, or whatever.

But so like with Memcache and Redis, the less times you can like offload some of that responsibility off of the file system or off of the database and you can just cache on one of those other technologies. They're generally really easy to install, not all hosts will let you install them, but most of the time, I mean if you have like command line access, it's seriously like a two-minute thing, and you don't really have to do anything different to use those. In your config, you just say, "Driver equals Redis or Memcache, or something like that."

My personal favorite is Redis. It's just lightning fast, and basically, so like the difference with like Redis versus the Database or with your files or something, and the same with APC and Memcache and those is that it will actually store that information in memory, and so instead of it actually having to go find the file, open file, read the file, it basically go and find the shelf, take it off the shelf and then serve it up. It basically is like, "Oh yeah, it's right here. It's in my hand already. Here you go." It's very, very quick and efficient.

Lea Alcantara: You did say it seems pretty simple to install as long as the host allows it to. What kind of host allows that, generally speaking? Is it mostly like if your client is on a shared server like one of those lower monthly costs \$10 a month thing, most likely you won't be allowed, or is it mostly for VPS clients?

Aaron Waldon: If you don't have command line access, or if you can't like get in the terminal and install stuff yourself, they probably won't allow you to do it, but I mean, there are like some hosts that will let you like I know that EngineHosting,



I think they offer Memcache, and then I know that like Nexcess. Nexcess actually has CE Cache bundled in, in one of their plans, which is awesome. They actually offer several different things too, like you can actually get plans with Redis for the higher end plans. They integrate with CE Cache quite a bit.

Timestamp: 00:30:03

Emily Lewis: And just sort of reiterate what you just said or to clarify because we did get a listener question from James Askew, and he asked what the actual difference of Memcache compared to file and database, and if I understand you correctly, it's basically like Memcache and then I think you also said Redis and APC or this way where it just stores that information in memory as opposed to a file. That's the main difference?

Aaron Waldon: Yeah, that's the main difference, and there's different technologies. They do have their differences, but basically they are just whatever technology is used to store and retrieve the data. I mean, it essentially is what it comes down to. So like what's cool about using like Redis or APC or Memcache besides them being in data and super readily available is that you're also not burdening your file system as much. You're not hitting the database at all to get those things quickly.

Emily Lewis: They are just fundamentally faster because of how they're structured to work?

Aaron Waldon: That's like their job is just to store key value data.

Lea Alcantara: So is there a situation ever where this is really overkill?

Aaron Waldon: Yeah, like if you have a blank page.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Aaron Waldon: I mean, that just really depends on what your expectations are and what your client's expectations are. There are some people that honestly they are perfectly happy if their page loads on a second, they are fine with it. There are some people that was in three seconds, they are fine. Some people don't care at all. If the page loads, they're really excited. But some clients, they are like, "This page has to be fast," and people cared.

Emily Lewis: Is that something that when you're working with a client and that you discussed before a project starts, or is that usually something that comes up through testing and as they are working with different pages and then they start to notice performance?

Aaron Waldon: I think it normally is something that happens towards the end of the project. I actually normally don't even put in CE Cache while I'm building a site.

Emily Lewis: Oh really?

Aaron Waldon: Yeah. I'll just build it and then afterwards I'll just drop in the tags. Another thing that's kind of cool on CE Cache is that you can set up cache refreshing. You can actually set it up so you can say like, "Okay, you know what, if I change my CE Cache home page or if I change my CE Cache add-on at all, I want to completely delete the caches for all my CE Caches on my site and then I want to have a refresh themselves right after." So if somebody goes to my site like right after I delete my cache, they are still going to get the fastest version because it refreshes itself as fast as it can. They don't have to wait, like the unfortunate soul who would land on it 30 minutes after the cache broke like with native caching and had to wait for the whole page load, you can more or less avoid that completely. I'm ashamed to say, I mean, there are times when I'll build a website and I will want it faster and like the client won't want to spend like two hours on improving the performance speed or they just don't care, and so, well, you know what, that's fine if you want to invest in a faster site on your shared hosting server that's really slow, then that's okay.

Emily Lewis: I think that's true for anything, especially with ExpressionEngine because it's so flexible and you can do so much with it that not all clients want to pay for the time to get everything that they could benefit from.



Aaron Waldon: Yeah, it can be frustrating, but at the end of the day, whatever makes them happy, I guess, [laughs], is why we're there.

Emily Lewis: Right.

Lea Alcantara: So I'm curious if there's ever any scenario where you need to use CE Cache in conjunction with other add-ons like are there any other pieces of software out there that might enhance it or work together with it, or et cetera?

Aaron Waldon: I think for the most part, CE Cache kind of just plays somewhat independent of other add-ons, like it's just caching like the entire page or parts of the page. But I mean, there are some add-ons, I can't really think of any that enhance it, but there are definitely some that kind of have some difficulties to work with, and I do my darnedest to give compatibility with most add-ons and the few that kind of have problems with it, like ones that make changes to your site's content, but not through entries.

So like an example, that would like Structure. It's like with Structure, if you like rearranging like your navigation order, there's like no hook or anything that says, "Hey, we just changed the navigation. CE Cache, why don't you go ahead and refresh your items or whatever." So I mean, if you edit an entry and update, CE Cache would be able to be notified and then it can do its thing.

But with that, even with that, they're like there's a Twitter user, Andre, who his username is @nDre. He wrote a blog post on like how easy it is to actually get the two to play together, and there are several little add-ons like that and there are some add-ons that have actually like NavEE, they added support for CE Cache like it comes with it, and I know that they use CE Cache on a lot of their projects. Like Matt Fordham, he also made a little add-on that would allow Low Variables entries to break the cache and stuff like that.

There are some thing that it takes a little bit of work, but I'd say the most complicated part about putting in CE Cache is actually just setting up your cache breaking so that you say, "Okay, whenever I updated entry, I want this and this and this to clear and this and this to happen." Some of that can be made pretty easy too because you can actually tag your entries so you can say like, "Okay, anytime an entry in my blog channel is updated, I want every cache on the entire site that has a tag of blog I just wanted to clear and refresh." And it can...

Emily Lewis: So it could get really granular then.

Aaron Waldon: Yeah.

Lea Alcantara: Cool.

Aaron Waldon: Yeah, I aim to please.

Emily Lewis: [Laughs]

Aaron Waldon: So...

Emily Lewis: Now, have you ever used any other caching add-on before creating your own?

Aaron Waldon: I haven't. I know there are some that are out there. Like I know that Isaac, but I don't know his last name, @airways, he has a cache tracker add-on. I think that kind of extends the native functionality quite a bit and gives a lot of different options. Solospace has Template Morsels, which I think it's kind of like fragment caching with CE Cache, but I could be wrong about that. I might be getting a lawsuit on my hands soon for this, but I don't think you actually implement it inside the template. I think it's something that you actually will implement more like in the back end, like you put in what



content you want to get processed, and then it will be just available in the template. Again, I'm probably speaking out of turn.

Then ExpressionEngine's own, Kevin Cupp, he's written an add-on called Purge. That's pretty interesting because it integrates ExpressionEngine with Varnish which is a really fantastic caching technology.

Emily Lewis: And that resides on the server. I think Nevin Lyne talked about it when we had him on for our past episode.

Aaron Waldon: Yeah, yeah. Varnish is a server technology. The thing that's kind of tricky on Varnish is that it's kind of another one of those things that all or nothing. So like if you have a membership site, it's going to be kind of hard to use Varnish on that. So if you have a completely flat site where your page is hardly ever need to be refreshed or the entire page can be cached and just served up whenever, then Varnish is a really good solution. You might need like a server administrator to help set it up though, but Varnish is really good stuff. Someday I might try to integrate it with CE Cache.

Lea Alcantara: All right. So we got a lot of info from you today. I'm so much more informed now and I'm going to be looking at my performance very soon. [Laughs] So before we let you go, do you have any final words of advice for anyone considering caching their sites?

Aaron Waldon: I think caching is a good thing, and again, like even if you're not going to go with one of the commercial options, whenever you can cache, it's almost always a good idea to cache contents so you're not constantly processing things and having a slower site. I'm sorry if this whole podcast came off as a CE Cache infomercial.

Lea Alcantara: [Laughs]

Emily Lewis: Not all. [Laughs]

Aaron Waldon: But that's kind of like my experience with the whole thing. So thanks for having me on.

Lea Alcantara: Perfect. Thanks for joining us, Aaron.

Emily Lewis: In case our listeners want to follow up with you, where can they find you online?

Aaron Waldon: You're welcome to email me at software@causingeffect.com, or just submit from the causingeffect.com contact form. I'm also on Twitter [@causingeffect](https://twitter.com/causingeffect). I don't really do support request on Twitter, but I'm always there if you want to chat or have a general question about caching.

Emily Lewis: Great. Thank you so much, Aaron.

Aaron Waldon: My pleasure. Thank you.

Lea Alcantara: [Music starts] Now, we'd like to thank our sponsors for this podcast, Responsive Web Design Summit and Pixel & Tonic.

Emily Lewis: We also want to thank our partners, EngineHosting, Devot:ee and EE Insider.

Lea Alcantara: And thanks to our listeners for tuning in. If you want to know more about the podcast, make sure you follow us on Twitter [@eeepodcast](https://twitter.com/eeepodcast) or visit our website, ee-podcast.com.

Emily Lewis: Don't forget to tune in to our next episode when Lea and I will discuss the user experience in ExpressionEngine. Be sure to check out our schedule on our site, ee-podcast.com/schedule for more upcoming topics.

Lea Alcantara: This is Lea Alcantara.

Emily Lewis: And Emily Lewis.

Lea Alcantara: Signing off for the unofficial ExpressionEngine Podcast. See you next time.

Emily Lewis: Cheers. [Music stops]