HTP://www.dotnetrocks.com

Carl Franklin and Richard Campbell interview experts to bring you insights into .NET technology and the state of software development. More than just a dry interview show, we have fun! Original Music! Prizes! Check out what you've been missing!

Text Transcript of Show #455
(Transcription services provided by PWOP Productions)

Paul Randal on Developers and Databases
June 16, 2009

Our Sponsors

Geoff Maciolek: The opinions and viewpoints expressed in .NET Rocks! are not necessarily those of its sponsors, or of Microsoft Corporation, its partners, or employees. .NET Rocks! is a production of Franklins.NET, which is solely responsible for its content. Franklins.NET - Training Developers to Work Smarter.

[Music]

Lawrence Ryan: Hey, Rock heads! Quit wondering if the dark ages were caused by the Y1K problem and listen up! It's time for another stellar episode of .NET Rocks! the Internet audio talk show for .NET developers, with Carl Franklin and Richard Campbell. This is Lawrence Ryan announcing show #455, with guest Paul Randal, recorded live, Tuesday, June 2, 2009. .NET Rocks! is brought to you by Franklins.NET - Training Developers to Work Smarter and now offering DotNetNuke video training with Chris Hammond from Engage Software on DVD, dnrTV style, order your copy now at www.franklins.net. Support is also provided by Telerik, combining the best in Windows Forms and ASP.NET controls with first class customer service, online at www.telerik.com, and by CoDe Magazine, the leading independent magazine for .NET developers, online at www.code-magazine.com. And now, the man who is busier than a one-toothed man at a corn-on-the-cob eating contest, Carl Franklin.

Carl Franklin: Thank you very much and welcome back to .NET Rocks! This is Carl Franklin in New London, Connecticut, and Richard out there in Vancouver. Hey man, what's up?

Richard Campbell: Hey, not much. I got your sunglasses.

Carl Franklin: Oh yeah, I left them in your house. Your house is amazing, dude.

Richard Campbell: Thanks, man.

Carl Franklin: I can't wait to see the Batmobile launcher.

Richard Campbell: Ah, the Batmobile launcher. It's just a lift, you know.

Carl Franklin: Yeah, I know.

Richard Campbell: Just a lift.

Carl Franklin: It's not everybody who has a lift in their garage for their second car.

Richard Campbell: Well, you know, got to put it somewhere.

Carl Franklin: You're excessive. All right man, let's get into Better Know a Framework.

Richard Campbell: I'm okay with that though.

[Music]

Richard Campbell: What have you got for me?

Carl Franklin: All right, well, so we've been talking about -- we've been doing a long series on Better Know a Framework on the System.Windows namespaces.

Richard Campbell: WPF.

Carl Franklin: Yeah, WPF and Silverlight, and we're going to get into System.Windows.Shapes this time.

Richard Campbell: Oh.

Carl Franklin: Very simple. This is where the library of shapes is that can be use in XAML or code. You got the ellipsis, you got the line, you got the path, you got the polygon, you got the polyline, you got the rectangle, you got the shape which is the base class.

Richard Campbell: Nice.

Carl Franklin: Any questions? Didn't think so.

Richard Campbell: I'm thinking like circle?

Carl Franklin: Yeah, pretty straight in. So that's where the shapes are.

Richard Campbell: All right.

Carl Franklin: You know, they can't be all glamorous.

Richard Campbell: Sometimes they just need a rectangle.

Carl Franklin: Man, DevTeach was cool, wasn't it?

Richard Campbell: It was a good little show, wasn't it?

Carl Franklin: It was a lot of fun.

Richard Campbell: Nice folks there.

Carl Franklin: We did a dnrTV on that show. By the way, some really good dnrTVs coming up. We're starting an MVP series which we've done a lot of dnrTV with MVPs but specifically the MVPs are getting involved in the beat, labeled and recognized

as such. So we're doing some very cool things. We've got about five or six or seven shows in the can now and we'll be releasing them a little bit sooner than once a week for a while. So catch up with them, dnrtv.com. Hey, you got an email for us?

Richard Campbell: I do indeed and it's funny that you mention dnrTV because this email mentions RunAs Radio.

Carl Franklin: Oh cool.

Richard Campbell: Let me read it to you. "Hi guys, I would first like to thank you for a great show. I've been listening to you for over a year now and my career has not been the same since then so thank you."

Carl Franklin: Awesome.

Richard Campbell: Yeah. What do you think, he just stop working, he is just listening to the show all the time, he is unemployed?

Carl Franklin: I think that's not what he meant.

Richard Campbell: Okay. "I'm currently building my own house and your voices have been my company while I bang away with a hammer, but in contrary to Richard, I still have some time left on my project. After listening to show 364 with Stacy Harris about Home Automation, my first thought was I've got to do that. I work in a consulting company as a web developer. In our role, we have to know and handle the whole range of technologies: ASP.NET, Networking, WCF, Web Services, Databases, you name it. One subject that I was wishing for is a Performance Tuning show like how to find bottlenecks in things we don't work with everyday like databases, I/O, network chatter, and so on, and I just listen to RunAs Radio with the guest Cliff Huffman and got thrilled about all the things you can discover with just a few tools. Are there more shows like that coming for developers to make our lives easier? Once again, thanks for a great show." From Cal Happe from Stockholm, Sweden.

Carl Franklin: Awesome.

Richard Campbell: I guess we should do more shows in this area. Of course, this is an area that I talk about in conferences all the time.

Carl Franklin: Yeah, yeah.

Richard Campbell: What I should do is do some dnrTV's with you.

Carl Franklin: Yeah, you should.

Richard Campbell: Yeah, absolutely. Walk people through some of these stuff and talk about performance tuning. Certainly we didn't really focus on performance tuning in RunAs Radio, that's much more IT topics, but we did talk about instrumenting web servers and other kinds of servers. Clint Huffman is one of the guys from the premier field engineering team at Microsoft and that's what those guys do, is they work in offices at companies helping them make their apps run better.

Carl Franklin: Speaking of performance, I was just doing a test here with Visual Studio trying to eek out as much performance as I can in a server, the persisting connection server that I'm working on.

Richard Campbell: Right.

Carl Franklin: Highly scalable so obviously performance is paramount to coding. Right?

Richard Campbell: Yup.

Carl Franklin: So we have -- basically I wanted to see where the performance gains can be and I'm using a binary formatter to convert message classes into byte arrays and stream them down through sockets and things like that. So I made up a little test thing to see, well, you know, what is serialization but just making a stream of bytes that represents an object.

Richard Campbell: Right.

Carl Franklin: And if you can do that in a more specific way, maybe perhaps you could squeeze out some performance and maybe some blow. So what I did was I created a class that had like five integers, five strings, and a byte array and then I made a little routine to populate 10,000 of those classes, you know, objects from that class and put them in a, you know, with random data, basically random strings, random bytes, the strings are all in the printable character range, random integers in an array of bytes that contains, I don't know, up to 5K, 25K, something like that, just random sizes. So I did a test using the binary formatter and then I also did a test. I like serialized all 10,000 of these things with the start time and in-time. Then I did a test manually using the bit converter to convert out in two bytes and write all these stuff. In both ways, I wrote it into a memory stream and then cut the array from the memory stream. It turns out doing it manually takes about half the time and cuts out about 40% of the size.

Richard Campbell: Huh.

Carl Franklin: Ain't that interesting?

Richard Campbell: Yeah.

Carl Franklin: So, you know, if you've got a class that you want to serialize and your performance is paramount, just take a look at it. It's not all that difficult to do. It certainly makes it a little more inflexible and you can't, you know, if you want to change your class around, now you have to change your serializer...

Richard Campbell: Right.

Carl Franklin: Because it's specific to that class, but if performance is your thing, hmm, it's interesting.

Richard Campbell: Possiblities.

Carl Franklin: Yup. Hey, you know, our friends in Infusion are still hiring. They're looking for people and we're getting more and more interested parties now. So if you're currently looking for another job and you've got some SharePoint chaps or some ASP.NET chaps or just looking for another gig, they have offices in London and in Dubai and in New York and in Toronto. So they're looking for talented people and that's why they came to me. They said, "Hey, your listeners are pretty smart." Send me an email, carl@franklins.net.

Richard Campbell: Awesome.

Carl Franklin: All right, out guest today is Paul Randal. Paul, of course, has been on the show before. He is the former Microsoft employee SQL Server guru who wrote CHECKDB for Microsoft SQL Server, and currently is an MVP and a Regional Director and works and lives with Kimberly Tripp...

Paul Randal: And is married.

Carl Franklin: And is married, he has a license to do that.

Paul Randal: Yes. That's my only claim to fame.

Carl Franklin: And with SQL skills. Hey Paul, what's up?

Paul Randal: Hey, I'm addicted at the moment, addicted to being online unfortunately. Kimberly is not here, and I'm a Twitter addict.

Carl Franklin: Yeah, Twitter is a time vampire, ain't it?

Paul Randal: My life went down the toilet three weeks ago when I joined Twitter, but that's another story. Actually, it's a pretty good community out there so I'm having a little fun helping people out and finding out some interesting stories of people doing things wrong and stuff so...

Carl Franklin: Awesome.

Paul Randal: So that to me was what's happening, like LEGO and my other sort of hobbies.

Carl Franklin: LEGO, so you're doing Mindstorm?

Paul Randal: No. Actually I have a Mindstorm set that I have never actually got to using it. The story of my life, I see a new toy, oh, let's have that, and then I never do anything with it. So like a model, speak like a model, I like making lego sets from...

Richard Campbell: Yeah, I saw on a twitpic your model of the millennium falcon and it's, what, 4 feet across?

Carl Franklin: Oh my God.

Paul Randal: It's like 2-1/2 feet long. It used to be the biggest set they did. It was about 5-1/2 thousand pieces, and they came out with the Taj Mahal which is 3 feet square and a foot and a half high so it was a lot at that

Carl Franklin: Oh my God.

Richard Campbell: That is a lot of lego.

Paul Randal: It's a lot of lego, yeah. I'm currently making the Death Star that I got for Christmas from the original Star Wars I might add, not the second death star lego model. I don't like that one. There you go, that's my life when Kimberly is not here.

Carl Franklin: Baboom.

Paul Randal: I play with lego and talk to you guys, very sad.

Richard Campbell: That's funny.

Carl Franklin: Hey, before we get into our real topic, there's some seriously cool stuff coming out from Microsoft lately, Bing, bing.com.

Paul Randal: I've heard them say that -- what did they say, it's something that's not Google because it's not Google or something, BING.

Richard Campbell: Bing is not Google.

Carl Franklin: Bing is not Google.

Richard Campbell: Recursor acronyms.

Paul Randal: It is a recursed acronym, it's kind of nice.

Carl Franklin: Yeah, I like it. Kind of like LAME. Lame ain't an MP3 codec.

Richard Campbell: There you go.

Carl Franklin: You know, what I like about Bing of course is the suggestions on the side depending on what you're searching for. If you put in a movie title, the first thing that comes up is a listing of local times in theaters. If you put in an actor or something like that, you'll see -- or an author, you'll see like a bibliography, a link to the bibliography or an artist's discography and those links sort of appear on the side and they're usually the stuff that you're looking for. You put in the name of a product, like an electronics product, just something with a manual, you'll get a link to the user manual on the side. Little things like that, just really, really cool.

Richard Campbell: It's an interesting stuff.

Carl Franklin: It is interesting.

Richard Campbell: I think you did this over Twitter but I saw now on your blog that you actually got your SLA feedback around a maximal allowable downtime and stuff like that. I'm sorry, it's very IT-ish but it's interesting to see what people are thinking in terms of what is the real downtime allowed.

Paul Randal: Or, actually it's kind of depressing the number of people that didn't respond given how many people usually respond to my surveys. It's only like 30 people responded and that's probably because most people either don't have SLA's defined or aren't measuring have or have no idea what an SLA actually is.

Richard Campbell: Or don't know what the number is. I mean, they may well have an SLA but they just don't know and I think that's very true of developers that, you know, how many times is the only time it comes up that we have an SLA and these are the numbers, it's when you didn't make them or in the meeting where they said, boy, that was a really sucky weekend.

Paul Randal: Right, it's like do you have a disaster recovery or HA plan. Well, of course not but as soon as your company actually has a disaster, it's the first thing on the CEO's mind.

Richard Campbell: Yeah.

Carl Franklin: Can you guys, you know, this is .NET Rocks!, not RunAs Radio.

Paul Randal: Yeah, but we're on RunAs Radio.

Carl Franklin: So what the hell is SLA? What is that?

Paul Randal: It's a TLA.

Richard Campbell: Nice.

Carl Franklin: Three letter acronym, yeah, I get that one. It's the only...

Paul Randal: Service Level Agreement. So in the IT world you've probably heard it. The two main ones are RTO, Recovery Time Objective, and RPO, Recovery Point Objective. They are how much time you're allowed and how much data loss you have.

Carl Franklin: Yeah.

Richard Campbell: Well, and you know, as much as these are suppose to be IT related topics, I think especially in today's market, a developer who has the sense of the operations of his organization and has a sense of where his company makes money and what the consequence of downtime are is the guy who is going to keep his job.

Paul Randal: Actually, you know what? There are a lot of things that developers can do to screw up the ability of a company to meet the SLAs. So for instance, imagine a developer writes a query that does a single batch of data about 10 billion row tables.

Richard Campbell: Right.

Carl Franklin: Ouch.

Paul Randal: Yeah. So if it gets to 10 billion minus 1 row that's updated, the server crashes, when the server comes back up, crash recovery is going to run and it has to roll back the entire thing before the database comes...

Carl Franklin: Oh, ow.

Paul Randal: Ow. They're not getting five's and nine's out of that one.

Carl Franklin: No.

Richard Campbell: Yeah, you just flushed your nine's down the toilet.

Paul Randal: Yes, you did, yes. Maybe a nine and an eight.

Richard Campbell: But then when you are doing this sort of work, what do we do to get in touch with production server in the first place? I'm always big in presentations saying to a developer, you know what, you don't want access of production servers. In that way, it can never be your fault.

Paul Randal: Yeah, I don't even mean they're actually on the production server. They just write an application but when they test it they don't test with the right amount of scale.

Richard Campbell: Right.

Paul Randal: So when the data table goes from 100 rows which is the test case to 10 billion rows which is reality, it doesn't scale very well in this performance that affects availability and people don't plan that kind of testing. So you'll never find that out until things actually hit the fence.

Richard Campbell: Yeah, it is the sad truth and I wonder how often you run into this, Paul, that you have organizations that don't actually have IT stuff at all, or if they do they're certainly not concern on the database. I'm surprised at how many times I've met a guy who says, "Yeah, I'm responsible for the database in my organization." I say, "Wow. Did you apply for that job?" He says, "No, I was standing closest to the server when the last guy quit."

Paul Randal: Now you're the DBA, congratulations.

Richard Campbell: Now I'm the DBA, yeah.

Carl Franklin: Good luck, involuntary DBA.

Paul Randal: I see quite a lot, mostly in the forums. There was a forum posted a couple of weeks ago where some poor guy had been told, "The DBA just left. You're now the DBA, the server's down, fix it by tomorrow or you're out for two."

Richard Campbell: Nice.

Paul Randal: Absolutely nice. We got it fixed for him with help over the forums. One of the main involuntary DBA things that I see now is SharePoint.

Carl Franklin: Oh yeah.

Paul Randal: You got a SharePoint installation, suddenly you've got an enterprise class SQL Server...

Carl Franklin: Yeah.

Richard Campbell: I don't think people think about the fact that SharePoint is totally SQL Server dependent, right?

Carl Franklin: Yeah.

Paul Randal: Absolutely. Oh yeah, yeah and it does some wacky things. Kimberly has blogged a bunch of times about SharePoint and some of the interesting choices that SharePoint developers make.

Richard Campbell: The guys who wrote SharePoint.

Paul Randal: Guys who wrote SharePoint, yeah.

Carl Franklin: Is 'interesting' word that you would choose to be polite or...?

Paul Randal: I like being an MVP and a Microsoft regional director, so...

Carl Franklin: Yeah.

Paul Randal: Yes, I'm saying interesting. So for instance, GUID cluster keys, okay.

Richard Campbell: Clustering in a non-sequential GUID at that?

Paul Randal: At a non-sequential random GUID, absolutely.

Carl Franklin: Yeah.

Richard Campbell: Ouch.

Paul Randal: Yeah, yikes.

Richard Campbell: The two database geeks know this is painful but let's talk...

Carl Franklin: No, I...

Paul Randal: I can explain.

Carl Franklin: I think I get it. Indexes are mathematical, aren't they? I mean, they're sequential. They need to be sequential.

Paul Randal: Well, you define an index key which means you're defining some borderings to the index.

Carl Franklin: Right.

Paul Randal: And in every road it gets puts in is inserted into the index bases on the key value.

The date is enough to check the records. If you're higher or the key is a random GUID generated say in your client here, then that means every record, I guess, they inserted is essentially a random insert into the middle of an index.

Carl Franklin: Yeah.

Paul Randal: So random insert in the middle of an index, eventually the index pages fill up and they do a thing called that page split which means because the page is completely full, another record comes in that has to be inserted on that page that's where the key says, and if there is no room page splits in half, another page gets allocated, some rows gets moved to range without getting too technical, and you basically create a fragmentation. So you've got couple of pages that are only half full and you've got an index that's no longer contiguous in terms of the order of the page of this...

Carl Franklin: Right.

Paul Randal: And the order of pages if you follow them in logical order, key order.

Carl Franklin: It seems like you might as well not have an index if you're going to use random GUIDs.

Paul Randal: Well, it depends with what you're doing with index. If you want the index to be able to -- if you want to be able to look up a single record based on that key, the index has been half baked. So that's the point of an index.

Carl Franklin: Right.

Paul Randal: It's being able to find a particular record really fast without table scan.

Richard Campbell: You said one other word here that affects all of this as well which is clustered, it's the clustered index.

Paul Randal: Well, the bad thing about it being part of the clustered index is because the clustered index keys are included in every known clustered index record as well because if the crew processor is using a known clustered index to be able to more efficiently get some results for a query, then if the result set has to include more columns than there are present in the non-clustered index, the key processor has to go back to the actual table itself which is either a clustered index or a heap to get the rest of the records. So there's some kind of a linkage between the non-clustered index records and back to base table. So in the case of a clustered index, that linkage is the clustered key itself. If the clustered key is a GUID or at least contains a GUID, then a GUID is 16 bytes, so that's at least 16 bytes of information pushed into every non-clustered index record as well. So it uses a whole bunch of extra space. It actually also has another effect depending on the non-clustered index keys. So matching your non-clustered index key is a date/time and you're inserting hundreds and hundreds of records per second, even thousands of records per second. The minimum time period that a date/time column in 2005 result is 3.3 milliseconds, so if you can actually insert hundreds of records every 3.3 milliseconds, then the insertion point in the non-clustered index essentially becomes determined by the cluster key, which, if it's a random GUID then you're doing random inserts into your non-clustered index too so it's actually a fragmentation in your clustered and non-clustered indexes.

Richard Campbell: So just to summarize here. When I use a non-sequential GUID as my clustered index key, I am slowing down the rate of inserts, period, whenever those things splits so the initial inserts are slowed down.

Paul Randal: Yup.

Richard Campbell: And fragmenting every index in the process.

Paul Randal: Absolutely.

Richard Campbell: So that subsequent queries of anything else are also impacted. Indexes get less efficient. It has significant consequences, but all this only matters at velocity.

Paul Randal: Yes and a lot of this things cause big problems. It depends. My favorite answer is always it depends. Any SQL Server question apart from shrink is it depends.

Richard Campbell: Yeah because the answer to shrink is no.

Paul Randal: Besides autoshrink. Autoshrink is always no never turn it off. But shrink, maybe, but let's not get into that. It's a whole other, you know... It depends what you're doing with the indexes. I mean, some things are bad if you're doing certain operations, some things if you're doing different operations it doesn't really matter. If you've -- oh it's so hard to say, it's like an enormous rat hole, the whole...

Richard Campbell: But you could make the ugliest database in the world, no indexes, no primary keys, nothing but as long as it's only 100 rows and there's only one user, it will be fine.

Paul Randal: Absolutely, which is an unfortunate problem. With so much of developer

testing on SQL Server, it's that the test cases done in any possible way reflect reality.

Richard Campbell: Right.

Paul Randal: Reality two years from now. Like the testing that I'm sure say MySpace or Facebook didn't reflect the fact that they have 10 million users overnight kind of thing.

Richard Campbell: And that's the experience that I think a lot of people have with SharePoint, it's that the initial site works like a hot dam, and then when you really start to get data into it, when the company is really dependent on it because all the things you want to know are now in SharePoint, now it has performance problems and it's just the consequence of you have a significant amount of data and these practices which were relatively painless at low velocity and low volume are now painful at large velocity and large volume.

Paul Randal: Yup. That's a great example to learn from SharePoint, and I'm not trying to use it as kind of the redheaded stepchild, but it is a prime example of an application that was developed seemingly without a huge amount of depths of knowledge about how SQL Server is going to behave under load with the schema that they chose.

Richard Campbell: Right.

Paul Randal: It's that interesting. One of the smaller -- I'm actually spending a day and Kimberly's spending a day on Friday teaching the SharePoint MCM candidates, because there is a SharePoint MCM running at the moment, we spend a day each teaching the SharePoint MCM folks about SQL Server and somebody's problems and the need for database maintenance and kind of enterprise class installations.

Richard Campbell: And just to finish off this whole discussion around the clustering indexes and so forth, so Paul, in your infinite wisdom what is the preferred clustered index?

Paul Randal: There are four things. The clustered index keys should be unique. It should be as narrow as possible. It should be static, in other words never changing and ever increasing.

Carl Franklin: Wow.

Paul Randal: Something like a big identity column.

Richard Campbell: Yes, begin and then the column is always going to be unique, it's relatively narrow, big, I mean 8 bytes, it's static. Once you set it, you're never going to change it and it is sequential.

Paul Randal: That's the thing. So narrow, unique, static, and ever increasing.

Carl Franklin: Okay.

Richard Campbell: And if you're really, really hook on GUIDs, there are sequential GUIDs now.

Paul Randal: You can use that as a new sequential idea and there's also a way of getting it to be able to output the -- you can only use it as a default for a column, but you can actually -- there's a clause for outputs where you can actually get the new sequential ID value back and pass it back to the client tier and then pass it back then to the SQL Server.

Richard Campbell: Okay. What's interesting about the staticness of it, and I've often said this, it's like when you have identity columns, don't ever show them to the user because if you show it to the user the user will want to change this. I learn that the hard way when I had a VP of Sales actually go to my boss and say, "You can't make that customer 413. He's out best customer, he needs to be customer 1." Don't show them the ID.

Paul Randal: Yeah, that's right.

Richard Campbell: It's a mistake.

Paul Randal: Or have a different column.

Richard Campbell: That's what I did, it's I created a new column that lied.

Paul Randal: Exactly, right.

Richard Campbell: I'm a big believer in that.

Paul Randal: Yeah, all kinds of funky choices that the developers can make which have implications. Another one is how do you store your carets or your love data. So do you store it in row or do you store it out of row? And so in row is actually part of the data record itself so when the stored engine reads in the data record it's got the actual character or low value there, or do you store out of rows which means that whenever the data record is read, the low value isn't there and another I/O has to be done to go and get into memory and there's pros and cons to each. In the first case, when it's part of the data row, then obviously it's only one I/O, and in the second case it's multiple I/Os. But in the first case, I mean your data rows are larger and you got less density of information on any particular page. In the second case, of course your data rows aren't large so you get better density. So data row density means you're having to do less I/Os to read more data, you're having to take less memory in the buffer

pool, or buffer cache as it's sometimes called, to have two to three more data. It's only when you actually want low phase that you get the stuff in but that's a huge choice that you have to make and it's very hard to make those kinds of choices...

Richard Campbell: This is a choice that a developer can make very easily because if I'm a developer I know, you know, most of the time I don't need that data so I'd rather go with the lighter weight row and in the few times that I need that data I'll take the extra I/O hit.

Paul Randal: But that's the catch. The developer has to actually know that that's the implications in making that choice.

Richard Campbell: So that matters.

Paul Randal: Without understanding what SQL Server is actually going to do internally and how it's going to store the stuff on this, then you don't know. So it's kind of hard. So there's argument saying why should developers know about this stuff. We had a whole discussion on the RDA list with Mr. Huckaby, Tim, about should a developer really be a database savvy developer. Do they have to be savvy enough to know these kinds of things?

Richard Campbell: Right and there's definitely a culture out there that says, "Hey, you just stored data for me. Here's some data. Go store it. I'll ask for it back later."

Paul Randal: That was the devil's advocate argument that Huckaby was making which is you shouldn't have to know. The SQL Server will just do it, but then SQL Server just does what you tell it to so if you tell it to store data and there's a proper way of storing it to your particular application, then it doesn't know that. It's just...

Richard Campbell: Right.

Paul Randal: The SQL Server isn't an intelligent product. There's nobody inside it that's going to, "Oh, that's what you really mean. Let's do this instead."

Richard Campbell: Although, you know, it can fool you too. I think the query processor SQL Server is a genius, certainly better than any other query processor of any other database I've ever used.

Paul Randal: It's pretty smart. The people that write the query processor, I know most of them, they have --a bunch of them have PhDs in one tiny area of query processing and query optimization.

Richard Campbell: It is a specialty.

Paul Randal: It's very much a specialty.

Richard Campbell: When society collapses, what are those people going to do for a living?

Paul Randal: We'll code query processor for food.

Richard Campbell: Yes.

Paul Randal: Those are quick though, you're not going to last very long.

Carl Franklin: This portion of .NET Rocks! is brought to you by our good friends at Telerik without whose support the show would not be possible. Hey, how many times have you drowned into endless CSS classes just to change the color of a single element of your application UI? How many times have you have to ask your designer to create custom skins so that your UI controls met your company's brand identity? It's time to turn to a new page. Telerik has launched the Visual Style Builder for ASP.NET AJAX, an online application that allows you to visually modify skins or design new ones with point and click. Colorizing a complete skin at once has never been easier. Just move the color slider and all elements will shift their color spectrum accordingly. That's cool. If the colorization is not enough, you can fine-tune individual elements to perfection where you'd want to change fonts and sizes and margins and padding background colors or just about any style property. It's all easy and intuitive to the Visual Style Builder's graphical interface. It sounds incredible so let's go and check it out at stylebuilder.telerik.com. Hey, and don't forget to thank them for supporting .NET Rocks!

Richard Campbell: The big thing I found was that in working with other databases, I'd write a query and get poor performance and so I'd rewrite the query in a different way and get better performance because I get different query plans, and in SQL Server I find no matter how I write the query, I get the same query plan.

Paul Randal: Oh no, you've just been lucky.

Richard Campbell: Have I've been lucky? You tell me otherwise.

Paul Randal: You're just lucky. You've just been lucky. It all depends and you've got the wrong person on the phone. Get Kimberly on the phone. She's the query processing person.

Richard Campbell: Oh yeah?

Paul Randal: Yeah. I just store the data and return it and make sure it's not corrupt. She's the one

who knows more about query processor, but it depends on what indexes you have, it depends on -- actually it's still the best, it depends on what you're indexing strategy is, what indexes you have, it depends on your statistics whether your statistics are up to date. Now if you've got out of date stats then the query processor is not going to make a good job of picking a query and you might have a query that works perfectly well until you try to select an area of the table that has a massive amount of data skew that the query processor doesn't know about and it's not the right date in which case the plan that is chosen might completely stuck.

Richard Campbell: I've talked to folks that run into this particular issue where I run this query on my test machine and it performs well, it's got a copy of the real data, but then when I run it in production I don't get the same results at all.

Paul Randal: There are a million different things. So there's a survey that I'm doing on my blog, I did kind of a weekly survey. If you're not reading my blog, quick, you should read my blog. It's very cool. I'm not advertising, nothing like that, just lots and lots of info and I post like a mad man. So sqlskills.com/blogs/paul, there you go. Anyway, I'm doing a survey. This week's survey is what's the most important thing when performance tuning? So you walk up to a box and it's not performing very well, what do you go for first? And this kind of ties into what we're talking about because all these different things can affect how well a particular query isolates performance in production. So my 10 choices that I want people to think are: 1) I/O subsistent design tuning including write. 2) Server hardware, CPU's memory. 3) Virtualized versus real server. 4) Database physical layout. 5) Table design. 6) Heaps versus clustered indexes. 7) Non-clustered index strategy. 8) Statistics. 9) Application design and code. 10) Database maintenance. Any single one of those apart from your app design and code can be actually the front-end production than they are in test.

Richard Campbell: Right.

Paul Randal: So anyone of those things can affect how production works, and as a developer, unless you're actually testing on something that represents SQL production, you're not going to get the same results which is why you get actually what you just said, Richard.

Richard Campbell: Well, an interesting area that we certainly run into in some of the RunAs conversations that I run out too out in the wild is SAN performance just not measuring up and significantly harming SQL Server's performance.

Paul Randal: Absolutely. So there's something I just learned about about a month ago which is this partition alignment and the problem that happens there. So by default on all operating systems before Windows Server 2008, the default partition alignment is 63 disk walks which is 31-1/2K and most SAN administrators are going to pick a RAID stripe size of 64K which means that you've got a misaligned disk. So every so often we're going to have an I/O that has to three stripes to go to get the data back.

Richard Campbell: Interesting.

Paul Randal: Interesting. Windows 2008 does it properly.

Richard Campbell: It actually stripes it to fit to the SAN block.

Paul Randal: It creates the partitioning offset to be the right one. Now if you upgraded the database to Windows Server 2008, you're still going to have potentially misaligned partitions. There's a great whitepaper that came out that explains all that and there's a slide back and so on, and you can get up to 30% performance improvement by changing this. It's insane.

Richard Campbell: It's a huge number.

Carl Franklin: Wow.

Paul Randal: Yeah and it's not very well known. I didn't even know about it and I'm suppose to be a HA person. The best way to find it is if you go to my blog and look under performance on the category. There's a post, a couple of posts there that says are your disks properly partitioned and stripes and the right cluster side. That's the best, it links to all the different things there. It's well worth checking. You can get a massive improvement. Of course there's something else that can be different between production and testing.

Carl Franklin: What about virtual server? Would you recommend running SQL Server or not running SQL Server in a virtual machine?

Paul Randal: This is something that I'm not an expert on, I'll be upfront. What I've heard from people is that doing things like a production SQL Server in things like VMware, in other words not Hyper-V, doesn't go very well because you're virtualizing the I/O as well. The I/O basically asks it for a software layer which means it sucks. They're okay in test, but again you're not going to be getting the same performance. With Hyper-V, what I've heard is that it's quite different.

Richard Campbell: Because in Hyper-V you can actually assign LANs and NICs to a given VM and it appears as if they're just on a regular machine with that hardware.

Paul Randal: Absolutely. So that's the extent of my knowledge about running SQL Server on virtual machines.

Richard Campbell: The other thing I've looked at is VM is perfectly harmless when you're not at velocity again.

Paul Randal: Absolutely, yeah.

Richard Campbell: I love virtual machine picking up that old NT 4.0 hardware and just moving it into a VM, the whole thing, so you can let that old gear die and let the virtual machine own it, you can move from machine to machine now, it's not a big deal.

Paul Randal: Yeah. So that's one interesting thing that could impact how your system performs in production. If the dev is working on a VM and sees decent performance, then it's no guarantee it's going to be the same thing in production.

Carl Franklin: Something we haven't talked about in a while, a long time actually, is backing up SQL data, making sure that you've got redundancy. I guess there is a way to replicate SQL Servers so that you can have one waiting in the wings if your SQL Server disk blows up and you're down or the machine fries or power supply goes out or something. What do you recommend?

Paul Randal: There's a bunch of different technologies -- okay, I'll just say it depends. How's that, it depends.

Carl Franklin: It depends.

Paul Randal: It depends. Okay, so saying what is the best, what is the recommended HA technology and we're going to RunAs territory again. It depends on what your SLAs are, it depends on your budget, it depends on what your requirements are in terms of uptime, fell overtime, it depends on how much on the actual load your operation is generating, there are a whole bunch of different things.

Carl Franklin: I guess the poor man's method would be to back-up, to do a regularly scheduled back-up everyday to an external hard drive or something that another machine can access. You could just pull up another machine.

Paul Randal: That's the absolute minimum I would recommend and I'm probably one of the most paranoid people on the planet about doing back-ups so you know, I have back-up back-ups of my laptop and all kinds of stuff. I even back-up my blog content onto a drive away from the host just in case something goes wrong with my host. I don't want to have to go and...

Richard Campbell: When I've been consulting, you bring the CTO in or the CIO in and say, well, how reliable does database needs to be, and if it only had a crash last week, they'd say 100%. You know, it's inevitable, they just throw that number out there, and then when you actually start pricing out a clustered infrastructure, you just call it the hot failover option. So here's a system that the only way to be that fast, to be up instantly if something fails is to have the computer do it itself and that's a hot failover and that's this much money roughly versus a warm failover solution, something where a person has to realize it's failed and switch if for you that's this much money and you look at something like log shipping or replication or any of those alternatives.

Paul Randal: Yeah, I mean money is usually one of the main things that come into play both in terms of what's your actual budget for buying stuff and then what's your budget for space, for power, for H-back, for people to run it.

Richard Campbell: But then you also got to add in the cost of downtime and the cost of data loss.

Paul Randal: That's the thing. It's what are your requirements, what are your limitations, and then compromising between the two and everybody has to agree on the compromise, but yeah, I was going to say supportive for ways, you can do clustering, you can do database marrying, in your log shipping you can do replication and each have different pros and cons, different impacts on what you can do and what happens on the database and the performance and so on and so on, but there's no easy way to say I would just recommend blah.

Richard Campbell: Right. Those are the reasons there are four methods, right. Of that list of four, only clustering and mirroring in theory offers that seamless failover.

Paul Randal: Well, clustering has its Achilles heel if there's only one copy of the data unless you have SAN replications in there too.

Richard Campbell: Right.

Paul Randal: So you're going to share the copy of the data, and even database mirroring it has its Achilles heel, it's only a single database at a time so if your application's ecosystem is more than one

database, then you can't do automatic failover but you can automatically failover multiple databases.

Richard Campbell: Okay but the bigger thing here I found is that you need programmers involve to create 100% uptime appearance because even when you have a cluster failover, you knockout one server to switch it to the other one, it's sometime, and in my experience it's been a couple of minutes for that machine to get back online.

Paul Randal: Yup. I mean, it totally depends. I mean, you've got to wait for the -- at worse case, you're going to wait for the SQL Server, it's a live check which actually, the cluster server logs into SQL Server, or at least tries to and does a select at that version to make sure the SQL Server doesn't just response to a ping that can actually be doing something interactive and that could take a minute before that fails.

Richard Campbell: Right.

Paul Randal: And then you've got to wait for the instance to start off on your other cluster mode, all the database is to run through crash recovery, and then you've got to have your application actually realize that the connection has been dropped and do a graceful reconnect. I remember the first days of amazon.com where I tried to buy something and I got an argument layer error message back.

Richard Campbell: Love it.

Paul Randal: That's a fail.

Richard Campbell: Fail?

Paul Randal: Of course they don't do that now. Now you get "we are down for downtime, blah, blah, blah." But the application designer has to be able to cope with something, a connection dropping out underneath and of course knowing that whatever the application does in the middle of, it lost. Anytime any kind of failover happens, this is a big misconception, anytime any kind of failover happens, everything that was happening at that point in the database gets rolled back.

Richard Campbell: Right.

Paul Randal: So your application either has to have some kind of states so it knows what it was doing or it has to be able to gracefully cope with. Everything it was doing suddenly gets drop on the floor and that can be hard to do.

Richard Campbell: So you've already sent your transaction off to the database and sometime after that you get back, not completion, but connection lost.

Paul Randal: Yes.

Richard Campbell: You have to presume your transaction has failed, remember what it was and go try it again, but it may be a couple of minutes before you can try it again.

Paul Randal: Absolutely. Now what's even more tricky for a developer is if you're not using some kind of system where there is a guarantee. If the transaction commits, then after the failover the transaction is there. For instance, if you're using clustering with design application say or you're using database mirroring, synchronous database mirroring, then once the transaction is actually committed back to the application, if the failover occurs, the application loads and the transaction is going to be there and the databases back-up again. If you're not using even these two technologies, there are no guarantees. So if you're using for instance transactional replications and you've got an error load balance and set up in your mid-tier then you do a commit on the main load and there's some latency before the transaction actually gets read and popping it to the distributor and then to the subscriber. So if a failover occurs before that transaction gets there, then the application has to be able to tow with the fact that transaction may not be there which is kind of funky and some of the problems occurs if you're using replication, peer-to-peer replication for instance as a query scale out solution for the developers. This is the problem that I came across where customers can actually do the mid-tier and then there's a network load bouncing layer which at the backend it goes to, and so if a customer connects in and it goes to say load 1 on the backend, the transaction then commits and then reconnects through websites and gets network load bounce to another mode, say mode number 4. How much time has to go past before the network load bouncing layer knows that it's safe to redirect that customer to a different mode than where it went to the last time?

Richard Campbell: Right.

Paul Randal: In other words, is there any way to know what that latency is between per transactions to be replaced in different modes, and that's an incredibly difficult problem to solve.

Carl Franklin: Right.

Richard Campbell: As an IT pro, I can't solve it. I need the developer's help in that.

Paul Randal: Absolutely.

Richard Campbell: I mean, that's where I think that as much as we want to have this sort of wall between

dev and IT, or that the people perceived it's there, these conversations about how is our app going to tolerate this is how failure actually looks, what are we going to do to survive that, how are we going to avoid spitting that error message back to the customer, that's a very interesting challenge and it works both ways because now you throw -- there are three parties involved here, there's a guy who's building the software, the guy who needs to operate the software, and the guy who has made the agreement with the customers, the business owner, of how they expect the software ultimately to behave.

Paul Randal: That's cool.

Richard Campbell: Clustering maybe your only option because this is the only thing that's reliable enough and you still have to go third party too in external site like how are we going to solve, and then Hurricane came through and destroy the datacenter. I hate the fact that we call in -- sometimes you just say this is a RunAs topic, like you know what? Developers need to be involved in this because you won't succeed without them.

Carl Franklin: Yeah, you're right.

Paul Randal: Sure.

Richard Campbell: They need to know that these things are important and that ultimately if we don't do them we are all going to fail.

Carl Franklin: And we had a really good, a lot of good feedback on the show that we did on how to design a database with Adam Machanic.

Paul Randal: Oh Adam, yeah. Any interesting comments that we should try to address in the show?

Carl Franklin: Well, I just think, you know, we basically came up because we hardly ever talk about SQL Server from a maintenance or an IT point of view, and there's a lot of , as you said before, reluctant sort of involuntary DBAs out there that just end up being DBAs because nobody else knows about it. Somebody who listens to .NET Rocks! and picks up these little things might know more about SQL Server than most of the developers in the organization.

Paul Randal: Yeah, true.

Carl Franklin: Frighteningly.

Paul Randal: Yeah.

Richard Campbell: Hey Paul, how would the guy who got recruited who has basically told you he's now the DBA, where should he start? Where is the primer?

Paul Randal: There isn't a good one, that's the problem. There isn't a good primer.

Richard Campbell: You need to write the primer, Paul.

Paul Randal: If only I had time. See, Twitter gets in the way, that's my problem. I can't write a book, I'm too busy twittering and making lego. So absolutely there's no really, really good primer. I have heard anecdotally, I haven't read it, I have heard anecdotally that there is a new database administration book out for 2008, for SQL Server 2008 called Rows Mystery I believe that has had some good reviews in terms of being good. Pick it up and run with it if you've never been a DBA before. So you might want to check that out. Apart from that, I was going to say go and read people's blog and stuff, but if you're an accident DBA how do you know which people to go and follow and stuff like that.

Richard Campbell: Besides you of course.

Paul Randal: Besides me of course, yeah. Seriously, I mean, how do you find me if you got no idea. You're not just go and randomly type in Paul Randal unless you're actually a DBA that's been following me and knows me. How do you even find the right people?

Richard Campbell: Right.

Paul Randal: You could start on books online but books online doesn't even have a -- if your accidental DBAs, start here, here's what you need to know because there are so many different gotchas that can happen with being an accidental DBA and of course the number 1 is in terms of recovery model and log back-ups, that old chestnut.

Richard Campbell: Yeah. I was just thinking about that. You know, there's a very fundamental thing that folks need to know if you're just getting started about the different recovery models and how we do back-ups so do you want to run them down for us?

Paul Randal: Sure. Actually, you know what? There is a good place to start. Last August I wrote an article for TechNet Magazine called Effective Database Maintenance or Essential Database Maintenance and it's written for the accidental DBA. So TechNet Magazine, August 2008, and it's the feature article on the cover of the magazine. That's a really good place to start, and then there's a whole bunch of other TechNet Magazine articles that I've written with the kind of accidental DBA, IT pro that doesn't know anything about SQL Server in mind. In

fact tomorrow, tomorrow's issue will be the July issue and this one is about back-ups and how they work, so talking about recovery models and back-ups. So a lot of the times I see people get into problems where the transaction log has filled up.

Richard Campbell: Yeah.

Paul Randal: And so the database stops. If the transaction log is not set to be able to grow automatically where it grows and grows and grows and grows and runs out of space because nobody is monitoring it because they don't know how to because they're volunteer DBAs, the number one cause, absolute number one cause of this is going into the full recovery model and then taking a full back-up, taking a database back-up which sounds like a really good thing to do. Hey, you're in the full recovery model, everything is being logged, you're not going to lose data. Ooh, we should take a database back-up so we got a point for recovery. As soon as you take that full database back-up you are telling SQL Server I will now take log back-ups forevermore so that the log does not grow out of control. However, when you take that first full back-up, there's no big flashing warning light that comes on saying you now need to take log back-ups so that's how people get into trouble.

Richard Campbell: Right and as a developer I'm thinking, well, why would I bother backing-up the log, I've already backed-up the database, that's all I need.

Paul Randal: Right, absolutely but it's one of these idiosyncrasies that the SQL Server has that when you first go into the full recovery model, you're not really in the full recovery model. You actually stay in what's called the pseudo simple recovery model and in the simple recovery model every time a, I think, a checkpoint occurs which occurs every minute or so, that's say roughly, the transactional log gets cleared out so it doesn't have to grow. As soon as you go into full, it doesn't do that anymore once you take that back-up.

Richard Campbell: And once you're committed to backing up now, at least it's true full, and a lot of folks do switch it to simple because it makes the problem go away.

Paul Randal: They do. Now the problem is if you switch to simple then you can't take log back-ups which means you can't do point in time recovery or what's called up-to-the-minute recovery.

Richard Campbell: Right.

Paul Randal: And so you got to trade off between what do you want to do in terms of disaster recovery and high availability, and your ability to do database maintenance and to do things like monitoring the sizes of your log and data files. There's all kinds of things...

Richard Campbell: Well, and in the used cased, it comes back to this same old problem of if you're really taking a back-up once a day, can you afford to lose a day's worth of data?

Paul Randal: Right, that's the thing that I say every time. Do you realize that you're going to lose everything that happens since your last full back-up.

Richard Campbell: Right.

Paul Randal: I get people doing essentially bad things where they'll go into the full recovery model and once a day they'll take a full back-up and then they'll switch to simple just to clear the log out and then switch back to full again. Don't do that. Either go in full and take log back-ups, or go in simple and don't.

Richard Campbell: One of the other.

Paul Randal: Yes but here's the catch. You know, some people, imagine you want to use database mirroring, if you want to use database mirroring, you have no choice, you must use full recovery model.

Richard Campbell: Okay.

Paul Randal: Which means suddenly you are now taking log back-ups. But you can't just back him up and throw it away if you're not interested. The other point is if you're going to implement HA technologies you can't take back-ups. You have to do both. If you want a proper HA strategy, it's back-ups and some kind of HA technologies because if your HA technologies fail and you lost all your data, then you don't have back-ups to restore from, it's your job too and I've seen that happen oddly enough.

Richard Campbell: So given that I actually am running in full mode and I'm backing the database once a day and I back-up my transaction log periodically, am I able to recover from stuff like my software accidentally renamed every customer John Smith?

Paul Randal: Yes. It depends on how you would want to do it or you could restore your database back to the point and time just before it did that but then you'd have lost all the work, the up ones.

Richard Campbell: Yes.

Paul Randal: Or you could restore your database with a different name and then pull all the

contents of that screwed up table back over without losing the rest of the things that you have on the database, but the odds are that you got relational and that gets constraints all over the place and what's happening in the database is part of other transactions so you may have to just write in full and go back in time. Or you can do what's called point in time recovery and at any point in time as long as you have a log back-up that recovers that point in time...

Richard Campbell: Down to the millisecond kind of thing?

Paul Randal: You can go down to individual log records depending on what you want to do.

Richard Campbell: Wow.

Paul Randal: There's another catch which is kind of geeky, but this is a geek show, which is if you have a log back-up and in the time period covered by that log back-up, if you switch the database to the boat log recovery model and you did what's called the minimally logged operation, and I'll define these terms in a second, if you do the minimally logged operation and the time period recovered by that log back-up you cannot do any kind of stop out operation, you can't stop the recovery process, the restore process using that log back-up, you can go to it before it or you could go after it and any point after it but not during that log back-up.

Richard Campbell: Okay. What's the minimally logged operation?

Paul Randal: A minimally logged operation, there are certain operations that do lots and lots of stuff. For instance, we build an index or doing a boat log data where you can switch to what's called a Boat Log Recovery Model and instead of generating transactional log records where everything that happens, all it does is it generates log records or parts of the database being allocated or the actual inserts of the data which means it generates a lot less transaction log so the transaction log does not grow so much. Now your log back-up will be back at the same size almost as if it has done full recovery model because even though it doesn't generate as much transaction log, the log back-up has to have all the information necessary to be able to replay that operation so it picks up those few log records plus all of the actual data pages that changed because of that minimally logged operation, and because that log back-up has data pages in it and there's no information to say when during that time period those data pages changed, so you can't stop any point during that time period.

Richard Campbell: Right.

Paul Randal: So one thing to be aware of if you're a developer or even if you're a DBA listening to this, it's be careful about doing stuff in that boat log recovery model because you might not be able to do a stop at that you need to be able to do.

Richard Campbell: Hasn't it always been the rule that when you're going to do one of these minimally logged operations or have to flip the boat log or anything like that, the next thing you should do after that is take a full back-up?

Paul Randal: No, not a full back-up. The rule is if you're going into the boat log recovery model, first stop make sure that nothing happening during that time is not regenerating in some other way.

Richard Campbell: Right.

Paul Randal: Just before going into boat log, take a log back-up, switch over to boat log, do your operation, switch back to full, immediately take another log back-up. You don't need to take a full back-up, just a log back-up.

Richard Campbell: Right, okay.

Paul Randal: Gives you the unbroken chain of log back-ups that you're going to need to restore pass that point in time.

Richard Campbell: And again, this only matters if you want to be able to recover point in time. If you're okay with losing the work of the day and going to the back full back-up, then fine.

Paul Randal: Absolutely.

Richard Campbell: It's just a question of, you know, often we make these bets and get away with it and it becomes a practice without realizing the real consequences of what we did.

Paul Randal: Until you actually have a disaster and finding tons you wouldn't want to do. So this brings me to a great point. I always say don't ever, ever plan a back-up strategy. Plan a restore strategy.

Richard Campbell: Ah, very nice. Okay.

Paul Randal: And then figure out what back-ups you need to build and take to the restores you want to be able to do if disaster occurs.

Richard Campbell: Well, I think it's incredibly valuable to let your customer know, whether that's your boss or anybody else, how long a restore actually is going to take. That's how we've always have gotten more money for back-up systems. We'll

say, "Oh, by the way, if this dies, the fastest I can get you back up, given I had everything I need is a day."

Paul Randal: Yup.

Richard Campbell: "Are you prepare to be down for the day?" "No." "Well, then we should talk. I'm just telling you what you're currently up against." It's unfortunate that many companies only find out how quickly they'll recover when they finish recovering. Currently this takes a week.

Paul Randal: Yup and there are a few things you can do to basically speed up how long your restores are going to take. One of the fastest restores that I know of, there's a company called VWin.com that we worked with in the past. They're an online gambling firm and they're one of the major top customers of Microsoft and their DBA, Michael Thomas does presentation in the past in a lot of conference about some other systems and we were over in their datacenter in Vienna and he was telling us that they can restore terabytes of data in 36 minutes.

Richard Campbell: Holy cow, that's like breaking the speed of light.

Paul Randal: Yeah. They're using SQL Server 2008 and their back-up device is 12 separate spindles, and so they're backing up to 12 separate files, one in each of these spindles, 15,000 RPM drives in a back-up stripe set and they're using 2008 back-up compressions. So they can do 2 terabytes in 36 minutes which is astonishing. So the things that you can do to speed up your restores are, one, use compression because that speeds up your back-up and speeds up your restore at the expense of a loaded CPU. For hardware methods, this is one where you can just throw hardware at the problem. The more spindles you can have and the faster they are, then the faster the reads and writes are going to be of those back-ups. Another thing you can do is you can use the thing called instant initialization on SQL Server and what this is is the first phase of a restore is always the file doesn't exist, create the file. By default, SQL Server is going to serialize the contents of that file, reason being the NTFS doesn't know what the trusted high watermark of that file is so the general way of doing that is write sequentially to the file and every time you do a write high up in the files, the NTFS high watermark moves up and NTFS knows to trust that, that portion of the file. Zeroing eyes of the file is very, very slow especially if you've got terabytes size files, you have zero bytes.

Richard Campbell: Yeah.

Paul Randal: So what you can do is you can grant permission to the SQL Server service again called Perform Volume Maintenance Task or SeManage Volume Data, and what that allows SQL Server to do is not have to be zeroing when it raise the file. What it can do is it can call NTFS API called SeFile Valid Data and what that does is to say here's the high watermark and the file trust me, don't ask questions.

Carl Franklin: Hey Paul, we're just out of time. Is there any last -- well, let me ask you this. RAID, RAID has been the biggest pain in my ass like I can't explain how frustrating RAID is.

Paul Randal: Always use a cushion.

[Laughter]

Carl Franklin: Yeah, I know, they have a cream for that.

Paul Randal: We've been so serious, we've got to say something rude.

Carl Franklin: They have a cream for that, yeah.

Paul Randal: They do, yeah.

Carl Franklin: But no, seriously I mean I can't wait for Solid State to really take over because we wouldn't need RAID.

Paul Randal: Why? Why would we not need RAID?

Carl Franklin: Well, that's another show, really.

Paul Randal: That's a whole other show.

Carl Franklin: Maybe you can tell me when we're done, but with regular old disks and SCSI and SATA and all of these stuff, what RAID configurations work best for what types of databases?

Paul Randal: That's pretty simple. So if you got a read mostly database, then you could stick it on RAID 5. If you have a read/write or write mostly, then RAID 10 or RAID 1, RAID 1 or RAID 10. You pay a performance penalty on writes with RAID 5.

Richard Campbell: In exchange for disk efficiency.

Paul Randal: In exchange for disk efficiency and your SAN administrator is going to try and give you RAID 5 because it uses the least amount of his disks to give you the capacity you want. RAID 10 uses the most amount of disks.

Carl Franklin: RAID 1 + 0.

Paul Randal: RAID 1 + 0, yeah. Create a couple of mirrors and stripe across them.

Richard Campbell: You know, the whole thing with RAID 10 is it's two drives for one and in RAID 5 it's number of drives plus one.

Paul Randal: Yup.

Richard Campbell: To get your capacities

Paul Randal: So going to SSDs, all it does is make the drives faster. You're still going to have to do RAID for redundancy.

Carl Franklin: Well, but you don't have to do striping. Striping is really where you get screwed up because if you got a mirror and one of them blows up, it's really easy to recover from that. Let's say, in one of your disk in a stripe blows up and then the software can't put it back together again because it's brain dead and you have a problem now you've got all this...

Paul Randal: Striping is more for performance rather than...

Carl Franklin: Yeah, I know that but if it can't rebuild the stripe, you're screwed is what I'm saying.

Paul Randal: But you're going to have the same problem with SSDs.

Carl Franklin: Well, you wouldn't stripe SSDs is what I'm saying.

Paul Randal: Why not? All SSDs do is reduce the latency in sick times where a known number of...

Carl Franklin: But also what striping does is it makes them dependent on each other. So you're dependent now on the RAID system's ability to rebuild that array and if it can't do it for whatever reason, you know, your driver maybe it's running some weird Linux embedded thing...

Paul Randal: Should I make a joke or should I not say...

Carl Franklin: I'm no that's alright. You get what you pay for.

Paul Randal: Yeah but that's why you have back-ups as well. You're using striping for performance and you've got to use back-ups as well for added dependency. You know, you do RAID 10, it gives you performance and it gives you redundancy and you have that back-up as well. You can't just trust the I/Os...

Carl Franklin: I would think using SSDs with spans would be safer because you don't -- do you really need the performance...?

Paul Randal: It depends.

Carl Franklin: That are RAID, you know, with SSDs?

Paul Randal: I mean, eventually people will push the limits of SSDs as well. I mean, in the same company that I was with VWin, they do 400,000 SQL statements per second.

Richard Campbell: That's a lot of SQL statements.

Paul Randal: That's a lot of SQL statements and I don't know if I ever work with this that's harder than that that's publicized and they need the performance so they need to be out of stripe as well, but that's a whole other show.

Richard Campbell: Definitely a whole other show, but it's interesting to hear you say if you can afford the disk space, RAID 10 is always the right way...

Paul Randal: Oh yeah.

Richard Campbell: And if you can't, then RAID 5 but with RAID 5 you always pay a penalty for writing.

Paul Randal: Right.

Richard Campbell: Nice.

Paul Randal: For SQL Server, there's a whitepaper called Physical Database Storage Design that talks about beautiful RAID configurations and database layouts and how to go about doing that, what the choices are for the various different workloads. So that's worth checking out as well.

Richard Campbell: And I presume you're in the camp that says the system drive, the database drive, and the long drive are separate drives.

Paul Randal: Yes but again it depends. I mean, if you got some really high performing SAN, does it really make a difference. It depends on the I/O subsistence underneath. In general, and generalizations are dangerous things to make, in general the answer is yes, they should be separated based on the degree of workload in terms of reads and writes.

Richard Campbell: I don't believe that the SANs actually make everything magically better. If the SAN

administrator is assigning everything into the same set of spindles, you're screwed.

Paul Randal: Right, so that's why I said it totally depends. It depends on -- in general, yes, you should be aware of what's happening for each of your different files and the I/O loads on them and monitoring your disk cue lengths. That's the thing. If your disk cue lengths are going up, then you've got to break it out.

Richard Campbell: Oh boy, we're talking about PerfMon on .NET Rocks!

Paul Randal: Sorry.

Richard Campbell: Yeah, I'm with you. I'm a big believer in PerfMon and it's been a good tool for me depending on what hat I had on. If you're in the performance tuning business, you need to know how PerfMon is going to help you and disk cue lengths is your tip, your drives are in trouble.

Paul Randal: I could argue a developer should be looking at this to see what effect the cruise that they're running on the database is having I/Os that's pushing out the I/O subsystem.

Richard Campbell: Right.

Paul Randal: See if it's going to overload the I/O subsystem that's in production. There's no reason developers shouldn't be looking at this stuff too and you start to talk about high breed developers, performance tuners, DBAs, and...

Richard Campbell: It is all the same problem but this cue lengths, the correct number is zero.

Paul Randal: Well, very low.

Richard Campbell: Yeah and as the number rises above one, you should be concerned.

Paul Randal: Exactly, yes.

Richard Campbell: Because really you're now talking there's an I/O operation waiting to be done, waiting for this system to do it.

Paul Randal: Right.

Richard Campbell: And that's always bad, that's time going off the clock.

Paul Randal: Yup and SQL Server is another interesting thing to look out for. If you see page I/O latch waits in your error log, that usually says your I/O subsystem is underpowered.

Richard Campbell: You're hammered.

Paul Randal: You're hammering it.

Carl Franklin: All right. Well, what can I say? It's been an interesting show for me to listen to, but, no seriously I learned a lot and I always do when we talk about this stuff and I hope the developers who are out there who are doing some more SQL Server content really appreciate it. If you like what you hear, or if you got any comments, send it to us at dotnetrocks@franklins.net. Paul, thank you.

Paul Randal: Thank you.

Carl Franklin: Thank that beautiful wife of yours for all the work she does with you...

Paul Randal: I certainly shall.

Carl Franklin: And we'll see you next time on .NET Rocks!

[Music]

Carl Franklin: .NET Rocks! is recorded and produced by PWOP Productions, providing professional audio, audio mastering, video, post production, and podcasting services, online at www.pwop.com. .NET Rocks! is a production of Franklins.NET, training developers to work smarter and offering custom onsite classes in Microsoft development technology with expert developers, online at www.franklins.net. For more .NET Rocks! episodes and to subscribe to the podcast feeds, go to our website at www.dotnetrocks.com.